A Unigram Orientation Model for Statistical Machine Translation

Christoph Tillmann
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
ctill@us.ibm.com

Abstract

In this paper, we present a unigram segmentation model for statistical machine translation where the segmentation units are blocks: pairs of phrases without internal structure. The segmentation model uses a novel orientation component to handle swapping of neighbor blocks. During training, we collect block unigram counts with **orientation**: we count how often a block occurs to the left or to the right of some predecessor block. The orientation model is shown to improve translation performance over two models: 1) no block re-ordering is used, and 2) the block swapping is controlled only by a language model. We show experimental results on a standard Arabic-English translation task.

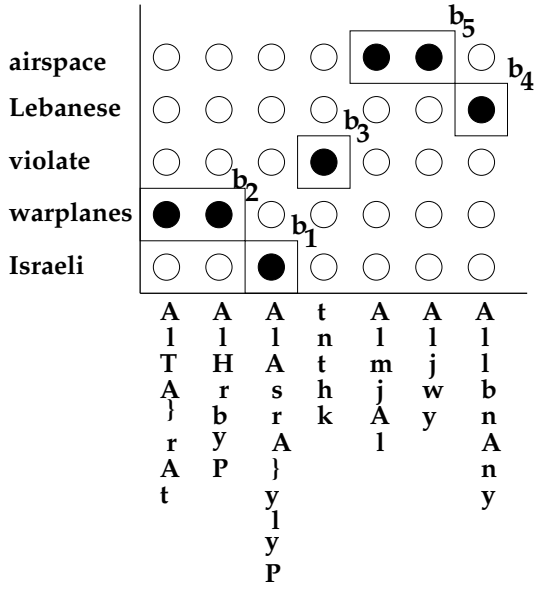

Figure 1: An Arabic-English block translation example taken from the **devtest** set. The Arabic words are romanized.

1 Introduction

In recent years, phrase-based systems for statistical machine translation (Och et al., 1999; Koehn et al., 2003; Venugopal et al., 2003) have delivered state-of-the-art performance on standard translation tasks. In this paper, we present a phrase-based unigram system similar to the one in (Tillmann and Xia, 2003), which is extended by an unigram orientation model. The units of translation are blocks, pairs of phrases without internal structure. Fig. 1 shows an example block translation using five Arabic-English blocks $b_1, \cdots, b_5$. The unigram orientation model is trained from word-aligned training data. During decoding, we view translation as a block segmentation process, where the input sentence is segmented from left to right and the target sentence is generated from bottom to top, one block at a time. A monotone block sequence is generated except for the possibility to swap a pair of neighbor blocks. The novel orientation model is used to assist the block swapping: as shown in section 3, block swapping where only a trigram language model is used to compute probabilities between neighbor blocks fails to improve translation performance. (Wu, 1996; Zens and Ney, 2003) present re-ordering models that make use of a straight/inverted orientation model that is related to our work. Here, we investigate in detail the effect of restricting the word re-ordering to neighbor block swapping only.

In this paper, we assume a block generation process that generates block sequences from bottom to top, one block at a time. The score of a successor block b depends on its predecessor block b' and on its orientation relative to the block b'. In Fig. 1 for example, block b_1 is the predecessor of block b_2, and block b_2 is the predecessor of block b_3. The target clump of a predecessor block b' is adja-

cent to the target clump of a successor block b. A right adjacent predecessor block b' is a block where additionally the source clumps are adjacent and the source clump of b' occurs to the right of the source clump of b. A left adjacent predecessor block is defined accordingly.

During decoding, we compute the score $P(b_1^n, o_1^n)$ of a block sequence b_1^n with orientation o_1^n as a product of block bigram scores:

$$P(b_1^n, o_1^n) \approx \prod_{i=1}^{n} p(b_i, o_i | b_{i-1}, o_{i-1}), \qquad (1)$$

where b_i is a block and $o_i \in \{L, R, N\}$ is a three-valued orientation component linked to the block b_i (the orientation o_{i-1} of the predecessor block is ignored.). A block b_i has **right** orientation ($o_i = R$) if it has a left adjacent predecessor. Accordingly, a block b_i has **left** orientation ($o_i = L$) if it has a right adjacent predecessor. If a block has **neither** a left or right adjacent predecessor, its orientation is neutral ($o_i = N$). The neutral orientation is not modeled explicitly in this paper, rather it is handled as a default case as explained below. In Fig. 1, the orientation sequence is $o_1^5 = (N, L, N, N, L)$, i.e. block b_2 and block b_5 are generated using left orientation. During decoding most blocks have right orientation ($o = R$), since the block translations are mostly monotone.

We try to find a block sequence with orientation (b_1^n, o_1^n) that maximizes $P(b_1^n, o_1^n)$. The following **three** types of parameters are used to model the block bigram score $p(b_i, o_i | b_{i-1}, o_{i-1})$ in Eq. 1:

- **Two unigram count-based models: $p(b)$ and $p_b(o)$.** We compute the unigram probability $p(b)$ of a block based on its occurrence count $N(b)$. The blocks are counted from word-aligned training data. We also collect unigram counts with **orientation**: a left count $N_L(b)$ and a right count $N_R(b)$. These counts are defined via an enumeration process and are used to define the orientation model $p_b(o)$:

$$p_b(o \in \{L, R\}) = \frac{N_o(b)}{N_L(b) + N_R(b)}.$$

- **Trigram language model:** The block language model score $p(b_i | b_{i-1})$ is computed as the probability of the first target word in the target clump of b_i given the final two words of the target clump of b_{i-1}.

The three models are combined in a log-linear way, as shown in the following section.

2 Orientation Unigram Model

The basic idea of the orientation model can be illustrated as follows: In the example translation in Fig. 1, block b_2 occurs to the left of block b_1. Although the joint block (b_2, b_1) consisting of the two smaller blocks b_1 and b_2 has not been seen in the training data, we can still profit from the fact that block b_2 occurs more frequently with left than with right orientation. In our Arabic-English training data, block b_2 has been seen $N_L(b_2) = 52$ times with left orientation, and $N_R(b_2) = 0$ with right orientation, i.e. it is always involved in swapping. This intuition is formalized using unigram counts with orientation.

The orientation model is related to the distortion model in (Brown et al., 1993), but we do not compute a block alignment during training. We rather enumerate all relevant blocks in some order. Enumeration does not allow us to capture position dependent distortion probabilities, but we can compute statistics about adjacent block predecessors.

Our baseline model is the unigram monotone model described in (Tillmann and Xia, 2003). Here, we select blocks b from word-aligned training data and unigram block occurrence counts $N(b)$ are computed: all blocks for a training sentence pair are enumerated in some order and we count how often a given block occurs in the parallel training data [1]. The training algorithm yields a list of about 65 blocks per training sentence pair. In this paper, we make extended use of the baseline enumeration procedure: for each block b, we additionally enumerate all its left and right predecessors b'. No optimal block segmentation is needed to compute the predecessors: for each block b, we check for adjacent predecessor blocks b' that also occur in the enumeration list. We compute left orientation counts $N_L(b)$ as follows:

$$N_L(b) = \sum_{\exists\ b' \text{ right adjacent predecessor of } b} 1.$$

Here, we enumerate all adjacent predecessors b' of block b over all training sentence pairs. The identity of b' is ignored. $N_L(b)$ is the number of times the block b succeeds some right adjacent predecessor block b'. The 'right' orientation count $N_R(b)$ is defined accordingly. Note, that in general the unigram count $N(b) \neq N_L(b) + N_R(b)$: during enumeration, a block b might have both left and right adjacent predecessors, either a left or a right adjacent predecessor, or no adjacent predecessors at all. The orientation count collection is illustrated in Fig. 2: each time a block b has a left or right adjacent predecessor in the parallel training data, the orientation counts are incremented accordingly.

The decoding orientation restrictions are illustrated in Fig 3: a monotone block sequence with right ($o = R$)

[1] We keep all blocks for which $N(b) \geq 2$ and the phrase length is less or equal 8. No other selection criteria are applied. For the $S1$ & OR model, we keep all blocks for which $N(b) \geq 3$.

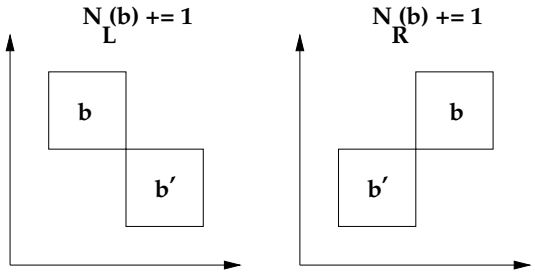

Figure 2: During training, blocks are enumerated in some order: for each block b, we look for left and right adjacent predecessors b'.

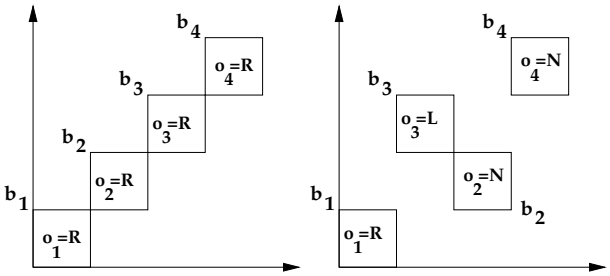

Figure 3: During decoding, a mostly monotone block sequence with $(o_i = R)$ orientation is generated as shown in the left picture. In the right picture, block swapping generates block b_3 to the left of block b_2. The blocks b_2 and b_4 do not have a left or right adjacent predecessor.

orientation is generated. If a block is skipped e.g. block b_3 in Fig 3 by first generating block b_2 then block b_3, the block b_3 is generated using left orientation $o_3 = L$. Since the block translation is generated from bottom-to-top, the blocks b_2 and b_4 do not have adjacent predecessors below them: they are generated by a default model $p(b_i|b_{i-1})$ without orientation component. The orientation model is given in Eq. 2, the default model is given in Eq. 3. The block bigram model $p(b_i, o_i \in \{L, R\}|b_{i-1}, o_{i-1})$ in Eq. 1 is defined as:

$$p(b_i, o_i \in \{L, R\} \mid b_{i-1}, o_{i-1}) = \quad (2)$$
$$= \quad p(b_i)^{\alpha_0} \cdot p(b_i|b_{i-1})^{\alpha_1} \cdot p_{b_i}(o_i)^{\alpha_2},$$

where $\alpha_0 + \alpha_1 + \alpha_2 = 1.0$ and the orientation o_{i-1} of the predecessor is ignored. The α_i are chosen to be optimal on the devtest set (the optimal parameter setting is shown in Table. 1). Only two parameters have to be optimized due to the constraint that the α_i have to sum to 1.0. The default model $p(b_i, o_i = N|b_{i-1}, o_{i-1}) = p(b_i|b_{i-1})$ is defined as:

$$p(b_i|b_{i-1}) \quad = \quad p(b_i)^{\alpha_0'} \cdot p(b_i|b_{i-1})^{\alpha_1'}, \quad (3)$$

where $\alpha_0' + \alpha_1' = 1.0$. The α_i' are not optimized separately, rather we define: $\alpha_0' = \frac{\alpha_0}{\alpha_0 + \alpha_1}$.

Straightforward normalization over all successor blocks in Eq. 2 and in Eq. 3 is not feasible: there are tens of millions of possible successor blocks b. In future work, normalization over a restricted successor set, e.g. for a given source input sentence, all blocks b that match this sentence might be useful for both training and decoding. The segmentation model in Eq. 1 naturally prefers translations that make use of a smaller number of blocks which leads to a smaller number of factors in Eq. 1. Using fewer 'bigger' blocks to carry out the translation generally seems to improve translation performance. Since normalization does not influence the number of blocks used to carry out the translation, it might be less important for our segmentation model.

We use a DP-based beam search procedure similar to the one presented in (Tillmann and Xia, 2003). We maximize over all block segmentations with orientation (b_1^n, o_1^n) for which the source phrases yield a segmentation of the input sentence. Swapping involves only blocks (b, b') for which $N_L(b) \geq 3$ for the successor block b, e.g. the blocks b_2 and b_5 in Fig 1. We tried several thresholds for $N_L(b)$, and performance is reduced significantly only if $N_L(b) > 30$. No other parameters are used to control the block swapping. In particular the orientation o' of the predecessor block b' is ignored: in future work, we might take into account that a certain predecessor block b' typically precedes other blocks.

3 Experimental Results

The translation system is tested on an Arabic-to-English translation task. The training data comes from the UN news sources: 87.5 million Arabic and 97.1 million English words. The training data is sentence-aligned yielding 3.3 million training sentence pairs. The Arabic data is romanized, some punctuation tokenization and some number classing are carried out on the English and the Arabic training data. As devtest set, we use testing data provided by LDC, which consists of $1\,043$ sentences with $25\,889$ Arabic words with 4 reference translations. As a blind test set, we use MT 03 Arabic-English DARPA evaluation test set consisting of 663 sentences with $16\,278$ Arabic words.

Three systems are evaluated in our experiments: $S0$ is the baseline block unigram model without re-ordering. Here, monotone block alignments are generated: the blocks b_i have only left predecessors (no blocks are swapped). This is the model presented in (Tillmann and Xia, 2003). For the $S1$ model, the sentence is translated mostly monotonously, and only neighbor blocks are allowed to be swapped (at most 1 block is skipped). The $S1$ & OR model allows for the same block swapping as the $S1$ model, but additionally uses the **orientation** component described in Section 2: the block swapping is controlled

Table 1: Effect of the orientation model on Arabic-English test data: LDC **devtest** set and DARPA MT 03 **blind test** set.

Test	Unigram Model	Setting $(\alpha_0/\alpha_1/\alpha_2)$	BLEUr4n4
Dev test	$S1$	$.74/.26$	0.344 ± 0.012
	$S0$	$.77/.23$	0.355 ± 0.013
	$S1$ & OR	$.66/.27/.07$	0.368 ± 0.014
Test	$S1$	$.74/.26$	0.336 ± 0.017
	$S0$	$.77/.23$	0.339 ± 0.016
	$S1$ & OR	$.66/.27/.07$	0.356 ± 0.017

Table 2: Arabic-English example blocks from the devtest set: the Arabic phrases are romanized. The example blocks were swapped in the development test set translations. The counts are obtained from the parallel training data.

Arabic-English blocks	$N_L(b)$	$N_R(b)$
('exhibition' | 'mErD')	97	32
('added' | 'wADAf')	285	68
('said' | 'wqAl')	872	801
('suggested | 'AqtrH')	356	729
('terrorist attacks' | hjmAt ArhAbyp')	14	27

by the unigram orientation counts. The $S0$ and $S1$ models use the block bigram model in Eq. 3: all blocks b are generated with neutral orientation ($o = N$), and only two components, the block unigram model $p(b_i)$ and the block bigram score $p(b_i|b_{i-1})$ are used.

Experimental results are reported in Table 1: three BLEU results are presented for both devtest set and blind test set. **Two** scaling parameters are set on the devtest set and copied for use on the blind test set. The second column shows the model name, the third column presents the optimal weighting as obtained from the devtest set by carrying out an exhaustive grid search. The fourth column shows BLEU results together with confidence intervals (Here, the word casing is ignored). The block swapping model $S1$ & OR obtains a statistical significant improvement over the baseline $S0$ model. Interestingly, the swapping model $S1$ without orientation performs worse than the baseline $S0$ model: the word-based trigram language model alone is too weak to control the block swapping: the model is too unrestrictive to handle the block swapping reliably. Additionally, Table 2 presents devtest set example blocks that have actually been swapped. The training data is unsegmented, as can be seen from the first two blocks. The block in the first line has been seen 3 times more often with left than with right orientation. Blocks for which the ratio $r = \frac{N_L(b)}{N_R(b)}$ is bigger than 0.25 are likely candidates for swapping in our Arabic-English experiments. The ratio r itself is not currently used in the orientation model. The orientation model mostly effects blocks where the Arabic and English words are verbs or nouns. As shown in Fig. 1, the orientation model uses the orientation probability $p_L(b_2)$ for the noun block b_2, and only the default model for the adjective block b_1. Although the noun block might occur by itself without adjective, the swapping is not controlled by the occurrence of the adjective block b_1 (which does not have adjacent predecessors). We rather model the fact that a noun block b is typically preceded by some block b'. This situation seems typical for the block swapping that occurs on the evaluation test set.

Acknowledgment

This work was partially supported by DARPA and monitored by SPAWAR under contract No. N66001-99-2-8916. The paper has greatly profited from discussion with Kishore Papineni and Fei Xia.

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of the HLT-NAACL 2003 conference*, pages 127–133, Edmonton, Canada, May.

Franz-Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC 99)*, pages 20–28, College Park, MD, June.

Christoph Tillmann and Fei Xia. 2003. A Phrase-based Unigram Model for Statistical Machine Translation. In *Companian Vol. of the Joint HLT and NAACL Conference (HLT 03)*, pages 106–108, Edmonton, Canada, June.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2003. Effective Phrase Translation Extraction from Alignment Models. In *Proc. of the 41st Annual Conf. of the Association for Computational Linguistics (ACL 03)*, pages 319–326, Sapporo, Japan, July.

Dekai Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In *Proc. of the 34th Annual Conf. of the Association for Computational Linguistics (ACL 96)*, pages 152–158, Santa Cruz, CA, June.

Richard Zens and Hermann Ney. 2003. A Comparative Study on Reordering Constraints in Statistical Machine Translation. In *Proc. of the 41st Annual Conf. of the Association for Computational Linguistics (ACL 03)*, pages 144–151, Sapporo, Japan, July.