Predictive Models of Performance in Multi-Engine Machine Translation

Tadashi Nomoto
National Institute of Japanese Literature
1-16-10 Yutaka Shinagawa
Tokyo 142-8585 Japan
nomoto@acm.org

Abstract

The paper describes a novel approach to Multi-Engine Machine Translation. We build statistical models of performance of translations and use them to guide us in combining and selecting from outputs from multiple MT engines. We empirically demonstrate that the MEMT system based on the models outperforms any of its component engine.

1 Introduction

As an increasing number of vendors are putting their MT systems on the market, an interesting question to ask is 'Is it possible to somehow combine them to get an MT system that beats every one of them?' This is a question we will address in the paper.

A quick survey of prior research on multi-engine machine translation (MEMT) shows that none ever addressed a question of using black box MT systems, which are what commercial systems usually are, to get an MEMT system. Frederking and Nirenburg (1994) developed a MEMT architecture which operates by combining outputs from three different engines based on the knowledge it has about inner workings of each of the component engines. Brown and Frederking (1995) is a continuation of Frederking and Nirenburg (1994) with an addition of a ngram-based mechanism for a candidate selection.

With this approach, however, one has to pay a high price when he or she wants to add a new engine or replace an existing one by something else: one needs to redesign a scoring mechanism so that it can compare scores from a newly added engine with those from existing ones.

In the paper, we take a novel approach to the problem of MEMT. Rather than devising an ad-hoc way of selecting from among candidate translations, we plan to build a generic statistical model of the reliability or performance of translations and let that model guide us in selecting the best candidate. Another feature of our approach is that we do not make any reference to an internal structure of MT engines. We exploit whatever information is available in the input/output of MT systems.

2 MEMT and Evaluation Measures

2.1 Using Black Box MTs

Typically the MEMT works by combining and selecting from among outputs of multiple MT engines. In the paper, we will be working with four commercially available, black box MT systems (call them 'Ai','Lo', 'At', 'Ib'). This is a marked difference from previous work on MEMT (Frederking and Nirenburg, 1994; Brown and Frederking, 1995; Hogan and Frederking, 1998), as they typically allow the MEMT to exploit information on inner workings of its component MT engines. While the availability of information on specifics of component engines may be crucial for improving estimates of the quality of the engines' outputs, its use severely limits the generality and scope of an approach that a given MEMT represents. This is why we pursue a different avenue here.

In an approach with black box MTs, all that is available to us is whatever information we may glean from a source text and its translation in a target language. Since none of the functionalities of the MEMT depends on properties specific to its component engines, in principle we should be able to work with any number of MTs of any type.

In addition, we will be concerned with a particular MEMT model where translation proceeds on a sentence by sentence basis. Thus the MEMT here collects and chooses among candidate *sentence* translations from each engine.

2.2 The problem with BLEU

Given that the job of MEMT is to choose among sentence translations generated by different MT systems, one way of evaluating its performance is by asking how frequently it finds the best translation available from the relevant MT systems. But this requires us to somehow articulate the notion of the 'best translation.'

Obviously, we could define the best translation to be one that gives the highest score in BLEU (Papineni et al., 2002).[1] However, one problem with using BLEU on a sentence-for-sentence basis is that translations often end up with zero because model translations they refer to do not contain n-grams of a particular length. This would make impossible a comparison and selection among possible translations.

Consider the following.

reference Thank you

translation Thank you very much

BLEU gives you 0 for the pair above with $N = 3$.

Our way out of this is to back off to a somewhat imprecise yet robust metric for evaluating translations. For a reference translation r and a machine-generated translation t, we define *m-precision* as:

$$\textit{m-precision} = \sum_{i}^{N} \frac{\sum_{v \in S_t^i} C(v, r)}{\sum_{v \in S_t^i} C(v, t)},$$

which is nothing more than Papineni et al. (2002)'s *modified n-gram precision* applied to a pair of a single reference and the associated translation. S_t^i here denotes a set of i-grams in t. $C(v, t)$ indicates the count of v in t. The m-precision of the previous example is $\frac{1}{2} + \frac{1}{3} + 0$ ($N = 3$).

Table 1: The average performance in BLEU of the systems over 22 blocks of EJP-99.

Ai	Lo	At	Ib	OpMEMT
0.2784	0.1674	0.1488	0.1559	0.3214

2.3 Is More Better?

Define the optimal MEMT or OpMEMT, as we might call it, as a system that operates by choosing, among outputs from MT systems, one that gives the best m-precision, for each translation. Note that the OpMEMT provides a practical upper bound for performance in BLEU for MEMTs, as an increase in m-precision generally translates into an increase in BLEU. (We may take a note that the first term of the exponential in Equation 1 is proportional to m-precision.)

Now the question we need to ask is, "Does the OpMEMT perform better than any one of the MTs which comprise it?" If it does not, then the whole enterprise of MEMT will not merit a serious consideration.

To answer the question, we ran some experiments using two data sets from separate domains. One is a parallel corpus derived from a phrase book for English letter writing in business (Takubo and Hashimoto, 1999). We call the corpus "EJP-99" for the sake of convenience throughout the paper. EJP-99 consists of the total of 10,965 pairs of English sentences and associated Japanese translations. We divided the corpus into 22 blocks of sentences of roughly equal size (each containing about 500 sentences), and ran each MT system on each of the 22 blocks, measuring its performance with BLEU. Another data set, which we will refer to as "CPC-02," is part of a huge bilingual corpus consisting of 150,000 semi-automatically aligned pairs of English and Japanese sentences from a newspaper domain (Utiyama and Isahara, 2002). CPC-02 contains 8,307 most closely aligned pairs of them.

Table 1 gives the average performance in BLEU over EJP-99 of the four commercial MT systems and the OpMEMT where BLEU takes into account

[1] BLEU is basically n-gram based precision and defined as:

$$\text{BLEU} = \exp\left(\log\left(\prod_{i}^{N} P_i\right)^{\frac{1}{N}} + \min(1 - \frac{r}{c}, 0)\right), \quad (1)$$

where N is the maximal length of n-grams, c the length of a candidate translation and r the length of a reference translation. P_i represents precision of n-grams of length i, *i.e.* the extent of overlaps between a reference and a translation as measured by n-grams of length i.

Table 2: Kendall's rank correlation between BLEU and m-precision for each MT ($N = 3$). The correlation is tested on BLEU and m-precision values of the 22 blocks of EJP-99. A figure in the parentheses indicates a p-value. Also, '**' indicates that the associated estimate is statistically significant at $p < 0.01$.

	Ai	Lo	At	Ib
τ	0.7446**	0.7446**	0.7662**	0.8442**

ngrams of up to 3 words in length, i.e., $N = 3$. (We assume that $N = 3$ throughout the rest of the paper. In particular, we may refer to BLEU as BLEU(3) as a way of making the policy explicit.)

We find in Table 1 that the OpMEMT outperforms, by a respectable margin, any of the four systems which together form the MEMT. The results make a strong case for the feasibility of MEMT.

Further Table 2 demonstrates that m-precision strongly correlates with BLEU, justifying the use of m-precision in place of BLEU for evaluating translation performance.

3 Predictive Models of Performance

Given that the idea of MEMT is in principle feasible, then the question we need to turn to is, how do we go about actually building one? Our idea is, instead of asking an oracle which translation to pick as in OpMEMT, we call on some statistical model to decide which one is worth a pick. The model's job is to decide a best pick among MT outputs based on a prediction it makes about how accurate the outputs are. Below, we will consider two kinds of such model: an ngram-based model and an alignment based model.

3.1 The Fluency Based Model (FLM)

For an English sentence e, and its Japanese translation $j_{(e)}$ by some MT system, we define the performance (or quality) of $j_{(e)}$ by:

$$FLM(e, j_{(e)}) = \log P(j_{(e)}),$$

where for a sequence of words $w^m = w_1, \ldots, w_m$,

$$P(w^m) = \prod_i^m P(w_i \mid w_{i-N+1} \cdots w_{i-1})$$

In the FLM, we assume that the translation quality of $j_{(e)}$ depends only on its ngram based fluency.

Throughout the paper, we will be working with the 4-gram based language model (i.e., $N = 4$), as other choices of N led to somewhat poorer results in a preliminary testing with the model.

3.2 The Alignment Based Model (ALM)

What we call the "Alignment Based Model" (or ALM for short) here heavily draws upon statistical translation models developed by Brown et al. (1993) (which are popularly dubbed 'IBM models'). Recall that given a pair of sentences, one in a source language and the other in a target language, the models are able to tell you the probability that a sentence in the target language is a translation of a sentence in the source language. Notice that one could turn them into performance models for MTs because it is possible and legitimate to ask them, 'What is the probability that a particular output from an MT system, could have generated a sentence which one likes to translate into the target language?' Put simply, given a pair of sentences, e and a machine generated translation of e (call it j), one could use the IBM models to find out how likely it is that j could have generated e and use that information to determine the quality of the translation.

Formally, we could represent the alignment based model of performance in the following manner. We write $j_{(e)}$ to mean that j is a translation of e by some MT system.

$$\begin{aligned} ALM(e, j_{(e)}) &= \log P(j_{(e)} \mid e) \\ &\approx \log P(j_{(e)}) P(e \mid j_{(e)}) \end{aligned} \quad (2)$$

Assume in addition that:

$$P(e \mid j_{(e)}) = \sum_{\mathbf{a}} P(e, \mathbf{a} \mid j_{(e)}) \quad (3)$$

'**a**' denotes some alignment between e and j, i.e., a table of associations between words in j and those in e. For instance, we may have an association $j_1\, j_2\, j_3 \rightarrow e_5$, which means that some sequence of words j_1 through j_3 is associated with or translated into a single word e_5. The above equation says that the probability of observing e given $j_{(e)}$ is the sum of the probabilities of observing e given $j_{(e)}$ under every possible alignment between e and j.

Also of some note is that we could in fact regard the ALM as embodying two features generally agreed to be most relevant for evaluating MTs: fidelity and fluency (White, 2001).[2] This could be easily seen if we rewrite Equation 2 as:

$$FLM(e, j_{(e)}) + \log P(e \mid j_{(e)}). \qquad (4)$$

Notice that the first term in Equation 4 is the N-gram based performance model discussed earlier, the second term represents an alignment model, which measures in probabilistic terms how well a source text associates itself with its translation.

It is worth noting that there is another effort in the literature to provide a quantitative characterization of fidelity. Rajman and Hartley (2001) introduces what they call the *D-score* to measure how much of a semantic content of the source text is preserved in its translation.[3]

IBM Models The ALM uses IBM Model 1 to estimate the alignment probability $P(e \mid j_{(e)})$. The IBM Model 1 is one of the translation models developed in Brown et al. (1993), which also include Model 2 and 3.

To give a rough picture of Model 1 and 2, Model 1 operates on a simplistic assumption that each word in e is equally likely to associated with any one of words in j, whereas Model 2 assumes that some words in e are more (or less) likely to be associated with particular words in j than other words that appear in e.

Model 3 is an refinement over the IBM Model 1 and 2 and differs from the latter two in the way it decomposes the probability $P(e \mid j_{(e)})$. Model 3 is motivated by the observation that some words in English could give rise to translations in French where each English word corresponds to a group of different French words. Arguably, the observation could hold for any pair of languages, not just English and French.

The difference among Model 1 to 3 boils down to the way each model defines the probability $P(e, \mathbf{a} \mid j_{(e)})$. Details aside, it can be thought of as the product of the probability of e_i being a *lexical* translation of $j_{(e)k}$ and the probability that the position i in e is aligned with the position k in $j_{(e)}$ under some alignment **a**, where e_i stands for the i-th word in e and $j_{(e)k}$ for k-th word in $j_{(e)}$. Models 1 to 3 furnish from impoverished to improved variations of the alignment probability.

3.3 Learning with Support Vector Regression

One of the problems with the ALM and FLM is that they do not take into account the reliability of an MT engine. It is reasonable to believe that some MT systems could be inherently more prone to errors and outputs they produce tend to be of less quality then those from other systems, regardless of the outputs' fluency or translation probability. As we might recall, the ALM and FLM make predictions solely based on how well-formed MT outputs are and how well they align with the source texts; they do not take it into consideration that outputs from a particular system could be inherently less reliable than those from others.

One way of dealing with the issue, which we pursue below, is to train a regression model on m-precision scores obtained by each MT system and use that model to correct estimates produced by the ALM and FLM. (Recall that the use of BLEU in a sentence-by-sentence evaluation often results in a translation scoring zero, making impossible a comparison of translation performance across MT systems, which motivated us to look to m-precision instead.) With regression, we would be able to correct the translation probability/fluency of a given output according to the reliability of an MT system that generates it.

Now a regression model of a particular interest here is Support Vector regression, which works pretty much like its enormously popular counterpart, i.e., Support Vector classification, except that we are going to work with real numbers for target values and construct the margin, using Vapnik's ϵ-insensitive loss function, which ignores errors less than ϵ (known as *precision* and depicted as E in

[2] Fidelity here refers to the extent to which meaning in the source text is conveyed to its translation. Fluency concerns the well-formedness of MT outputs. Both form part of the DARPA MT evaluation metrics (White, 2001).

[3] They work on the premise that a similarity matrix among source texts should be similar to that for their translation counterparts. A similarity matrix here is thought of as a matrix of cosine based similarity values found among texts. To find out a d-score of a source text s and its translation $\tau(s)$, one builds a vector of similarity scores for s and every one of texts in the source language and also that for $\tau(s)$ for each corresponding translation, then calculates a cosine value for the vector for the source text and that for the translation.

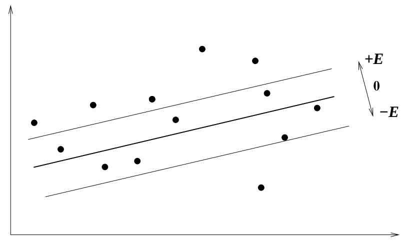

Figure 1: Support Vector Regression

Fig.1) (Schölkopf et al., 1998). One drawback here is that ϵ needs to be specified manually.

In Fig.1, the line running between the two boundary lines represents a target function y. In Support Vector regression, one's job is to find an optimal fit to the data with a tube with the radius of ϵ.

Consider a linear regression

$$h(\vec{x}) = \vec{w} \cdot \vec{x} + b,$$

with input data $\vec{x} = (x_i, \ldots, x_m)$ and the corresponding weights $\vec{w} = (w_i, \ldots, w_m)$. '$x \cdot y$' denotes the inner product of x and y. Support Vector regression determines parameters $\vec{w}$ and b, in such a way as to minimize, together with the model complexity, the cost of the distance between the tube and points falling outside of it.

It is straightforward to extend the ALM and FLM with Support Vector regression, which consists of plugging in either model as an input variable in the regressor. This would give us the following two regression models.

Regressive FLM (rFLM)

$$h(FLM(e,j)) = w \cdot FLM(e,j) + b$$

Regressive ALM (rALM)

$$h(ALM(e,j)) = w \cdot ALM(e,j) + b$$

A variant of rALM is also possible, where the fluency and alignment estimates are assigned to separate parameters, and takes the following form.

Regressive ALM$^+$ (rALM$^+$)

$$h(\vec{x}) = \vec{w} \cdot \vec{x} + b,$$

where $\vec{x} = (\log P(j), \log P(e \mid j))$.

Notice that unlike the ALM and FLM, the regression models, once trained, will predict the m-precision of MT outputs.

4 Evaluation

4.1 The Data sets

We ran several sets of experiments to find out how the MEMT models fare on EJP-99 as well as on CPC-02. We set up the experiments roughly the same way we have for evaluating correlations between m-precision and BLEU (section 2.3). We divided EJP-99 into 22, roughly equal-sized (500-sentence long) blocks of sentences, and ran the usual cross validation on them to get estimates of performance of the models. As for CPC-02, we divided them into 17 blocks of sentences, so that each of them comes out about 500-sentence long. We had a total of 10,965 pairs of English/Japanese sentences for EJP-99, and 8,307 pairs of aligned sentences for CPC-02.

Recall that we are dealing here with English to Japanese translation and are going to evaluate performance in terms of how well the translations the MEMT models produce look like associated reference translations, which is what BLEU tells us.

Also one thing to take note of is that although the MEMT models choose among translation outputs from the engines based on m-precision,[4] we can safely measure the entirety of selections by BLEU, which is what we do below.

4.2 Results and Discussion

Before delving into the details, let us briefly acknowledge credits to software tools we took advantage of. FLMs are built using a language modeling tool kit known as 'CMU-Cambridge-SLM ToolKit.'[5] We trained the FLM on a collection of several Japanese news paper corpora, which is separate from EJP-99 and from CPC-02, and contains the total of 167,010 distinct words. Second, alignment based models, i.e., ALMs, are built using GIZA, a tool kit for statistical translation modeling.[6] The translation alignment part of ALMs, namely, $P(e \mid j_{(e)})$, is trained through cross validation. Finally, the Support Vector regression makes use of a package known as mySVM which is freely available.[7]

[4] Recall that the MEMT operates by selecting an MT output ranking highest in whatever metric is used.

[5] svr-www.eng.cam.ac.uk/ prc14/toolkit.html

[6] www.clsp.jhu.edu/ws99/projects/mt/toolkit/index-old.html

[7] www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM

Table 3: Average performance of the MT systems on EJP-99.

Model	BLEU(3)	System	BLEU(3)
OpMEMT	0.3214	Ai	**0.2784**
rFLM	**0.2799**	Lo	0.1674
FLM	0.2535	At	0.1488
rALM$^+$	0.2789	Ib	0.1559
rALM	0.2794		
ALM$^-$	0.2523		
ALM	0.2696		

Table 4: Accuracy of predictions on EJP-99.

Model	accuracy	System	accuracy
OpMEMT	1.0	Ai	**0.4260**
rFLM	**0.4354**	Lo	0.2123
FLM	0.4091	At	0.1691
rALM$^+$	0.4324	Ib	0.1905
rALM	0.4348		
ALM$^-$	0.3710		
ALM	0.4215		

Now let us look at results we got from the experiments, which are listed in Table 3. We see that rFLM takes a lead followed by rALM$^+$ with FLM further behind. ALM and ALM$^-$ are falling out of the race, altogether. (ALM$^-$ is an 'alignment only' model, makes use of the probability $P(e \mid j_{(j)})$ alone.) Notice, however, subtle differences between rFLM and Ai, the former slightly ahead of the latter.[8] Also we note effects of regression on performance. As Table 3 shows, harnessing MEMTs with regression usually leads to an improved performance: rFLM is superior to FLM by over 2 percent points, and rALM performs better than ALM by about 1 percent point.

Hogan and Frederking (1998) introduces a new kind of yardstick for measuring the effectiveness of MEMT systems. The rationale for this is that it is often the case that the effectiveness of MEMT systems does not translate into performance of outputs that they generate. We recall that with BLEU, one measures performance of translations, not how often a given MEMT system picks the best translation among candidates. The problem is, even if a MEMT is right about its choices more often than its best component engine, BLEU may not tell that the MEMT is performing better than the latter. This happens because BLEU works with a *block* of translations,[9] and therefore is insensitive to small differences in outputs of MT systems: if the MEMT and its competing engine do not differ much in their choices for translations on a particular block, as is apparently the case here, what small contributions the MEMT makes is overshadowed by a large number of identical choices the two systems make on that block and may not show itself in BLEU. This is why we turn to Hogan and Frederking (1998).

Now what they suggest is the following.

$$d(\tau_m) = \frac{\sum_i^N \delta(\sigma_{im}, \max\{\sigma_{i1} \cdots \sigma_{oM}\})}{N} \qquad (5)$$

where $\delta(i, j)$ is the Kronecker delta function, which gives 1 if $i = j$ and 0 otherwise. Here τ_m represents some MEMT system, σ_{im} indicates whatever a score is assigned to a translation generated by τ_m for a sentence i, $\sigma_{i1}, \ldots, \sigma_{iM}$ denotes a list of scores assigned to translations that systems τ_i through τ_M generated. N is the number of sentences.

Equation 5 provides a straightforward way of capturing precision of outputs of a MEMT system, and moreover, gives a more accurate picture of performance of a MEMT system than BLEU. Now let us see how things look with this new yardstick, which we call "accuracy." We take σ_{ij} to be an m-precision predicted by a system j for some translation i.

Table 4 gives performance in accuracy for each of the systems. Notice that differences between rFLM and Ai clearly stand out.

Let us move on to the CPC-02 task, where we had the same setup as for EJP-99, except that the data set is comprised of 8,307 sentences and is from a newspaper domain. We divided the data into 17 about equal-sized blocks of sentences (each about 500-sentence long), and ran a 17-fold cross validation to evaluate performance of the systems. Table 5 lists the results. (The scores there are averaged over 17 folds.) We note here that Ai, a winner for EJP-99, now lost in performance to At, which is performing best on CPC-02.

[8] Though the differences are admittedly marginal, a statistical test (two-tailed t.test) do find that they are statistically significant with $p < 0.01$.

[9] Recall that each measurement of BLEU in our experiments is based on a block of 500 sentences.

Now how are the MEMT models doing on CPC-02? As we find from Table 5, top performing MEMT models are comparable in performance to At: there is no statistically significant difference between FLM and At, for instance. Accuracy results in Table 6 show also that top performing MEMT systems are in a tie with At.

So one may want to conclude from the results from EJP-99 and CPC-02 that the MEMTs are doing at least as good as the top performing MT system, but not better, which obviously means a failure for the MEMTs.

However, an entirely different picture of the MEMTs emerges when one combines the results from EJP-99 and CPC-02 and look at average performance over the two data sets, which is given in Table 7 and Table 8. Table 7 shows performance in BLEU of the systems over the combined data set. What we find there is a clear indication that rFLM, a MEMT model, outperforms Ai, which tops the four MT engines run on the combined data set. The difference in performance between rFLM and Ai is statistically significant ($p < 0.01$).

Accuracy of rFLM is also found to be significantly higher than that of Ai, as shown in Table 8. It is notable that regression based MEMT models are doing fairly well compared to those without regression.

A respectable increase in BLEU as well as in accuracy of the MEMT systems, rFLM in particular, on the combined data results largely because the MEMTs are good at producing performance comparable rather than superior to a best MT engine available, which could vary from domain to domain. Note that while Ai performs best on EJP-99, At performs best on CPC-02. What is happening in Table 7 and 8 is that by feeding upon a best engine available for a particular domain it is working on, rFLM produces on average a better performance across the two domains than any of its component engines.

5 Conclusion

We opened the paper with the question ‘Is it possible to get a MEMT system that beats every one of MT systems that comprise it?’ Our answer to the question should be yes: while rFLM performs marginally superior to the four MT engines in EJP-99 and com-

Table 5: Average performance of the MT systems on CPC-02. The scores are averaged over 17 blocks.

Model	BLEU(3)	System	BLEU(3)
OpMEMT	0.2083	Ai	0.1471
rFLM	0.1706	Lo	0.1414
FLM	**0.1708**	At	**0.1709**
rALM$^+$	0.1706	Ib	0.1365
rALM	0.1708		
ALM$^-$	0.1465		
ALM	0.1567		

Table 6: Accuracy of predictions for CPC-02.

Model	accuracy	System	accuracy
OpMEMT	1.0	Ai	0.2386
rFLM	0.4190	Lo	0.1702
FLM	**0.4222**	At	**0.4192**
rALM$^+$	0.4193	Ib	0.1699
rALM	0.4201		
ALM$^-$	0.2403		
ALM	0.3178		

Table 7: Average performance of the MT systems over EJP-99 and CPC-02.

Model	BLEU(3)	System	BLEU(3)
OpMEMT	0.2649	Ai	**0.2128**
rFLM	**0.2253**	Lo	0.1544
FLM	0.2122	At	0.1599
rALM$^+$	0.2248	Ib	0.1462
rALM	0.2251		
ALM$^-$	0.1994		
ALM	0.2132		

Table 8: Accuracy of predictions averaged over EJP-99 and CPC-02.

Model	accuracy	System	accuracy
OpMEMT	1.0	Ai	**0.3323**
rFLM	0.4272	Lo	0.1913
FLM	0.4157	At	0.2942
rALM+	0.4259	Ib	0.1802
rALM	**0.4275**		
ALM-	0.3057		
ALM	0.3697		

Table 9: Effects on BLEU(3) of the alignment model $P(e \mid j_{(e)})$. The results are from EJP-99.

Model	IBM Model 1	IBM Model 3
rALM^{+}	0.2789	0.1679
rALM	0.2794	0.1662
ALM	0.2696	0.1614
ALM^{-}	0.2523	0.1591

parably to a top performer in CPC-02, it beats each of them over a combined data set by a significant margin. This happens because rFLM is able to pick whatever best engine is available for a given domain and feeds upon it. The results in Table 7 and Table 8 could be look at as showing that the best system for one domain may not be best for another. A careful look at properties of a domain might indeed reveal some useful predictors of performance of an MT system.

Curiously enough, in neither of the data sets, EJP-99 and CPC-02, are the ALM and its variants performing significantly better than less sophisticated models like FLM and rFLM. This would mean that information on alignments, i.e., $P(e \mid j_{(e)})$, has little to contribute to predicting translation performance.

To find out more about the issue, we looked at what happens if we use a more refined alignment model like Model 3 in place of Model 1, which the (r)ALMs are all based on.

Table 9 shows the results: we find a huge decline in performance of ALM and ALM^{-}, and also for rALM and rALM^{+}. We could think of several reasons for this. One is that as Model 3 makes use of more parameters than Model 1 (i.e. fertility), it may require more training data than EJP-99 provides to produce reliable estimates for them.

Another may have to do with the languages involved: since Japanese is a language with no fixed word order, this could have made it harder to get good estimates for alignments than when working with a fixed word-order language like German. Note that using the alignment probability $a(i \mid j, m, l)$, as Model 3 does, is a good idea when both of the languages involved exhibit a stable word order, but perhaps not when one of them has a free word order.

In any case, the results from EJP-99 and CPC-02, and also from Table 9 suggest that the effectiveness of $P(e \mid j_{(e)})$ for predicting performance is suspect at best.

References

Ralf Brown and Robert Frederking. 1995. Applying statistical English language modelling to symbolic machine translation. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'95)*, pages 221–239, Leuven, Belgium, July.

Peter F. Brown, Stephen A. Della Pietra, Vincent J.Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Rober Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, Stuttgart.

Christopher Hogan and Robert E. Frederking. 1998. An evaluation of the multi-engine MT architecture. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas (AMTA '98)*, pages 113–123, Berlin, October. Springer-Verlag. Lecture Notes in Artificial Intelligence 1529.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei ing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, July.

Martin Rajman and Anthony Hartley. 2001. Automatically predicting MT systems rankings compatible with fluency, adequacy or informativeness score. In *Proceedings of the workshop "MT Evaluation: Who did What to Whom"*, pages 29–34.

Bernhard Schölkopf, Chirstpher J. C. Burges, and Alexander J. Smola, editors. 1998. *Advances in Kernerl Methods: Support Vector Learning*. The MIT Press.

Kohei Takubo and Mitsunori Hashimoto. 1999. A Dictionary of English Business Letter Expressions. Published in CDROM. Nihon Keizai Shinbun Sha.

Masao Utiyama and Hitoshi Isahara. 2002. Alignment of japanese-english news articles and sentences. In *IPSJ Proceedings 2002-NL-151*, pages 15–22. In Japanese.

John White. 2001. Predicting intelligibility from fidelity in MT evaluation. In *Proceedings of the workshop "MT Evaluation: Who did What to Whom"*, pages 35–37.