A Projection Extension Algorithm 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 describe a phrase-based unigram model for statistical machine translation that uses a much simpler set of model parameters than similar phrase-based models. The units of translation are blocks – pairs of phrases. During decoding, we use a block unigram model and a word-based trigram language model. During training, the blocks are learned from source interval projections using an underlying high-precision word alignment. The system performance is significantly increased by applying a novel block extension algorithm using an additional high-recall word alignment. The blocks are further filtered using unigram-count selection criteria. The system has been successfully test on a Chinese-English and an Arabic-English translation task.

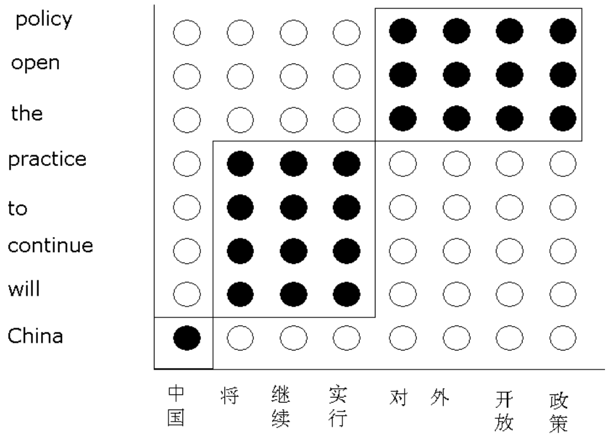

Figure 1: A block sequence that jointly generates 3 target and source phrases. The example is actual decoder output and the English translation is slightly incorrect.

1 Introduction

Various papers use phrase-based translation systems (Och et al., 1999; Marcu and Wong, 2002; Yamada and Knight, 2002) that have shown to improve translation quality over single-word based translation systems introduced in (Brown et al., 1993). In this paper, we present a similar system with a much simpler set of model parameters. Specifically, we compute the probability of a block sequence b_1^n. A block b is a pair consisting of a contiguous source and a contiguous target phrase. The block sequence probability $Pr(b_1^n)$ is decomposed into conditional probabilities using the chain rule:

$$\begin{aligned} Pr(b_1^n) &\approx \prod_{i=1}^{n} Pr(b_i|b_{i-1}) \qquad (1) \\ &= \prod_{i=1}^{n} p^{\alpha}(b_i) \cdot p^{(1-\alpha)}(b_i|b_{i-1}) \end{aligned}$$

We try to find the block sequence that maximizes $Pr(b_1^n)$: $b_1^n = argmax_{b_1^n} Pr(b_1^n)$. The model proposed is a joint model as in (Marcu and Wong, 2002), since target and source phrases are generated jointly. The approach is illustrated in Figure 1. The source phrases are given on the x-axis and the target phrases are given on the y-axis. During block decoding a bijection between source and target phrases is generated. The two types of parameters in Eq 1 are defined as:

- **Block unigram model $p(b_i)$:** We compute unigram probabilities for the blocks. The blocks are simpler than the alignment templates (Och et al., 1999) in that they do not have an internal structure.

- **Trigram language model:** the probability $p(b_i|b_{i-1})$ between adjacent blocks 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 exponent α is set in informal experiments to be 0.5. No other parameters such as distortion probabilities are used.

To select blocks b from training data, we compute unigram block co-occurrence counts $N(b)$. $N(b)$ cannot be computed for all blocks in the training data: we would obtain hundreds of millions of blocks. The blocks are restricted by an underlying word alignment. In this paper, we present a block generation algorithm similar to the one in (Och et al., 1999) in full detail: source intervals are projected into target intervals under a restriction derived from a high-precision word alignment. The projection yields a set of high-precision block links. These block links are further extended using a high-recall word alignment. The block extension algorithm is shown to improve translation performance significantly. The system is tested on a Chinese-English (CE) and an Arabic-English (AE) translation task.

The paper is structured as follows: in Section 2, we present the baseline block generation algorithm. The block extension approach is described in Section 2.1. Section 3 describes a DP-based decoder using blocks. Experimental results are presented in Section 4.

2 Block Generation Algorithm

Starting point for the block generation algorithm is a word alignment obtained from an HMM Viterbi training (Vogel et al., 1996). The HMM Viterbi training is carried out twice with English as target language and Chinese as source language and vice versa. We obtain two alignment relations:

$$\mathcal{A}_1 \;=\; \{(x, a_x) | a : x \rightarrow y\}$$

$$\mathcal{A}_2 \;=\; \{(b_y, y) | b : y \rightarrow x\}$$

$a : x \rightarrow y$ is an alignment function from source to target positions and $b : y \rightarrow x$ is an alignment function from target to source positions [1]. We compute the union and the intersection of the two alignment relations $\mathcal{A}_1$ and $\mathcal{A}_2$:

$$\begin{aligned}\mathcal{P} &= \mathcal{A}_1 \cap \mathcal{A}_2 \\ \mathcal{R} &= \mathcal{A}_1 \cup \mathcal{A}_2\end{aligned}$$

We call the intersection relation $\mathcal{P}$, because it represents a high-precision alignment, and the union alignment $\mathcal{R}$, because it is taken to be a lower precision higher recall alignment (Och and Ney, 2000). The intersection $\mathcal{P}$ is also a (partial) bijection between the target and source positions: it covers the same number of target and source positions and there is a bijection between source and target positions that are covered. For the CE experiments reported in Section 4 about 45% of the target and source positions are covered by word links in $\mathcal{P}$, for the AE experiments about 65% are covered. The extension algorithm presented assumes that $\mathcal{P} \subseteq \mathcal{R}$, which is valid in this case since $\mathcal{P}$ and $\mathcal{R}$ are derived from intersection and union. We introduce the following additional piece of notation:

$$col(\mathcal{P}) \;=\; \{x \mid \exists y \text{ and } (x,y) \in \mathcal{P}\} \qquad (2)$$

$col(\mathcal{P})$ is the set of all source positions that are covered by some word links in $\mathcal{P}$, where the source positions are shown along the x-axis and the target positions are shown along the y-axis. To derive high-precision block links from the high-precision word links, we use the following projection definitions:

$$P_x([x', x]) \;=\; \{v \mid \exists u \in [x', x] \text{ and } (u, v) \in \mathcal{P}\}$$

Here, $P_x(\cdot)$ projects source intervals into target intervals. $P_y(\cdot)$ projects target intervals into source intervals and is defined accordingly. Starting from the high-precision word alignment $\mathcal{P}$, we try to derive a high-precision block alignment: we project source intervals $[x', x]$, where $x', x \in col(\mathcal{P})$. We compute the minimum target index y' and maximum target index y for the word links $p \in \mathcal{P}$ that fall into the

[1] x and u denote a source positions. y and v denote a target positions.

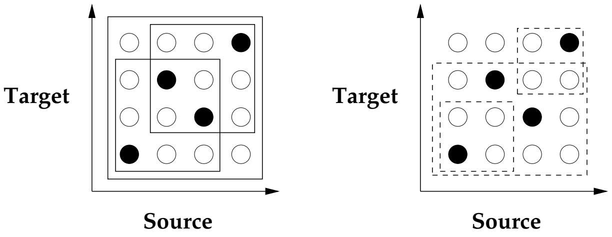

Figure 2: The left picture shows three blocks that are learned from projecting three source intervals. The right picture shows three blocks that cannot be obtain from source interval projections .

Table 1: Block learning algorithm using the intersection $\mathcal{P}$.

```
input: High-precision alignment $\mathcal{P}$
$\mathcal{A} := \emptyset$
for each interval $[x', x]$, where $x', x \in col(\mathcal{P})$ do
    $y' = \min_{v \in P_x([x',x])} v$;   $y = \max_{v \in P_x([x',x])} v$;
    Extend block link $a = ([x', x], [y', y])$ 'out--wards' using the algorithm in Table 2 and
    add extended block link set to $\mathcal{A}$
output: Sentence block link set $\mathcal{A}$.
```

interval $[x', x]$. This way, we obtain a mapping of source intervals into target intervals:

$$[x', x] \quad \rightarrow \quad [\min_{y \in P_x([x',x])} y \ , \ \max_{y \in P_x([x',x])} y] \qquad (3)$$

The approach is illustrated in Figure 2, where in the left picture, for example, the source interval $[1, 3]$ is projected into the target interval $[1, 3]$. The pair $([x', x], [y', y])$ defines a block alignment link a. We use this notation to emphasize that the identity of the words is not used in the block learning algorithm. To denote the block consisting of the target and source words at the link positions, we write $b = w(a)$:

$$\begin{aligned} b = w(a) \quad &= \quad (\ w([x', x])\ ,\ w([y', y])\ ) \\ &= \quad ((f_{x'}, \cdots, f_x), (e_{y'}, \cdots, e_y)), \end{aligned}$$

where f_x denote target words and e_y denote source words. $w(\cdot)$ denotes a function that maps intervals to the words at these intervals. The algorithm for generating the high-precision block alignment links is given in Table 1. The order in which the source intervals are generated does not change the final link set.

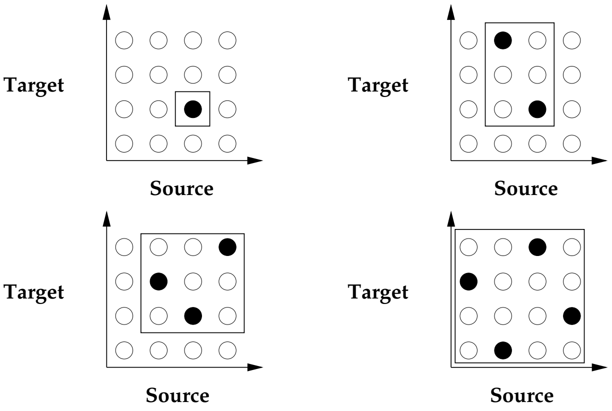

Figure 3: One, two, three, or four word links in $\mathcal{P}$ lie on the frontier of a block. Additional word links may be inside the block.

2.1 Block Extension Algorithm

Empirically, we find that expanding the high-precision block links significantly improves performance. The expansion is parameterised and described below. For a block link $a = ([x', x], [y', y])$, we compute its frontier $fr(a)$ by looking at all word links that lie on one of the four boundary lines of a block. We make the following observation as shown in Figure 3: the number of links (filled dots in the picture) on the frontier $fr(a)$ is less or equal 4, since in every column and row there is at most one link in $\mathcal{P}$, which is a partial bijetion. To learn blocks from a general word alignment that is not a bijection more than 4 word links may lie on the frontier of a block, but to compute all possible blocks, it is sufficient to look at all possible quadruples of word links. We extend the links on the frontier by links of the high-recall alignment $\mathcal{R}$, where we use a parameterised way of locally extending a given word link. We compute an extended link set $\mathcal{E}$ by extending each word link on the frontier separately and taking the union of the resulting links. The way a word link is extended is illustrated in Figure 4. The filled dot in the center of the picture is an element of the high-precision set $\mathcal{P}$. Starting from this link, we look for

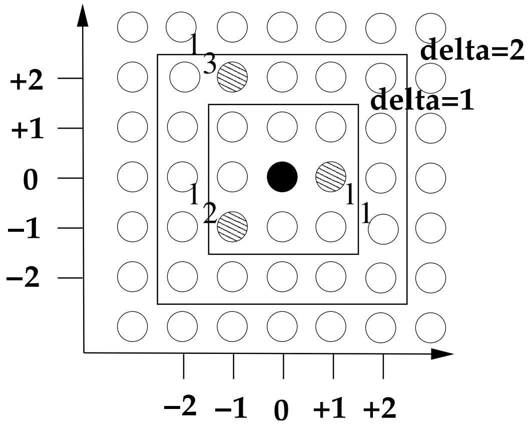

Figure 4: Point extension scheme. Solid word links lie in $\mathcal{P}$, striped word links lie in $\mathcal{R}$.

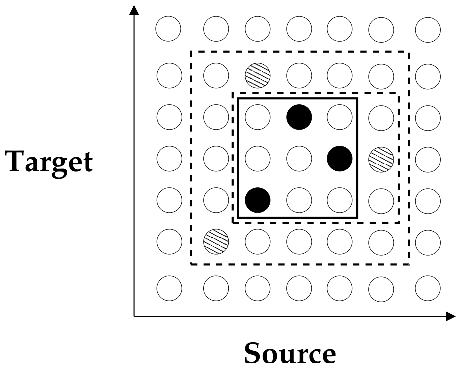

Figure 5: 'Outward' extension of a high-precision block link.

Table 2: Block link extension algorithm. The min and max function compute the minimum and the maximum of 4 integer values.

```
input: Block link $a = ([x', x], [y', y])$
$\mathcal{A} := \emptyset$
Compute extension set $\mathcal{E}$ from frontier $fr(a)$
for each $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \in \mathcal{E}$
    $u' = \min_{1 \leq k \leq 4} x_k$ ; $u = \max_{1 \leq k \leq 4} x_k$
    $v' = \min_{1 \leq k \leq 4} y_k$ ; $v = \max_{1 \leq k \leq 4} y_k$
    if ( $a \subseteq e = ([u', u], [v', v])$ )  $\mathcal{A} := \mathcal{A} \cup e$
output: Extended block link set $\mathcal{A}$.
```

extensions in its neighborhood that lie in $\mathcal{R}$, where the neighborhood is defined by a cell width parameter ω and a distance parameter δ. For instance, link l_1 in Figure 4 is reached with cell width $\omega = 1$ and distance $\delta = 1$, the link l_2 is reached with $\omega = 1$ and $\delta = 2$, the link l_3 is reached with $\omega = 2$ and $\delta = 3$. The word link l is added to $\mathcal{E}$ and it is itself extended using the same scheme. Here, we never make use of a row or a column covered by $\mathcal{P}$ other than the rows y and y' and the columns x and x'. Also, we do not cross such a row or column using an extension with $\delta \geq 2$: this way only a small fraction of the word links in $\mathcal{R}$ is used for extending a single block link. The extensions are carried out iteratively until no new alignment links from $\mathcal{R}$ are added to $\mathcal{E}$. The block extension algorithm in Table 2 uses the extension set $\mathcal{E}$ to generate all word link quadruples: the extended block e that is defined by a given quadruple is generated and a check is carried out whether e includes the seed block link a. The following definition for block link inclusion is used:

$$a' \subseteq a \quad := \quad [u', u] \subseteq [x', x] \text{ and } [v', v] \subseteq [y', y],$$

where the block $a' = ([u', u], [v', v])$ is said to be included in $a = ([x', x], [y', y])$. $[v', v] \subseteq [y', y]$ holds iff $v' \geq y'$ and $v \leq y$. The 'seed' block link a is extended 'outwardly': all extended blocks e include the high-precision block a. The block link a itself may be included in other high-precision block links a' on its part, but $a \subseteq e \subseteq a'$ holds. An extended block e derived from the block a never violates the projection restriction relative to $\mathcal{P}$ i.e., we do not have to re-check the projection restriction for any generated block, which simplifies and fastens up the generation algorithm. The approach is illustrated in Figure 5, where a high-precision block with 3 elements on its frontier is extended by two blocks containing it.

The block link extension algorithm produces block links that contain new source and target intervals $[x', x]$ and $[y', y]$ that extend the interval mapping in Eq. 3. This mapping is no longer a function, but rather a relation between source and target intervals i.e., a single source interval is mapped to several target intervals and vice versa. The extended block set constitutes a subset of the following set of interval pairs:

$$\{ \; ([x', x], [y', y]) \;\mid\; P_x([x', x]) \subseteq [y', y] \; \}$$

The set of high-precision blocks is contained in this set. We cannot use the entire set of blocks defined by all pairs in the above relation, the resulting set of blocks cannot be handled due to memory restrictions, which motivates our extension algorithm. We also tried the following symmetric restriction and tested the resulting block set:

$$P_x([x', x]) \subseteq [y, y] \text{ and } P_y([y', y]) \subseteq [x', x] \quad (4)$$

The modified restriction is implemented in the context of the extension scheme in Table 1 by inserting an **if** statement before the alignment link a is extended: the alignment link is extended only if the restriction $P_y([y', y]) \subseteq [x', x]$ also holds.
Considering only block links for which the two way projection in Eq. 4 holds has the following interesting interpretation: assuming a bijection $\mathcal{P}$ that is complete i.e., all source and target positions are covered, an efficient block segmentation algorithm exists to compute a Viterbi block alignment as in Figure 1 for a given training sentence pair. The complexity of the algorithm is quadratic in the length of the source sentence. This dynamic programming technique is not used in the current block selection but might be used in future work.

2.2 Unigram Block Selection

For selecting blocks from the candidate block links, we restrict ourselves to block links where target and source phrases are equal or less than 8 words long. This way we obtain some tens of millions of blocks on our training data including blocks that occur only once. This baseline set is further filtered using the unigram count $N(b)$: Nk denotes the set of blocks b for which $N(b) \geq k$. For our Chinese-English experiments, we use the $N2$ restriction as our baseline, and for the Arabic-English experiments the $N3$ restriction. Blocks where the target and the source clump are of length 1 are kept regardless of their count[2]. We compute the unigram probability $p(b)$ as relative frequency over all selected blocks.

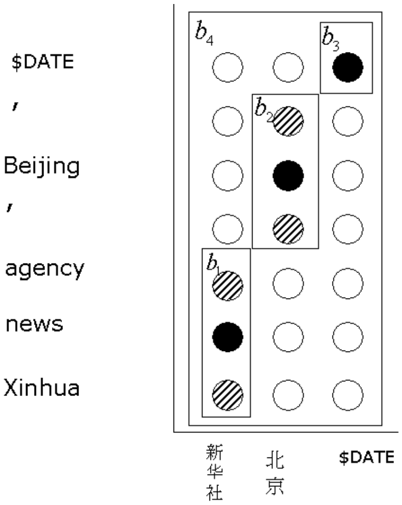

Figure 6: An example of 4 recursively nested blocks b_1, b_2, b_3, b_4.

An example of 4 blocks obtained from the Chinese-English training data is shown in Figure 6. '$DATE' is a placeholder for a date expression. Block b_4 contains the blocks b_1 to b_3. All 4 blocks are selected in training: the unigram decoder prefers b_4 even if b_1,b_2, and b_3 are much more frequent. The solid word links are word links in $\mathcal{P}$, the striped word links are word links in $\mathcal{R}$. Using the links in $\mathcal{R}$, we can learn one-to-many block translations, e.g. the pair (c_1,'Xinhua news agency') is learned from the training data.

3 DP-based Decoder

We use a DP-based beam search procedure similar to the one presented in (Tillmann and Ney, 2003). We maximize over all block segmentations b_1^n for which the source phrases yield a segmentation of the input source sentence, generating the target sentence simultaneously. The decoder processes search states of the following form:

$$\sigma \;=\; < \, e', \, e, \, \mathcal{C}, \, j, \, l', \, l \, > .$$

e and e' are the two predecessor words used for the trigram language model, $\mathcal{C}$ is the so-called coverage vector to keep track of the already processed source position, j is the last processed source position. l is the source phrase length of the block

[2] To apply the restrictions exhaustively, we have implemented tree-based data structures to store up to 46 million blocks with phrases of up to length 8 in less than 3 gigabyte of RAM.

Table 3: Effect of the extension scheme $\mathcal{P}_{\omega,\delta}$ on the CE translation experiments.

Scheme	# blocks $N(b) \geq 1$	# blocks $N(b) \geq 2$	BLEUr4n4
$\mathcal{R}_{0,0}$	41.88 M	6.53 M	0.148 ± 0.01
$\mathcal{P}_{0,0}$	14.77 M	2.67 M	0.160 ± 0.01
$\mathcal{P}_{1,1}$	24.47 M	4.50 M	0.180 ± 0.01
$\mathcal{P}_{1,2}$	35.23 M	6.18 M	0.183 ± 0.01
$\mathcal{P}_{2,2}$	37.92 M	6.65 M	0.183 ± 0.01
$\mathcal{P}_{2,3}$	45.81 M	7.66 M	0.181 ± 0.01

Table 4: Effect of the unigram threshold on the BLEU score. The maximum phrase length is 8.

Selection Restriction	# blocks selected	BLEUr4n4
N2	6.18 M	0.183 ± 0.01
N3	1.69 M	0.185 ± 0.01
N5	0.85 M	0.178 ± 0.01
N10	0.45 M	0.176 ± 0.01
N25	0.26 M	0.166 ± 0.01
N100	0.18 M	0.154 ± 0.01

currently being matched. l' is the length of the initial fragment of the source phrase that has been processed so far. l' is smaller or equal l: $l' \leq l$. Note, that the partial hypotheses are not distinguished according to the identity of the block itself. The decoder processes the input sentence 'cardinality synchronously': all partial hypotheses that are active at a given point cover the same number of input sentence words. The same beam-search pruning as described in (Tillmann and Ney, 2003) is used. The so-called observation pruning threshold is modified as follows: for each source interval that is being matched by a block source phrase at most the best 12 target phrases according to the joint unigram probability are hypothesized. The list of blocks that correspond to a matched source interval is stored in a chart for each input sentence. This way the matching is carried out only once for all partial hypotheses that try to match the same input sentence interval. In the current experiments, decoding without block re-ordering yields the best translation results. The decoder translates about 180 words per second.

4 Experimental Results

4.1 Chinese-English Experiments

The translation system is tested on a Chinese-to-English translation task. For testing, we use the DARPA/NIST MT 2001 dry-run testing data, which consists of 793 sentences with 20 333 words arranged in 80 documents [3]. The training data is provided by the LDC and labeled by NIST as the Large Data condition for the MT 2002 evaluation. The Chinese sentences are segmented into words. The training data contains 23.7 million Chinese and 25.3 million English words. The block selection algorithm described below runs less than one hour on a single 1-Gigahertz linux machine.

Table 3 presents results for various block extension schemes. The first column describes the extension scheme used. The second column reports the total number of blocks in millions collected - including all the blocks that occurred only once. The third column reports the number of blocks that occurred at least twice. These blocks are used to compute the results in the fourth column: the BLEU score (Papineni et al., 2002) with 4 reference translation using 4-grams along with 95% confidence interval is reported [4]. Line 1 and line 2 of this table show results where only the source interval projection without any extension is carried out. For the $\mathcal{R}_{0,0}$ extension scheme, the high-recall union set itself is used for projection. The results are worse than for all other schemes, since a lot of smaller blocks are discarded due to the projection approach. The $\mathcal{P}_{0,0}$ scheme, where just the $\mathcal{P}$ word links are used is too restrictive leaving out bigger blocks that are admissible according to $\mathcal{P}$. For the Chinese-English test data, there is only a minor difference between the different extension schemes, the best results are obtained for the $\mathcal{P}_{1,1}$ and the $\mathcal{P}_{1,2}$ extension schemes.

Table 4 shows the effect of the unigram selection threshold, where the $\mathcal{P}_{1,2}$ blocks are used. The second column shows the number of blocks selected. The best results are obtained for the $N2$ and the $N3$

[3] We removed the first 25 documents that are contained in the training data.

[4] The test data is split into a certain number of subsets. The BLEU score is computed on each subset. We use the t-test to compare these scores.

sets. The number of blocks can be reduced drastically where the translation performance declines only gradually.

Table 5 shows the effect of the maximum phrase length on the BLEU score for the $N2$ block set. Including blocks with longer phrases actually helps to improve performance, although already a length of 4 obtains nearly identical results.

We carried out the following control experiments (using $N(b) \geq 2$ as threshold): we obtained a block set of 3.91 million blocks by generating blocks from all quadruples of word links in $\mathcal{P}$ [5]. This set is a proper superset of the blocks learned for the $\mathcal{P}_{0,0}$ experiment in Table 3. The resulting BLEU score is 0.157. Including additional smaller blocks even hurts translation performance in this case. Also, for the extension scheme $\mathcal{P}_{1,2}$, we carried out the inverse projection as described in Section 2.1 to obtain a block set of 2.58 million blocks and a BLEU score of 0.165. This number is smaller than the BLEU score of 0.183 for the $\mathcal{P}_{1,2}$ restriction: for the translation direction Chinese-to-English, selecting blocks with longer English phrases seems to be important for good translation performance. It is interesting to note, that the unigram translation model is symmetric: the translation direction can be switched to English-to-Chinese without re-training the model - just a new Chinese language model is needed. Our experiments, though, show that there is an unbalance with respect to the projection direction that has a significant influence on the translation results. Finally, we carried out an experiment where we used the $\mathcal{P}_{0,0}$ block set as a baseline. The extension algorithm was applied only to blocks of target and source length 1 producing one-to-many translations, e.g. the blocks b_1 and b_2 in Figure 6. The BLEU score improved to 0.166 with a block set of 3.10 million blocks. It seems to be important to carry out the block extension also for larger blocks.

We also ran the N2 system on the June 2002 DARPA TIDES Large Data evaluation test set. Six research sites and four commercial off-the-shelf systems were evaluated in Large Data track. A majority of the systems were phrase-based translation systems. For comparison with other sites, we quote the NIST score (Doddington, 2002) on this test set: the N2 system scores 7.56 whereas the official top two systems scored 7.65 and 7.34 respectively.

Table 5: Effect of the maximum phrase length on the BLEU score. Both target and source phrase are shorted than the maximum. The unigram threshold is $N(b) \geq 2$.

maximum phrase length	# blocks selected	BLEUr4n4
8	6.18 M	0.183 ± 0.01
7	5.60 M	0.182 ± 0.01
6	4.97 M	0.182 ± 0.01
5	4.25 M	0.179 ± 0.01
4	3.40 M	0.178 ± 0.01
3	2.34 M	0.167 ± 0.01
2	1.07 M	0.150 ± 0.01
1	0.16 M	0.118 ± 0.01

Table 6: Effect of the extension scheme $\mathcal{P}_{\omega,\delta}$ on the AE translation experiments.

Scheme	# blocks $N(b) \geq 1$	# blocks $N(b) \geq 3$	BLEUr3n4
$\mathcal{P}_{0,0}$	79.0 M	6.79 M	0.209 ± 0.03
$\mathcal{P}_{1,1}$	96.6 M	8.29 M	0.223 ± 0.03
$\mathcal{P}_{1,2}$	113.16 M	9.87 M	0.232 ± 0.03

4.2 Arabic-English Experiments

We also carried out experiments for the translation direction Arabic to English using training data from UN documents. For testing, we use a test set of 177 sentences with $5,826$ words arranged in 19 documents The training data contains 122.0 million Arabic and 95.8 million English words. The training data is pre-processed using some morphological analysis. For the Arabic experiments, we have tested the 3 extension schemes $\mathcal{P}_{0,0}$, $\mathcal{P}_{1,1}$, and $\mathcal{P}_{1,2}$ as shown in Table 6. Here, the results for the different schemes differ significantly and the $\mathcal{P}_{1,2}$ scheme produces the best results. For the AE experiments, only blocks up to a phrase length of 6 are computed due to disk memory restrictions. The training data is split into several chunks of $300,000$ training sentence pairs each, and the final block set together with the unigram count is obtained by merging the block

[5] We cannot compute the block set resulting from all word link quadruples in $\mathcal{R}$, which is much bigger, due to CPU and memory restrictions.

files for each of the chunks written onto disk memory. The word-to-word alignment is trained using 5 iterations of the IBM Model 1 training followed by 5 iterations of the HMM Viterbi training. This training procedure takes about a day to execute on a single machine. Additionally, the overall block selection procedure takes about 2.5 hours to execute.

5 Previous Work

Block-based translation units are used in several papers on statistical machine translation. (Och et al., 1999) describe the alignment template system for statistical MT: alignment templates correspond to blocks that do have an internal structure. Marcu and Wong (2002) use a joint probability model for blocks where the clumps are contiguous phrases as in this paper. Yamada and Knight (2002) presents a decoder for syntax-based MT that uses so-called phrasal translation units that correspond to blocks. Block unigram counts are used to filter the blocks. The phrasal model is included into a syntax-based model. Projection of phrases has also been used in (Yarowsky et al., 2001). A word link extension algorithm similar to the one presented in this paper is given in (Koehn et al., 2003).

6 Conclusion

In this paper, we describe a block-based unigram model for SMT. A novel block learning algorithm is presented that extends high-precision interval projections by elements from a high-recall alignment. The extension method is shown to improve translation performance significantly. For the Chinese-to-English task, we obtained a NIST score of 7.56 on the June 2002 DARPA TIDES Large Data evaluation test set.

Acknowledgements

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 Fei Xia and Kishore Papineni.

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.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. of the Second International Conference of Human Language Technology Research*, pages 138–145, March.

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, Alberta, Canada, May.

Daniel Marcu and William Wong. 2002. A Phrased-Based, Joint Probability Model for Statistical Machine Translation. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP 02)*, pages 133–139, Philadelphia, PA, July.

Franz-Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *Proc. of the 38th Annual Meeting of the Association of Computational Linguistics (ACL 2000)*, pages 440–447, Hong-Kong, China, October.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of machine translation. In *Proc. of the 40th Annual Conf. of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, July.

Christoph Tillmann and Hermann Ney. 2003. Word Reordering and a DP Beam Search Algorithm for Statistical Machine Translation. *Computational Linguistics*, 29(1):97–133.

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM BasedWord Alignment in Statistical Machine Translation. In *Proc. of the 16th Int. Conf. on Computational Linguistics (COLING 1996)*, pages 836–841, Copenhagen, Denmark, August.

Kenji Yamada and Kevin Knight. 2002. A Decoder for Syntax-based Statistical MT. In *Proc. of the 40th Annual Conf. of the Association for Computational Linguistics (ACL 02)*, pages 303–310, Philadelphia, PA, July.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis tools via Robust Projection across Aligned Corpora. In *Proc. of the HLT 2001 conference*, pages 161–168, San Diego, CA, March.