To appear in Proc. FSMNLP 2008 – Finite-State Methods and Natural Language Processing, Ispra, Lago Maggiore, Italy, September 2008

Large-Scale Statistical Machine Translation with Weighted Finite State Transducers

Graeme Blackwood, Adrià de Gispert, Jamie Brunning, and William Byrne

Machine Intelligence Laboratory
Department of Engineering, Cambridge University
Trumpington Street, Cambridge, CB2 1PZ, U.K.
{gwb24,ad465,jjjb2,wjb31}@cam.ac.uk

Abstract. The Cambridge University Engineering Department phrase-based statistical machine translation system follows a generative model of translation and is implemented by the composition of component models of translation and movement realised as Weighted Finite State Transducers. Our flexible architecture requires no special purpose decoder and readily handles the large-scale natural language processing demands of state-of-the-art machine translation systems. In this paper we describe the CUED system's participation in the NIST 2008 Arabic-English machine translation evaluation task.

Key words: Statistical machine translation, weighted finite state transducers, large-scale natural language processing, finite state grammars.

1 Introduction

In the source-channel model of statistical machine translation [1], target sentences are viewed as source sentences that have passed through a noisy communication channel corrupting their surface form. The task of translation is to recover the source sentence that generated the observed target. The search for the best source sentence $\boldsymbol{S} = s_1, s_2, \ldots, s_I$ for a given target $\boldsymbol{T} = t_1, t_2, \ldots, t_J$ is typically inverted and decomposed as

$$\widehat{\boldsymbol{S}} = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, P(\boldsymbol{S}|\boldsymbol{T}) = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, P(\boldsymbol{T}|\boldsymbol{S})P(\boldsymbol{S})\,, \tag{1}$$

where $P(\boldsymbol{T}|\boldsymbol{S})$ is the translation probability, $P(\boldsymbol{S})$ is the language model probability, and the argmax denotes the search for the best translation $\boldsymbol{S}$.

The Cambridge University Engineering Department statistical machine translation system follows the Transducer Translation Model (TTM) [2, 3], a phrase-based generative model of translation that applies a series of transformations specified by conditional probability distributions and encoded as Weighted Finite State Transducers [4]. The main advantages are modularity, which facilitates the development and evaluation of individual components, and implementation simplicity, which allows us to focus on modelling issues rather than complex decoding and search algorithms. The TTM scales naturally to very large data sets and no special-purpose decoder is required; by this we mean that standard

WFST operations such as weighted composition can be used to obtain the 1-best translation or a lattice of alternative hypotheses. Finally, our system architecture readily supports speech translation, in which input ASR lattices can be translated in the same way as text [5].

2 The Transducer Translation Model

Under the Transducer Translation Model, the generation of target language sentence $\boldsymbol{T} = t_1^J$ starts with the generation of a source language sentence $\boldsymbol{S} = s_1^I$ by the source language model $P_G(s_1^I)$. Next, the source language sentence is segmented into phrases according to the unweighted uniform source phrasal segmentation model $P_W(u_1^K, K|s_1^I)$. This source phrase sequence generates a reordered target language phrase sequence according to the phrase translation and reordering model $P_R(x_1^K|u_1^K)$. Next, target language phrases are inserted into this sequence according to the insertion model $P_\Phi(v_1^R|x_1^K, u_1^K)$. Finally, the sequence of reordered and inserted target language phrases are transformed to word sequences t_1^J under the unweighted target phrasal segmentation model $P_\Omega(t_1^J|v_1^R)$. These component distributions together form a joint distribution over the source and target language sentences and their possible intermediate phrase sequences as $P(t_1^J, v_1^R, x_1^K, u_1^K, s_1^I)$.

In translation under the generative model, we start with the target sentence $\boldsymbol{T}$ in the foreign language and then search for the best source sentence $\widehat{\boldsymbol{S}}$. Encoding each distribution as a WFST leads to a model of translation as the series of compositions

$$L = G \circ W \circ R \circ \Phi \circ \Omega \circ T\,, \tag{2}$$

in which T is an acceptor for the target language word sequence and L is the word lattice of source language translations obtained during decoding. There is a direct correspondence between each distribution and the transducer in which it is realized (denoted by the distribution subscripts). The most likely translation $\widehat{\boldsymbol{S}}$ is then the path in L with least cost (i.e. the minimum negative log-likelihood in the tropical semiring).

2.1 Phrase Reordering Transducers

The TTM reordering model is implemented by means of a phrase jump transducer, typically combined through composition with the one-state phrase translation WFST. In qualitative terms, this reordering model describes a jump sequence associated with each admissible permutation of the phrases [2]. In practice, it takes input source phrase sequences and outputs their translations in both monotonic and non-monotonic order.

In the simplest reordering model, known as MJ1-Flat, two adjacent phrases are allowed to swap positions with a fixed jump probability β_1 that is determined empirically. Figure 1 shows the WFST reordering transducer for the two phrases x_1 and x_2. This simple model is effective since it significantly broadens the search

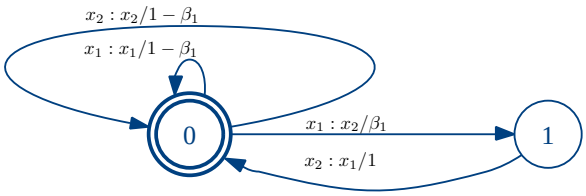

Fig. 1. The MJ1-Flat reordering transducer for a sequence of two phrases "x_1 x_2" with a fixed jump probability of β_1.

space and, as source phrases can be arbitrarily long, individual words may move quite far in translation. However, it makes no distinction as to which phrases are more likely to be reordered in translation. This problem can be addressed by defining a separate jump probability $\beta_1(v_k, u_k)$ for each phrase pair. The probabilities can be estimated from word alignments by examining adjacent phrase pairs and their orientation with respect to (v_k, u_k) and computing relative frequency estimates, in a similar fashion to Tillmann [6]. The actual WFST implementation is analogous to MJ1-Flat, but a new state is required for each phrase bigram, since the jump probability differs in each case.

2.2 Phrase Segmentation Transducers

In first-pass TTM translation all phrasal segmentations of a sentence are considered equally likely. The segmentation transducers are therefore unweighted and simply provide a mapping between the words and phrases of source and target language sentences. On the source side, the source language segmentation transducer W maps a source language word string to a lattice of all possible phrasal segmentations using the phrases of the phrase pair inventory. For example, if an acceptor for the source string "*exhibition of students returning from abroad*" is composed with the source language segmentation transducer, the result is the lattice of phrases shown in Figure 2. A similar segmentation process is applied to the target language sentence using the target language segmentation transducer Ω. The resulting lattice of phrases is the input to the decoding process in the TTM. Our flexible model architecture is such that additional inputs can be easily incorporated. For example, it may be useful to include alternative Arabic morphological analyses, variant Chinese character segmentations, or a lattice of recognition hypotheses output by an ASR system.

2.3 Language Model Acceptor

The language model $P_G(s_1^I)$ is encoded as weighted finite state acceptor G. The topology of this acceptor is such that states encode histories and arcs specify

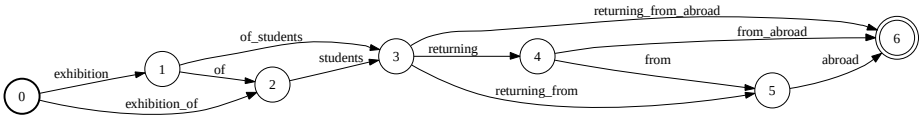

Fig. 2. Phrase lattice encoding all possible segmentations of the source language string "*exhibition of students returning from abroad*" using only the phrases of the phrase pair inventory. The phrase label on each arc shows the constituent words of the phrase.

the n-gram conditional probability of the labelled word given the history, or the context-specific backoff weight when there is no matching word. In first-pass translation we use the offline approximation in which backoff is implemented via epsilon transitions [7]. Prior to decoding, a filtering procedure is used to generate individual sentence-specific WFST language model acceptors for each sentence to be translated. This significantly improves decoding efficiency and is possible because the words which might be postulated in translation are determined by the target language input sentence and the contents of the phrase pair inventory.

2.4 Finite State Grammars for Source Language Subsequences

It often happens that the system is presented with mixed text to translate, for example ASCII characters appearing in Chinese or Arabic text, as in the following example taken from a Chinese-to-English translation task consisting of mixed text extracted from web pages:

此外，大约 三十 个 摊位 也 以 各类 行动 电视 手机 如 t-dmb (terrestrial digital media broadcasting)，
s-dmb (satellite digital multimedia broadcasting) 及 dvb-h (digital video broadcasting-handhelds)，
提供 杜林 冬 运 现场 实况 转播 的 画面，藉 以 吸引 参观 者 注意 .

The source text in such sentences should be 'translated' without change, i.e. it should pass through the translation system intact. One solution is to segment the target sentences, translate only the target language portions, and then to form a complete translation by concatenation. However, segmentation is not ideal since it prevents long-span translation and language models from looking across segmentation boundaries. To avoid this problem, a source language acceptor can be included which ensures that the desired segments appear correctly in the translation. For example, suppose two source phrases u_1 and u_2 are found in the target sentence. The acceptor would then accept sequences $V^* \cdot u_1 \cdot V^* \cdot u_2 \cdot V^*$, where V is the source language vocabulary. If degenerate translations for the source phrases are added to the translation and reordering transducer, this acceptor can be included in the translation pipeline as the last step before composition with the English language model. In this way all translations produced (including lattices) have the desired subsequences in the correct order, and all translation

scores are based on long-span translation and language model likelihoods. This is a straightforward method to impose many useful constraints in translation, such as ensuring parentheses and quotes are correctly matched, names are correctly transliterated, etc.

2.5 Minimum Error Rate Training

Minimum error rate training under BLEU [8, 9] can be used to adjust multiplicative scale factors applied to the component transducers which together make up the TTM. Although only a small number of parameters are adjusted - typically one parameter per component model or distribution - MET can be very effective in tuning systems to domain-specific development sets.

In the systems described here, MET is applied to adjust the lexical language model scale factor, word and phrase insertion penalties, phrase reordering scale factor, phrase insertion scale factor, u-to-v translation model scale factor, v-to-u translation model scale factor, and three phrase pair count features. The phrase-pair count features track whether each phrase-pair occurred once, twice, or more than twice in the parallel text [10].

MET parameter search procedures as described by Och [9] are now widely used; the only difficulty in apply them to WFSTs is to extract the contribution of each component transducer to the overall translation log likelihood. For this, we use encoded transducers as described by Roark et al. [11, 2, 12] and implemented in the OpenFST libraries [13].

3 Lattice Rescoring

This section describes lattice rescoring techniques applied to the translation output produced by the first-pass MET baseline system. Apart from MBR (section 3.4) which requires n-best lists, these operations could be applied in first-pass translation; however, we apply these techniques in rescoring subsequent to pruning of the first-pass lattices.

3.1 Large Language Model Rescoring

We apply a second-pass language model that is able to effectively utilise very large quantities of monolingual training text. Large memory and considerable time is required for the estimation of zero cutoff higher-order n-gram language models, typically necessitating partitioning of data and multiple rounds of paired interpolation to produce the final model. An alternative is to build sentence-specific language models. Firstly, counts are gathered for each training text and merged to form a single large counts file. The vocabulary used during the counting process is determined by the set of English words covering the phrases found in the parallel text. There are no cut-offs, so all observed n-grams are included in the model. Sentence-specific counts are obtained by filtering according to the vocabulary of English n-grams in each lattice. The resulting filtered counts are

then used to generate sentence-specific language models with "stupid backoff" smoothing [14] in which n-gram scores are defined as

$$S(s_i|s_{i-n+1}^{i-1}) = \begin{cases} \frac{f(s_{i-n+1}^{i})}{f(s_{i-n+1}^{i-1})} & \text{if } f(s_{i-n+1}^{i}) > 0 \\ \alpha S(s_i|s_{i-n+2}^{i-1}) & \text{otherwise} \end{cases} \tag{3}$$

The backoff weight α is the same for each order and the recursion ends with the unigram maximum likelihood estimate.

3.2 Phrasal Segmentation Model Rescoring

Phrasal segmentation models define a mapping from the words of a sentence s_1^I to sequences of translatable phrases u_1^K. Sentences cannot be segmented arbitrarily: the space of possible segmentations is constrained by the contents of the phrase table and contains only those translatable phrases found in the parallel text. We define a probability distribution over phrase sequences and estimate the model parameters from naturally occurring sequences of phrases in a large monolingual source-language training corpus. An order-n phrasal segmentation model assigns a probability to a phrase sequence u_1^K according to

$$P(u_1^K|K, s_1^I) = \prod_{k=1}^{K} P(u_k|u_1^{k-1}, K, s_1^I) \tag{4}$$

$$\approx \begin{cases} C(K, s_1^I) \prod_{k=1}^{K} P(u_k|u_{k-n+1}^{k-1}) & \text{if } u_1^K = s_1^I \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

with the additional constraint that each u_k must be a phrase with a known translation. For a fixed s_1^I, the normalisation term $C(K, s_1^I)$ can be calculated. In translation, however, the s_1^I are not fixed so we use the unnormalised likelihoods as scores. The phrase n-gram parameters of equation (5) are estimated from the frequencies of occurrence of phrase sequences in the training text. Standard discounting and context-dependent backoff [15] are applied to smooth the maximum likelihood estimates.

The word lattice L produced during first-pass translation is composed with unweighted transducer W to obtain a lattice of phrases ($L \circ W$); this lattice contains phrase sequences and translation scores consistent with the first-pass translation. We now wish to apply the phrase segmentation model distribution of equation (5) to this phrase lattice. The conditional probabilities and backoff structure are encoded as weighted finite state acceptor Ψ in the same way as for a regular word language model [7]. The phrasal segmentation model acceptor is then composed with the phrase lattice and projected on the input to obtain the rescored word lattice:

$$L' = (L \circ W) \circ \Psi \,. \tag{6}$$

The most likely translation after phrasal segmentation model rescoring is given by the path in L' with least cost.

3.3 Model-1 Lattice-to-String Alignment Scores

IBM Model-1 is a simple model of word alignment used in parallel text alignment. Model-1 is not powerful enough to be used alone for translation, but can be used to rank competing translation hypotheses produced by more powerful systems. Introducing a variable a_j which denotes the alignment of t_j in t_1^J to s_{a_j} in s_1^I, the Model-1 alignment distribution is

$$P_{M1}(t_1^J, a_1^J, J|s_1^I) = P_L(J|I)\frac{1}{I^J}\prod_{j=1}^{J} p_T(t_j|s_{a_j})\,. \tag{7}$$

The model is such that the maximum likelihood alignment

$$\max_{a_1^J} P_{M1}(t_1^J, a_1^J, J|s_1^I)\,, \tag{8}$$

is readily found via dynamic programming. It is also straightforward to find, for a fixed target sentence t_1^J, the most likely alignment of every translation hypothesis s_1^I in a lattice L, i.e. to simultaneously find the best alignment of every lattice path to the target string. We refer to this as Model-1 lattice-to-string alignment. Of course this could be done by expanding the lattice into a list of distinct hypotheses and aligning each to the target string; however lattice-to-string alignment is faster and retains the compact lattice representation of hypotheses. However, as discussed by Knight and Al-Onaizan [16], this process cannot be implemented easily with WFSTs. In adding Model-1 alignment scores to the TTM translation lattices, we therefore depart from the WFST formalism and add the Model-1 likelihoods to the TTM lattice scores with non-WFST based lattice-to-string alignment procedures.

3.4 Minimum Bayes Risk Decoding

The final step in translation is Minimum Bayes Risk decoding (MBR) which searches for a hypothesis to minimise the expected loss of translation errors under loss functions that measure translation performance. The rationale is to reconcile estimation criteria (e.g. maximum likelihood) with translation criteria (e.g. BLEU). Since the goal is to maximise the BLEU score, the loss is the negative sentence level BLEU score [17]. Exact computation of statistics needed for BLEU cannot easily be done over lattices, or with finite state approaches, so each translation lattice is expanded into a list of translation hypotheses $\mathcal{N}$ with posterior scores, and the hypothesis is selected which has the least risk relative to the collection of other hypotheses:

$$\widehat{\boldsymbol{S}} = \operatorname*{argmin}_{\boldsymbol{S}\in\mathcal{N}} \sum_{\boldsymbol{S}'\in\mathcal{N}} -\mathrm{BLEU}(\boldsymbol{S}', \boldsymbol{S})P(\boldsymbol{S}'|\boldsymbol{T})\,. \tag{9}$$

4 System Development

We describe experiments on the NIST Arabic-English translation task. The development set *mt02-05-tune* is formed from the odd numbered sentences of the NIST MT02 through MT05 evaluation sets; the even numbered sentences form the validation set *mt02-05-test*. Test performance is evaluated using the NIST subsets from the MT06 evaluation: *mt06-nist-nw* for newswire data and *mt06-nist-ng* for newsgroup data. We also report results for the NIST MT08 evaluation. Each set contains four references and BLEU scores are computed for lower-case translations.

The TTM baseline system is trained using all of the available Arabic-English data for the NIST MT08 evaluation. In first-pass translation, decoding proceeds with a 4-gram language model estimated over the parallel text and a 965 million word subset of monolingual data from the English Gigaword Third Edition. Minimum error rate training under BLEU optimises the decoder feature weights using the development set *mt02-05-tune*. In the second pass, a 5-gram zero-cutoff stupid-backoff language model estimated using approximately 4.7 billion words of English newswire text is used to rescore the first-pass lattices. The phrasal segmentation model parameters are trained using a 1.8 billion word subset of the same monolingual training data used to build the second-pass word language model. Further post-processing steps incorporate the Model-1 lattice-to-string alignment scores and MBR.

4.1 Results and Discussion

Table 1 shows translation performance for each of the various development and evaluation sets as measured by BLEU and TER[1]. All of the results in the table were obtained using the MJ1 reordering model with orientation probabilities estimated from alignments. The 1-best output obtained from the lattices after minimum error rate training results in the scores shown in the row labelled 'TTM+MET'. These lattices are then rescored by each of the post-processing techniques described in section 3, resulting in significant improvements across all sets. While large gains of between 1.5 and 2.7 BLEU points are observed after 5-gram rescoring (row labelled '+5g'), phrase segmentation model rescoring results in more modest improvements (row labelled '+PSM'). However, these gains are interesting since the models are trained on a subset of the same monolingual data used to train the 5-gram word language model, suggesting that some degree of useful complementary information has been captured by the phrasal segmentation models. The final post-processing step (row labelled '+MBR') shows the results obtained after rescoring the 1000-best list for each sentence using minimum Bayes risk decoding.

In order to demonstrate the advantage of estimating the phrase-specific β_1 reordering probabilities, Table 2 shows translation scores when a flat distribution

[1] Full MT08 results are available at http://www.nist.gov/speech/tests/mt/2008/. It is worth noting that many of the top entries make use of system combination; the results reported here are for single system translations.

Table 1. Arabic-English translation results (lower-cased BLEU / TER) for best performing system configuration using phrase pair count features and β_1 probabilities estimated from the alignments.

Method	mt02-05-tune	mt02-05-test	mt06-nist-nw	mt06-nist-ng	mt08-nist
TTM+MET	50.9 / 42.8	50.3 / 43.3	48.1 / 44.3	37.5 / 53.5	43.1 / 49.5
+5g	53.5 / 41.8	52.4 / 42.4	49.6 / 43.9	39.0 / 54.0	43.7 / 49.3
+PSM	53.9 / 42.1	53.3 / 42.7	50.1 / 44.3	39.0 / 54.7	44.3 / 49.3
+MBR	54.0 / 41.7	53.7 / 42.2	51.0 / 43.9	39.4 / 54.1	45.0 / 48.9

over all phrase pairs is applied, i.e. the MJ1-Flat reordering model described in section 2.1. These results show that there is a degradation of around 0.4 BLEU points in the MET results, and this degradation is seen throughout the subsequent rescoring steps. A more informed phrase reordering model produces a higher quality MET lattice for rescoring. Therefore, we expect that further improvements in the reordering model will be complementary and benefit even more from our large language model rescoring techniques. However, preliminary experiments with a simplified MJ2 reordering did not yield significant improvements for this Arabic-English translation task and so are not reported here.

Table 2. Arabic-English translation results (lower-cased BLEU / TER) without estimation of the β_1 orientation probabilities for the MJ1 reordering model (MJ1-Flat).

Method	mt02-05-tune	mt02-05-test
TTM+MET	50.4 / 43.3	50.0 / 43.8
+5g	53.0 / 42.2	52.2 / 42.8
+PSM	53.4 / 42.5	53.1 / 43.1

To conclude our analysis of the contribution of each system component, Table 3 shows results obtained when the phrase pair count features are not included in MET. The phrase pair count features clearly contribute significantly to the generation of higher quality first-pass lattices since there is a degradation of between 1.7 and 2.0 BLEU points with respect to the baseline system.

Table 3. Translation results (lower-cased BLEU / TER) without phrase pair count features. Two different lattice rescoring orders are compared. On the left, Model-1 (MOD1) rescoring precedes 5-gram and PSM rescoring. On the right, Model-1 rescoring is performed as the final step.

Method	mt02-05-tune	mt02-05-test	Method	mt02-05-tune	mt02-05-test
TTM+MET	48.9 / 43.8	48.6 / 44.1	TTM+MET	48.9 / 43.8	48.6 / 44.1
+MOD1	50.5 / 42.5	50.4 / 43.0	+5g	51.5 / 42.2	51.5 / 42.7
+5g	52.2 / 41.6	52.1 / 42.3	+PSM	52.6 / 42.3	52.6 / 42.7
+PSM	52.9 / 41.9	52.6 / 42.6	+MOD1	53.0 / 41.8	52.6 / 42.5

Table 3 also compares the application of Model-1 rescoring at two different stages in the translation pipeline. Model-1 rescoring proves especially beneficial when directly rescoring the MET lattice (with an improvement of up to 1.8 BLEU points). However, if Model-1 rescoring is applied after 5-gram and phrase segmentation model rescoring there are no real improvements. Two conclusions may be drawn from this. Firstly, that each of the rescoring techniques are a useful source of information when rescoring lattices, and, secondly, applying these techniques sequentially to the same MET lattice does not always provide gains. This suggests it is important to integrate these information sources directly in minimum error rate training prior to generating the lattice.

4.2 Efficiency Considerations

Large-scale statistical machine translation is computationally intensive and an efficient implementation is crucial. To tackle this, we carefully build separate WFSTs that only include the model parameters relevant to each input sentence by prior inspection of input phrases and the general phrase inventory. Using the MJ1 reordering model, the memory required during decoding is less than ∼4Gb for most sentences in the reported tasks.

For the longest input sentences, which can exceed 100 words in length, memory requirements may grow beyond this limit and this necessitates pruning. Several pruning strategies may be used, such as standard cost-based pruning for the translation WFST prior to composition with the language model.

However, experience shows that better results are achieved by selecting, for those sentences with a number of states in the translation WFST (prior to language model composition) above a certain threshold, only those phrase segmentations that match the number of phrases in the minimum-number-of-phrases segmentation. This favours segmentations with longer phrases and limits memory requirements without any significant change in translation performance. For *mt02-05-tune* only 39 of the 2075 sentences are affected.

Without further pruning, our system translates the *mt02-05-tune* set (2075 sentences, ∼60k words) in a total time of 420 minutes and this can be accomplished in very reasonable time by parallelisation. Figure 3 shows translation time per input word as a function of the sentence length.

As the graph shows, the longer the input sentence, the longer it takes to translate each word. By applying our pruning strategy, we ensure that translation time does not exceed an average rate of 0.43 seconds per word even for the longest sentences. However, around 80% of the sentences are of 40 words or less in length and these are translated with a much quicker rate of 0.30 seconds per word.

5 Summary and Future Work

We have described the Cambridge University Engineering Department statistical machine translation system that formulates translation as a series of transformations encoded in weighted finite state transducers and decodes using standard

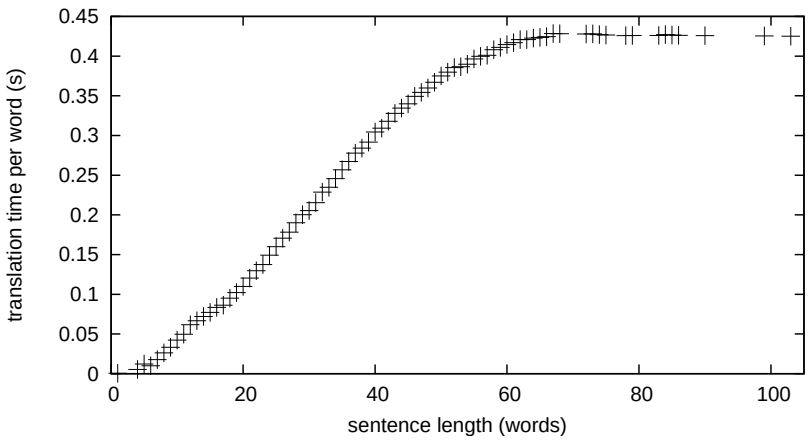

Fig. 3. Translation time per word as a function of sentence length for *mt02-05-tune*.

finite state operations and algorithms. The system is able to handle very large quantities of data efficiently and effectively and achieves good performance on the 2008 NIST Arabic-English machine translation task, even with the relatively simple MJ1 reordering model.

Future work will investigate whether larger and more consistent gains are possible by integrating the phrasal segmentation models and Model-1 rescoring directly into the MET baseline system. It is also interesting to consider more flexible phrase reordering models by allowing jumps of more than one phrase, although this can lead to a very large search space with many unnecessary hypotheses [2]. One possible solution is to only allow such jumps for a particular list of phrase pairs observed to occur with long-range reorderings in the parallel text from which the phrases are extracted.

Acknowledgements. This work was supported in part under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

References

1. Brown, P.F., Pietra, S.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2) (1994) 263–311
2. Kumar, S., Byrne, W.: Local phrase reordering models for statistical machine translation. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. (2005) 161–168
3. Kumar, S., Deng, Y., Byrne, W.: A weighted finite state transducer translation template model for statistical machine translation. Natural Language Engineering **12**(1) (2006) 35–75

4. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. In: Computer Speech and Language. Volume 16. (2002) 69–88
5. Mathias, L., Byrne, W.: Statistical phrase-based speech translation. In: 2006 IEEE International Conference on Acoustics, Speech and Signal Processing. (2006)
6. Tillmann, C.: A unigram orientation model for statistical machine translation. In: HLT-NAACL 2004: Short Papers, Boston, Massachusetts, USA, Association for Computational Linguistics (May 2 - May 7 2004) 101–104
7. Allauzen, C., Mohri, M., Roark, B.: Generalized algorithms for constructing statistical language models. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics. (2003) 557–564
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Meeting of the Association for Computational Linguistics, Morristown, NJ, USA (2001) 311–318
9. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, Morristown, NJ, USA (2003) 160–167
10. Bender, O., Matusov, E., Hahn, S., Hasan, S., Khadivi, S., Ney, H.: The RWTH Arabic-to-English spoken language translation system. In: Proceedings of the 2007 Automatic Speech Understanding Workshop. (2007) 396–401
11. Roark, B., Saraclar, M., Collins, M.: Discriminative n-gram language modeling. Computer Speech and Language **21**(2) (2007) 373–392
12. Mathias, L.: Statistical Machine Translation and Automatic Speech Recognition under Uncertainty. Dissertation, Johns Hopkins University (2007)
13. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFST: a general and efficient weighted finite-state transducer library. In: Proceedings of the 9th International Conference on Implementation and Application of Automata, Springer (2007) 11–23
14. Brants, T., Popat, A.C., Xu, P., Och, F.J., Dean, J.: Large language models in machine translation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. (2007) 858–867
15. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In Joshi, A., Palmer, M., eds.: Proceedings of the 34th Meeting of the Association for Computational Linguistics, San Francisco, Morgan Kaufmann Publishers (1996) 310–318
16. Knight, K., Al-Onaizan, Y.: Translation with finite-state devices. In: AMTA '98: Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas, London, UK, Springer-Verlag (1998) 421–437
17. Kumar, S., Byrne, W.: Minimum Bayes-risk decoding for statistical machine translation. In: HLT-NAACL 2004, Boston, Massachusetts, USA, Association for Computational Linguistics (May 2 - May 7 2004)