SyMGiza++: Symmetrized Word Alignment Models for Statistical Machine Translation

Marcin Junczys-Dowmunt, Arkadiusz Szał

Faculty of Mathematics and Computer Science
Adam Mickiewicz University
ul. Umultowska 87, 61-614 Poznań, Poland
{junczys,arekszal}@amu.edu.pl

Abstract. SyMGiza++ — a tool that computes symmetric word alignment models with the capability to take advantage of multi-processor systems — is presented. A series of fairly simple modifications to the original IBM/Giza++ word alignment models allows to update the symmetrized models between chosen iterations of the original training algorithms. We achieve a relative alignment quality improvement of more than 17% compared to Giza++ and MGiza++ on the standard Canadian Hansards task, while maintaining the speed improvements provided by the capability of parallel computations of MGiza++.
Furthermore, the alignment models are evaluated in the context of phrase-based statistical machine translation, where a consistent improvement measured in BLEU scores can be observed when SyMGiza++ is used instead of Giza++ or MGiza++.

1 Introduction

Word alignment is a key component of the training procedure for statistical machine translation systems. The classic tool used for this task is Giza++ [1] which is an implementation of the so-called IBM Models 1-5 [2], the HMM model by [3] and its extension by [1], and Model 6 [1].

All these models are asymmetric, i.e. for a chosen translation direction, they allow for many-to-one alignments, but not for one-to-many alignments. Training two models in opposite directions and symmetrizing the resulting word alignments is commonly employed to improve alignment quality and to allow for more natural alignments. The two alignment models are trained fully independently from each other. Symmetrization is then performed as a post-processing step. Previous work [4, 5] has shown that the introduction of symmetry during training results in better alignment quality than post-training symmetrization.

The approaches from [4, 5] as well as our method still require the computation of two directed models which use common information during the training. Employing a multi-processor system for the parallel computation of theses models is a natural choice. However, Giza++ was designed to be single-process and single-thread. MGiza++ [6] is an extension of Giza++ which allows to start multiple threads on a single computer.

We therefore choose to extend MGiza++ with the capability to symmetrize word alignments models to tackle both problems in one stroke. The resulting tool SyMGiza++ is described in this work.[1] The paper will be organized as follows: Section 2 provides a short overview of Giza++ and MGiza++ and the above mentioned methods of symmetrized alignment model training. In Sec. 3 we give a formal description of our modifications introduced into the classical word alignment models implemented in Giza++ and MGiza++. The evaluation methodology and results are provided in Sec. 4. Section 4 is divided into two parts: in the first part we give results for alignment quality alone, the second part deals with the influence of the improved alignment method on machine translation results. Finally, conclusions are presented in Sec. 5.

2 Previous Work

2.1 Giza++ and MGiza++

Giza++ implements maximum likelihood estimators for several statistical alignment models, including Model 1 through 5 described by [2], a HMM alignment model by [3] and Model 6 from [1]. The EM [7] algorithm is employed for the estimation of the parameters of the models. During the EM algorithm two steps are applied in each iteration: in the first step, the E-step, the previously computed model or a model with initial values is applied to the data. The expected counts for specific parameters are collected using the probabilities of this model. In the second step, the M-step, these expected counts are taken as fact and used to estimate the probabilities of the next model. A correct implementation of the E-step requires to sum over all possible alignments for one sentence pair. This can be done efficiently for Model 1 and 2, and using the Baum-Welch algorithm also for the HMM alignment model [1].

For Models 3 through 6, a complete enumeration of alignments cannot be accomplished in a reasonable time. This can be approximated by using only a subset of highly scored alignments. In [2] it has been suggested to use only the alignment with the maximum probability, the so-called Viterbi alignment. Another approach resorts to the generation of a set of high probability alignments obtained by making small changes to the Viterbi alignment. [8] proposed to use the neighbour alignments of the Viterbi alignment.

MGiza++ [6] is a multi-threaded word alignment tool that utilizes multiple threads to speed up the time-consuming word alignment process. The implementation of the word alignment models is based on Giza++ and shares large portions of source code with Giza++. The main differences rely on multiple thread management and the synchronization of the counts collecting process. Similarly, our tool in turn incorporates large portions of the MGiza++ source code extending MGiza++'s capabilities of using multiple processors with the ability to compute symmetrized word alignment models in a multiprocessor environment. Since the multiprocessing aspect is mainly a feature of the original

[1] SyMGiza++ is available at http://psi.amu.edu.pl/en/index.php?title=Downloads

MGiza++, we will not discuss it in this paper and refer the reader to the original paper on MGiza++ [6].

2.2 Symmetrized Word Alignment Models

The posteriori symmetrization of word alignments has been introduced by [1]. This method does not compute symmetrized word alignment models during the training procedure, but uses heuristic combination methods after the training. We described it in more detail in Sec. 3.5. The best results of [1] for the Hansards task are 9.4% AER (using Model 4 in the last training iterations) and 8.7% AER (using the more sophisticated Model 6).

[4] improve the IBM alignment models, as well as the Hidden-Markov alignment model using a symmetric lexicon model. This symmetrization takes not only the standard translation direction from source to target into account, but also the inverse translation direction from target to source. In addition to the symmetrization, a smoothed lexicon model is used. The performance of the models is evaluated for Canadian Hansards task, where they achieve an improvement of more than 30% relative to unidirectional training with Giza++ (7.5% AER) is achieved.

In [9], the symmetrization is performed after training IBM and HMM alignment models in both directions. Using these models, local costs of aligning a source word and a target word in each sentence pair are estimated and graph algorithms are used to determine the symmetric alignment with minimal total costs. The automatic alignments created in this way are evaluated on the German–English Verbmobil task and the French–English Canadian Hansards task (6.6% AER).

Another unsupervised approach to symmetric word alignment is presented by [5]. Two simple asymmetric models are trained jointly to maximize a combination of data likelihood and agreement between the models. The authors restrict their experiments to IBM Models 1 and 2 and a new jointly trained HMM alignment model. They report an AER of 4.9% — a 29% reduction over symmetrized IBM model 4 predictions — for the Canadian Hansards task.

3 SyMGiza++ — Symmetrized MGiza++

In this section we will describe our modifications to the well known alignment models from [2] and [1].

We do not introduce changes to the main parameter estimation procedure. Instead, we modify the counting phase of each model to adopt information provided by both directed models simultaneously. The parameter combination step is executed in the main thread. In the following subsections, the formal aspects of the parameter combination will be outlined separately for each model. The notation has been adopted from [2] and we refer the reader to this work for details on the original models that will not be repeated in this paper.

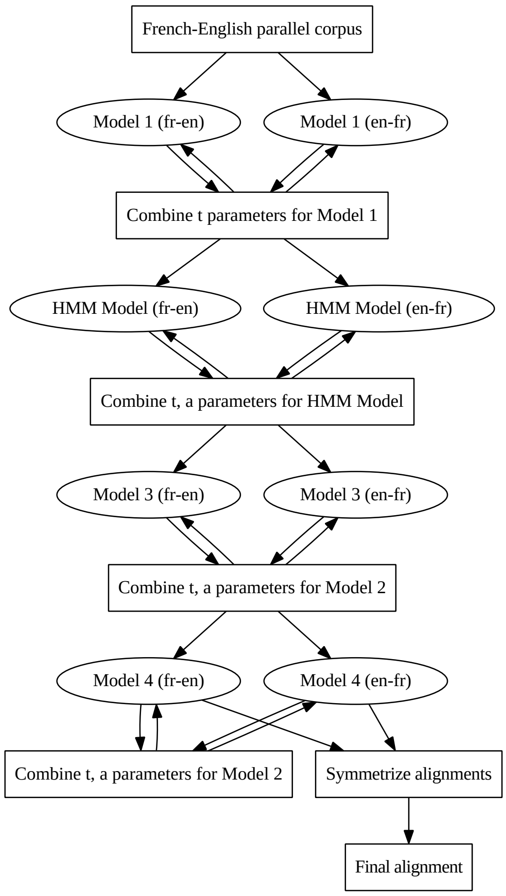

Fig. 1: General training scheme for SyMGiza++

3.1 Model 1

Model 1 is the first of the IBM models described extensively by [2] which have been implemented accurately in Giza++ and MGiza++.

In order to distinguish between the parameters of the two simultaneously computed alignment models we will use α and β as subscripts for the parameters of the first and second model respectively. For our English-French training corpus we compute the following two models:

$$Pr_\alpha(\mathbf{f}|\mathbf{e}) = \frac{\epsilon(m|l)}{(l+1)^m} \sum_{\mathbf{a}} \prod_{j=1}^{m} t_\alpha(f_j|e_{a_j}) \tag{1}$$

$$Pr_\beta(\mathbf{e}|\mathbf{f}) = \frac{\epsilon(l|m)}{(m+1)^l} \sum_{\mathbf{b}} \prod_{i=1}^{l} t_\beta(e_i|f_{b_i}) \tag{2}$$

where l and m are the lengths of the French sentence $\mathbf{f}$ and the English sentence $\mathbf{e}$ respectively, $\mathbf{a}$ and $\mathbf{b}$ are the directed alignments between the sentences and t_α and t_β the directed *translation probabilities* between the French and English words f and e. Due to the simplicity of this model, it is straightforward to introduce our changes in the counting method used during the E-step of the EM-algorithm. The only free parameters of Model 1 are the translation probabilities t_α and t_β which are estimated by:

$$t_\alpha(f|e) = \frac{\sum_{s=1}^{S} c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}{\sum_{f'} \sum_{s=1}^{S} c(f'|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}, \tag{3}$$

where S is the number of sentences in the parallel training corpus. $c(f|e; \mathbf{f}, \mathbf{e})$ is the expected count of times the words f and e form translations in the given sentences $\mathbf{f}$ and $\mathbf{e}$, in the inverted model $c(e|f; \mathbf{e}, \mathbf{f})$ is used.

In the original model, the expected counts $c(f|e; \mathbf{f}, \mathbf{e})$ are calculated from the t values of the preceding iteration with the help of the following two formulas:

$$c(f|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) \sum_{i,j} \delta(f, f_j)\delta(e, e_i), \tag{4}$$

and

$$Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \frac{\prod_{j=1}^{m} t_\alpha(f_j|e_{a_j})}{\sum_{\mathbf{a}} \prod_{j=1}^{m} t_\alpha(f_j|e_{a_j})}, \tag{5}$$

where δ is the Kronecker function[2]. Equations (3) and (4) are common for all models discussed in this section. Our modifications are restricted to (5) which is replaced by

$$\begin{aligned} Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) &= \frac{\prod_{j=1}^{m} \bar{t}(f_j, e_{a_j})}{\sum_{\mathbf{a}} \prod_{j=1}^{m} \bar{t}(f_j, e_{a_j})} \\ &= \frac{\prod_{j=1}^{m} \left(t_\alpha(f_j|e_{a_j}) + t_\beta(e_{a_j}|f_j)\right)}{\sum_{\mathbf{a}} \prod_{j=1}^{m} \left(t_\alpha(f_j|e_{a_j}) + t_\beta(e_{a_j}|f_j)\right)} \end{aligned} \tag{6}$$

Here we see the only difference between the standard Model 1 and our symmetrized version. By taking into account the translation probabilities from the previous iteration of both directed models we inform each model about the estimates of its counterparts. The following intuition applies: a French word is a good translation of an English word, if the English word is a good translation of the French word as well. This cannot be easily captured in the directed models without breaking up its sound probabilistic interpretation, as it happens here. However, since we modify only the way expected counts are obtained, the requirement imposed by [2] that

$$\sum_f t(f|e) = 1$$

[2] $\delta(i, j) = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases}$.

still applies. Our modifications do not interfere with the EM procedure. The parameters for the inverted model are obtained analogously.

It should be noted that most of the time — despite the symmetry of the sum $t_\alpha(f|e) + t_\beta(e|f)$ occurring in both counts — $c(f|e;\mathbf{f},\mathbf{e})$ and $c(e|f;\mathbf{e},\mathbf{f})$ will have different values for the same words and sentences. This is due to the differences in the alignment direction. Therefore $t_\alpha(f|e) \neq t_\beta(e|f)$ in the general case.

3.2 Model 2

Although it is common practice to replace Model 2 during the training procedure with the HMM Model described in the next subsection, we need to modify its counting procedure as well. Model 2 is used to score a subset of alignments during the training procedure of the more sophisticated Models 3 and 4 which — in contrast to the lower models — cannot efficiently enumerate all possible alignments.

Model 2 introduces a second type of free parameters: the *alignment probabilities* a. These a parameters capture the probability that given the lengths of both sentences, a French word at position j is aligned with an English word at position a_j. The complete model is given by [2] as:

$$Pr_\alpha(\mathbf{f}|\mathbf{e}) = \epsilon(m|l) \sum_{\mathbf{a}} \prod_{j=1}^{m} \big(t_\alpha(f_j|e_{a_j}) a_\alpha(a_j|j,m,l)\big) \tag{7}$$

The general scheme described in (3) and (4) for the estimation of t values is the same for Model 2 as for Model 1. The alignment probabilities are estimated similarly:

$$a_\alpha(i|j,m,l) = \frac{\sum_{s=1}^{S} c(i|j,m,l;\mathbf{f}^{(s)},\mathbf{e}^{(s)})}{\sum_{i'} \sum_{s=1}^{S} c(i'|j,m,l;\mathbf{f}^{(s)},\mathbf{e}^{(s)})}, \tag{8}$$

$$c(i|j,m,l;\mathbf{f},\mathbf{e}) = \sum_{\mathbf{a}} Pr_\alpha(\mathbf{a}|\mathbf{f},\mathbf{e}) \delta(i,a_j). \tag{9}$$

Again, we only modify $Pr(\mathbf{a}|\mathbf{f},\mathbf{e})$ in (4) and (9) to obtain our symmetrized version of the alignment models:

$$Pr_\alpha(\mathbf{a}|\mathbf{f},\mathbf{e}) = \frac{\prod_{i=1}^{m} \big(\bar{t}(f_j,e_{a_j}) \bar{a}(a_j,j,m,l)\big)}{\sum_{\mathbf{a}} \prod_{j=1}^{m} \big(\bar{t}(f_j,e_{a_j}) \bar{a}(a_j,j,m,l)\big)} \tag{10}$$

where $\bar{t}(f,e)$ is defined as before for Model 1 and $\bar{a}(i,j,m,l) = a_\alpha(i|j,m,l) + a_\beta(j|i,l,m)$. The effect of information sharing between the two inverted models Pr_α and Pr_β is even increased for Model 2 since translation and alignment probabilities interact during the estimation of both types of parameters for the next iteration.

3.3 HMM Model

The HMM Alignment Model has been introduced by [3] and is used in the Giza++ family of alignment tools as a replacement for the less effective Model 2. The HMM alignment model is given by the following formula which at first looks very similar to (7):

$$P_\alpha(\mathbf{f}|\mathbf{e}) = \epsilon(m|l) \sum_{\mathbf{a}} \prod_{j=1}^{m} \big(t_\alpha(f_j|e_{a_j}) a_\alpha(a_j|a_{j-1}, l)\big) \tag{11}$$

The alignment probabilities from Model 2, however, are replaced by a different type of alignment probabilities. Here the probability of alignment a_j for position j has a dependence on the previous alignment a_{j-1} which turns the alignment model into a first order Markov model. The counts for the new a parameter are defined as follows:

$$a_\alpha(i|i', l) = \frac{\sum_{s=1}^{S} c(i|i', l; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}{\sum_{i''} \sum_{s=1}^{S} c(i''|i', l; \mathbf{f}^{(s)}, \mathbf{e}^{(s)})}, \tag{12}$$

$$c(i|i', l; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) \sum_{j} \delta(i', a_{j-1}) \delta(i, a_j) \tag{13}$$

The definition of the t parameter and corresponding counts remains the same as for Model 1 and 2. Like before we only have to modify the definition of $Pr(\mathbf{a}|\mathbf{f}, \mathbf{e})$:

$$Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \frac{\prod_{j=1}^{m} t_\alpha(f_j|e_{a_j}) a_\alpha(a_j|a_{j-1}, l)}{\sum_{\mathbf{a}} \prod_{j=1}^{m} t_\alpha(f_j|e_{a_j}) a_\alpha(a_j|a_{j-1}, l)} \tag{14}$$

is replaced by

$$Pr_\alpha(\mathbf{a}|\mathbf{f}, \mathbf{e}) = \frac{\prod_{i=1}^{m} \big(\bar{t}(f_j, e_{a_j}) a_\alpha(a_j|a_{j-1}, l)\big)}{\sum_{\mathbf{a}} \prod_{j=1}^{m} \big(\bar{t}(f_j, e_{a_j}) a_\alpha(a_j|a_{j-1}, l)\big)}. \tag{15}$$

$\bar{t}$ is defined as before for Model 1 and 2.

Here, the alignment probabilities a remain unchanged. For Model 2 we are able to find the symmetrically calculated a parameters just by swapping source and target values. Doing the same for the Markov model would change the interpretation of the alignment probabilities. We would require neighbouring source language words to be aligned only with neighbouring target language words which is to strong an assumption. Nevertheless, their values are still influenced by both models due to the appearance of $\bar{t}$ in the re-estimation.

3.4 Model 3 and 4

We already mentioned that the parameters specific for Models 3 and 4 are calculated from fractional counts collected over a subset of alignments that have been identified with the help of the Viterbi alignments calculated by Model 2.

Therefore it is not necessary to revise the parameter estimation formulas for Model 3 and 4, instead we simply adopt the previous changes made for Model 2. This influences the parameters of Model 3 and 4 indirectly by choosing better informed Viterbi alignments during each iteration.

3.5 Final Symmetrization

Although the two directed models influence each other between each iteration, the two final alignments produced at the end of the training procedure differ due the restrictions imposed by the models. Alignments are directed and since alignments are functions, there are no one-to-many or many-to-many alignments for the respective directions. There are, however, many-to-one alignments. [1] have proposed a method for the symmetrization of alignment models, which they call *refined symmetrization* and which is reported to have a positive effect on alignment quality.

They first map each directed alignment into a set of alignment points and create a new alignment as the intersection of these two sets. Next, they iteratively add alignment points (i, j) from the union of the two sets to the newly created alignment occurring only in the first alignment or in the second alignment if neither f_j nor e_i has an alignment in the new alignment, or if both of the following conditions hold:

- The alignment (i, j) has a horizontal neighbour$(i-1, j)$, $(i+1, j)$ or a vertical neighbour $(i, j-1)$, $(i, j+1)$ that is already in the new alignment.
- Adding (i, j) to the new alignment does not created alignments with both horizontal and vertical neighbours.

This method is applied as the final step of our computation and will also be applied to the directed alignments created by Giza++ and MGiza++, our baseline systems. Final symmetrization methods are included in SyMGiza++ and can be applied without the need for external programs. Apart from the mentioned refined method it is also possible to use multiple variants of *grow-diag* featured in the Moses training procedure.

4 Evaluation

4.1 Word Alignment Quality

Evaluating word alignment quality, we compare three systems on the same training and test data: Giza++, MGiza++, and SyMGiza++. For the Giza++ and MGiza++ we run both directed models separately and in parallel and recombine the resulting final alignments with the refined method described in 3.5. We experimented with different training schemes and found the following to work best for Giza++ and MGiza++: $5 \times$ Model 1, $5 \times$ HMM Model, $3 \times$ Model 3 and $3 \times$ Model 4. This is consistent with the findings of [1] for the same training data and test set.

Table 1: Results for the HLT/NAACL 2003 test set

Alignment Method	Prec [%]	Rec [%]	AER [%]
Giza++ en-fr	91.19	92.20	8.39
Giza++ fr-en	91.82	87.96	9.79
Giza++ refined	93.24	92.59	7.02
MGiza++ en-fr	91.19	92.22	8.40
MGiza++ fr-en	91.84	87.96	9.78
MGiza++ refined	93.25	92.60	7.01
SyMGiza++	94.34	94.08	**5.76**

The training scheme for SyMGIZA++ has been determined as $5 \times$ Model 1, $5 \times$ HMM Model, $5 \times$ Model 3 and $5 \times$ Model 4. The models are symmetrized between model transitions. Using this training scheme for Giza++ or MGiza++ causes a small decline in alignment quality.

The standard metric *Alignment Error Rate* (AER) proposed by [1] is used to evaluate the quality of the introduced input word alignments. AER is calculated as follows:

$$\text{Precision} = \frac{|A \cap P|}{|A|} \qquad \text{Recall} = \frac{|A \cap S|}{|S|}$$
$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{16}$$

where P is the set of possible alignment points in the reference alignment, S is the set of sure alignments in the reference alignment ($S \subset P$), and A is the evaluated word alignment.

In order to obtain results that can be easily compared with the work summarized in 2.2, we evaluated our system on the Canadian Hansards task made available during the HLT-NAACL 2003 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond" [10]. The training data comprises 1.1M sentences from the Canadian Hansards proceedings and a separate test set of 447 manually word-aligned sentences provided by [1].

Our results — which comprise alignment quality and processing time — are summarized in Tab. 1. Processing time is measured from the beginning of processing till the end of the symmetrization process. It is not surprising that there are no significant differences between Giza++ and MGiza++ when AER is considered. SyMGiza++ achieves the best AER results with a relative improvement of more than 17% compared to Giza++ and MGiza++.

4.2 Machine Translation Results

We agree with [11] that the evaluation of alignment quality on its own may not be very meaningful. It should be considered good practice to include an evaluation of statistical machine translation models produced from the seemingly improved word alignment. In this section, such an evaluation is presented.

Table 2: BLEU scores for WMT08 data and test sets

(a) Europarl (test2008)

	fr-en	en-fr	es-en	en-es	de-en	en-de	es-de	de-es
MGiza++	0.3189	0.2944	0.3241	0.3184	0.2656	0.1982	0.1996	0.2706
SyMGiza++	0.3193	**0.3000**	0.3231	0.3172	0.2657	0.1993	0.2014	**0.2741**

(b) News Commentary (nc-test2008)

	fr-en	en-fr	es-en	en-es	de-en
MGiza++	0.2565	0.2339	0.3367	0.3220	0.2305
SyMGiza++	**0.2630**	0.2359	**0.3380**	0.3234	**0.2381**

	en-de	es-de	de-es	cz-en	en-cz
MGiza++	0.1531	0.1233	0.1775	0.2281	0.1285
SyMGiza++	**0.1575**	0.1234	**0.1822**	**0.2369**	**0.1329**

(c) Hunglish (newstest2008)

	hu-en	en-hu
MGiza++	0.0587	0.0449
SyMGiza++	**0.0632**	0.0458

For our base line systems, we configured Moses [12] as described by the ACL 2008 Third Workshop on Statistical Machine Translation (WMT-08) – Shared Translation Task guidelines. The data sets provided for the WMT-08 Shared Task[3] were used as training, tuning, and test data. Furthermore, training, tuning, and evaluation were performed in compliance with these guidelines. Several base line systems were created to account for different language pairs, training corpora and translation directions. All base line systems make use of MGiza++ to produce the word alignment models which serve as input to the bilingual phrase extraction phase of the Moses training process.

For our systems, we modified the training process only by replacing MGiza++ with SyMGiza++, no other parameters or steps were altered. Thus the *grow-diag* post-alignment symmetrization method is used and not the refined method introduced previously. We refer to the translation systems by the name of the alignment tool used. Thus the baselines are simply denoted by MGiza++, the systems created from the jointly trained alignment models by SyMGiza++.

The BLEU scores for all systems and language pairs have been compiled into Tab. 2. We split the results according to three different training corpora used. Bold figures mark results that are statistically significantly better than their counterpart. Significance has been calculated as proposed by [13].

The BLEU results for the SyMGiza++ translations exceed the results for the MGiza++ systems for all translation directions and language pairs with the exception of es-en and en-es. The results for these two translation directions, however, are statistically not significant. All statistically significant results hint towards small improvements in translation quality when the jointly trained

[3] Available at http://www.statmt.org/wmt08/

alignment models of SyMGiza++ are used. These effect seems to be more visible for small training corpora like the news commentary parallel corpus which comprises about 70.000 sentence pairs. The Europarl parallel corpus and the English-Hungarian corpus feature both more than one million sentence pairs. The very low BLEU scores for the English-Hungarian language pair result from the use of an out-of-domain test set provided with the WMT-08 data. The test set has been compiled from a news source while the training data consists of various legal texts and other data scraped from the internet.

5 Conclusions

We have presented SyMGiza++, a tool that computes symmetric word alignment models with the capability to take advantage of multi-processor systems. Our fairly simple modification to the well-known IBM Models implemented in Giza++ and MGiza++ achieves quite impressive improvements for AER on the standard Canadian Hansards task. Our symmetrized models outperform post-training symmetrization methods.

Improvements in translation quality — though less significant than in terms of pure AER — are also visible when SyMGiza++ is used as a drop-in replacement for Giza++ or MGiza++ in the training procedure of the phrase-based statistical machine translation system Moses. Translation quality improved for 18 out of 20 directions tested for three different training corpora and test sets provided for WMT-08. The differences for a majority of the improved results were statistically significant.

It can be safely concluded that SyMGiza++ can be employed anywhere instead of Giza++ or MGiza++ and in most cases will yield better or at least not worse results than any of the two widely used tools.

References

1. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics **29**(1) (2003) 19–51
2. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics **19**(2) (1993) 263–311
3. Vogel, S., Ney, H., Tillmann, C.: Hmm-based word alignment in statistical translation. In: Proceedings of ACL. (1996) 836–841
4. Zens, R., Matusov, E., Ney, H.: Improved word alignment using a symmetric lexicon model. In: Proceedings of ACL-COLING. (2004) 36
5. Liang, P., Taskar, B., Klein, D.: Alignment by agreement. In: Proceedings of ACL-COLING. (2006) 104–111
6. Gao, Q., Vogel, S.: Parallel implementations of word alignment tool. In: Proceedings of SETQA-NLP. (2008) 49–57
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistcial Society, series B **39**(1) (1977) 1–38

8. Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I., Och, F., Purdy, D., Smith, N., Yarowsky, D.: Statistical machine translation. Technical report, JHU workshop (1999)
9. Matusov, E., Zens, R., Ney, H.: Symmetric word alignments for statistical machine translation. In: Proceedings of ACL-COLING. (2004) 219–225
10. Mihalcea, R., Pedersen, T.: An evaluation exercise for word alignment. In: Proceedings of HLT-NAACL. (2003) 1–10
11. Fraser, A., Marcu, D.: Measuring word alignment quality for statistical machine translation. Computational Linguistics **33** (2007) 239–303
12. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: ACL, ACL (2007)
13. Koehn, P.: Statistical significance tests for machine translation evaluation. In: EMNLP. (2004) 388–395