Improving IBM Word-Alignment Model 1*

Robert C. MOORE
Microsoft Research
One Microsoft Way
Redmond, WA 90052
USA
bobmoore@microsoft.com

Abstract

We investigate a number of simple methods for improving the word-alignment accuracy of IBM Model 1. We demonstrate reduction in alignment error rate of approximately 30% resulting from (1) giving extra weight to the probability of alignment to the null word, (2) smoothing probability estimates for rare words, and (3) using a simple heuristic estimation method to initialize, or replace, EM training of model parameters.

1 Introduction

IBM Model 1 (Brown et al., 1993a) is a word-alignment model that is widely used in working with parallel bilingual corpora. It was originally developed to provide reasonable initial parameter estimates for more complex word-alignment models, but it has subsequently found a host of additional uses. Among the applications of Model 1 are segmenting long sentences into subsentental units for improved word alignment (Nevado et al., 2003), extracting parallel sentences from comparable corpora (Munteanu et al., 2004), bilingual sentence alignment (Moore, 2002), aligning syntactic-tree fragments (Ding et al., 2003), and estimating phrase translation probabilities (Venugopal et al., 2003). Furthermore, at the 2003 Johns Hopkins summer workshop on statistical machine translation, a large number of features were tested to discover which ones could improve a state-of-the-art translation system, and the only feature that produced a "truly significant improvement" was the Model 1 score (Och et al., 2004).

Despite the fact that IBM Model 1 is so widely used, essentially no attention seems to have been paid to whether it is possible to improve on the standard Expectation-Maximization (EM) procedure for estimating its parameters. This may be due in part to the fact that Brown et al. (1993a) proved that the log-likelihood objective function for Model 1 is a strictly concave function of the model parameters, so that it has a unique local maximum. This, in turn, means that EM training will converge to that maximum from any starting point in which none of the initial parameter values is zero. If one equates optimum parameter estimation with finding the global maximum for the likelihood of the training data, then this result would seem to show no improvement is possible.

However, in virtually every application of statistical techniques in natural-language processing, maximizing the likelihood of the training data causes overfitting, resulting in lower task performance than some other estimates for the model parameters. This is implicitly recognized in the widespread adoption of early stopping in estimating the parameters of Model 1. Brown et al. (1993a) stopped after only one iteration of EM in using Model 1 to initialize their Model 2, and Och and Ney (2003) stop after five iterations in using Model 1 to initialize the HMM word-alignment model. Both of these are far short of convergence to the maximum likelihood estimates for the model parameters.

We have identified at least two ways in which the standard EM training method for Model 1 leads to suboptimal performance in terms of word-alignment accuracy. In this paper we show that by addressing these issues, substantial improvements in word-alignment accuracy can be achieved.

2 Definition of Model 1

Model 1 is a probabilistic generative model within a framework that assumes a source sentence S of length l translates as a target sentence T, according to the following stochastic process:

- A length m for sentence T is generated.
- For each target sentence position $j \in \{1, \ldots, m\}$:
  - A generating word s_i in S (including a null word s_0) is selected, and

* From *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, pp. 519–526.

– The target word t_j at position j is generated depending on s_i.

Model 1 is defined as a particularly simple instance of this framework, by assuming all possible lengths for T (less than some arbitrary upper bound) have a uniform probability ϵ, all possible choices of source sentence generating words are equally likely, and the translation probability $tr(t_j|s_i)$ of the generated target language word depends only on the generating source language word—which Brown et al. (1993a) show yields the following equation:

$$p(T|S) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} tr(t_j|s_i) \qquad (1)$$

Equation 1 gives the Model 1 estimate for the probability of a target sentence, given a source sentence. We may also be interested in the question of what is the most likely alignment of a source sentence and a target sentence, given an instance of Model 1; where, by an alignment, we mean a specification of which source words generated which target words according to the generative model. Since Model 1, like many other word-alignment models, requires each target word to be generated by exactly one source word (including the null word), an alignment a can be represented by a vector $a_1, \ldots, a_m$, where each a_j is the sentence position of the source word generating t_j according to the alignment. It is easy to show that for Model 1, the most likely alignment $\hat{a}$ of S and T is given by this equation:

$$\hat{a} = \text{argmax}_a \prod_{j=1}^{m} tr(t_j|s_{a_j}) \qquad (2)$$

Since in applying Model 1, there are no dependencies between any of the a_js, we can find the most likely aligment simply by choosing, for each j, the value for a_j that leads to the highest value for $tr(t_j|s_{a_j})$.

The parameters of Model 1 for a given pair of languages are normally estimated using EM, taking as training data a corpus of paired sentences of the two languages, such that each pair consists of sentence in one language and a possible translation in the other language. The training is normally initialized by setting all translation probability distributions to the uniform distribution over the target language vocabulary.

3 Problems with Model 1

Model 1 clearly has many shortcomings as a model of translation. Some of these are structural limitations, and cannot be remedied without making the model significantly more complicated. Some of the major structural limitations include:

- (Many-to-one) Each word in the target sentence can be generated by at most one word in the source sentence. Situations in which a phrase in the source sentence translates as a single word in the target sentence are not well-modeled.
- (Distortion) The position of any word in the target sentence is independent of the position of the corresponding word in the source sentence, or the positions of any other source language words or their translations. The tendency for a contiguous phrase in one language to be translated as a contiguous phrase in another language is not modeled at all.
- (Fertility) Whether a particular source word is selected to generate the target word for a given position is independent of which or how many other target words the same source word is selected to generate.

These limitations of Model 1 are all well known, they have been addressed in other word-alignment models, and we will not discuss them further here. Our concern in this paper is with two other problems with Model 1 that are not deeply structural, and can be addressed merely by changing how the parameters of Model 1 are estimated.

The first of these nonstructural problems with Model 1, as standardly trained, is that rare words in the source language tend to act as "garbage collectors" (Brown et al., 1993b; Och and Ney, 2004), aligning to too many words in the target language. This problem is not unique to Model 1, but anecdotal examination of Model 1 alignments suggests that it may be worse for Model 1, perhaps because Model 1 lacks the fertility and distortion parameters that may tend to mitigate the problem in more complex models.

The cause of the problem can be easily understood if we consider a situation in which the source sentence contains a rare word that only occurs once in our training data, plus a frequent word that has an infrequent translation in the target sentence. Suppose the frequent source word has the translation present in the target sentence only 10% of the time in our training data, and thus has an estimated translation probability of around 0.1 for this target word. Since the rare source word has no other occurrences in the data, EM training is free to assign whatever probability distribution is required to maximize the joint probability of this sentence pair. Even if the

rare word also needs to be used to generate its actual translation in the sentence pair, a relatively high joint probability will be obtained by giving the rare word a probability of 0.5 of generating its true translation and 0.5 of spuriously generating the translation of the frequent source word. The probability of this incorrect alignment will be higher than that obtained by assigning a probability of 1.0 to the rare word generating its true translation, and generating the true translation of the frequent source word with a probability of 0.1. The usual fix for over-fitting problems of this type in statistical NLP is to smooth the probability estimates involved in some way.

The second nonstructural problem with Model 1 is that it seems to align too few target words to the null source word. Anecdotal examination of Model 1 alignments of English source sentences with French target sentences reveals that null word alignments rarely occur in the highest probability alignment, despite the fact that French sentences often contain function words that do not correspond directly to anything in their English translation. For example, English phrases of the form $\langle \text{noun}_1 \rangle \langle \text{noun}_2 \rangle$ are often expressed in French by a phrase of the form $\langle \text{noun}_2 \rangle$ *de* $\langle \text{noun}_1 \rangle$, which may also be expressed in English (but less often) by a phrase of the form $\langle \text{noun}_2 \rangle$ *of* $\langle \text{noun}_1 \rangle$.

The structure of Model 1 again suggests why we should not be surprised by this problem. As normally defined, Model 1 hypothesizes only one null word per sentence. A target sentence may contain many words that ideally should be aligned to null, plus some other instances of the same word that should be aligned to an actual source language word. For example, we may have an English/French sentence pair that contains two instances of *of* in the English sentence, and five instances of *de* in the French sentence. Even if the null word and *of* have the same initial probabilty of generating *de*, in iterating EM, this sentence is going to push the model towards estimating a higher probabilty that *of* generates *de* and a lower estimate that the null word generates *de*. This happens because there are are two instances of *of* in the source sentence and only one hypothetical null word, and Model 1 gives equal weight to each occurrence of each source word. In effect, *of* gets two votes, but the null word gets only one. We seem to need more instances of the null word for Model 1 to assign reasonable probabilities to target words aligning to the null word.

4 Smoothing Translation Counts

We address the nonstructural problems of Model 1 discussed above by three methods. First, to address the problem of rare words aligning to too many words, at each interation of EM we smooth all the translation probability estimates by adding virtual counts according to a uniform probability distribution over all target words. This prevents the model from becoming too confident about the translation probabilities for rare source words on the basis of very little evidence. To estimate the smoothed probabilties we use the following formula:

$$tr(t|s) = \frac{C(t,s) + n}{C(s) + n \cdot |V|} \tag{3}$$

where $C(t,s)$ is the expected count of s generating t, C(s) is the corresponding marginal count for s, $|V|$ is the hypothesized size of the target vocabulary V, and n is the added count for each target word in V. $|V|$ and n are both free parameters in this equation. We could take $|V|$ simply to be the total number of distinct words observed in the target language training, but we know that the target language will have many words that we have never observed. We arbitrarily chose $|V|$ to be 100,000, which is somewhat more than the total number of distinct words in our target language training data. The value of n is empirically optimized on annotated development test data.

This sort of "add-n" smoothing has a poor reputation in statistical NLP, because it has repeatedly been shown to perform badly compared to other methods of smoothing higher-order n-gram models for statistical language modeling (e.g., Chen and Goodman, 1996). In those studies, however, add-n smoothing was used to smooth bigram or trigram models. Add-n smoothing is a way of smoothing with a uniform distribution, so it is not surprising that it performs poorly in language modeling when it is compared to smoothing with higher order models; e.g, smoothing trigrams with bigrams or smoothing bigrams with unigrams. In situations where smoothing with a uniform distribution is appropriate, it is not clear that add-n is a bad way to do it. Furthermore, we would argue that the word translation probabilities of Model 1 are a case where there is no clearly better alternative to a uniform distribution as the smoothing distribution. It should certainly be better than smoothing with a unigram distribution, since we especially want to benefit from smoothing the translation probabilities for the rarest words, and smoothing with a unigram distribution would assume that rare words are more likely to translate to frequent words than to other rare words, which seems counterintuitive.

5 Adding Null Words to the Source Sentence

We address the lack of sufficient alignments of target words to the null source word by adding extra null words to each source sentence. Mathematically, there is no reason we have to add an integral number of null words, so in fact we let the number of null words in a sentence be any positive number. One can make arguments in favor of adding the same number of null words to every sentence, or in favor of letting the number of null words be proportional to the length of the sentence. We have chosen to add a fixed number of null words to each source sentence regardless of length, and will leave for another time the question of whether this works better or worse than adding a number of null words proportional to the sentence length.

Conceptually, adding extra null words to source sentences is a slight modification to the structure of Model 1, but in fact, we can implement it without any additional model parameters by the simple expedient of multiplying all the translation probabilities for the null word by the number of null words per sentence. This multiplication is performed during every iteration of EM, as the translation probabilities for the null word are re-estimated from the corresponding expected counts. This makes these probabilities look like they are not normalized, but Model 1 can be applied in such a way that the translation probabilities for the null word are only ever used when multiplied by the number of null words in the sentence, so we are simply using the null word translation parameters to keep track of this product pre-computed. In training a version of Model 1 with only one null word per sentence, the parameters have their normal interpretation, since we are multiplying the standard probability estimates by 1.

6 Initializing Model 1 with Heuristic Parameter Estimates

Normally, the translation probabilities of Model 1 are initialized to a uniform distribution over the target language vocabulary to start iterating EM. The unspoken justification for this is that EM training of Model 1 will always converge to the same set of parameter values from any set of initial values, so the intial values should not matter. But this is only the case if we want to obtain the parameter values at convergence, and we have strong reasons to believe that these values do not produce the most accurate sentence alignments. Even though EM will head towards those values from any initial position in the parameter space, there may be some starting points we can systematically find that will take us closer to the optimal parameter values for alignment accuracy along the way.

To test whether a better set of initial parameter estimates can improve Model 1 alignment accuracy, we use a heuristic model based on the log-likelihood-ratio (LLR) statistic recommended by Dunning (1993). We chose this statistic because it has previously been found to be effective for automatically constructing translation lexicons (e.g., Melamed, 2000; Moore, 2001). In our application, the statistic can be defined by the following formula:

$$\sum_{t? \in \{t, \neg t\}} \sum_{s? \in \{s, \neg s\}} C(t?, s?) \log \frac{p(t?|s?)}{p(t?)} \qquad (4)$$

In this formula t and s mean that the corresponding words occur in the respective target and source sentences of an aligned sentence pair, $\neg t$ and $\neg s$ mean that the corresponding words do not occur in the respective sentences, $t?$ and $s?$ are variables ranging over these values, and $C(t?, s?)$ is the observed joint count for the values of $t?$ and $s?$. All the probabilities in the formula refer to maximum likelihood estimates.[1]

These LLR scores can range in value from 0 to $N \cdot \log(2)$, where N is the number of sentence pairs in the training data. The LLR score for a pair of words is high if the words have either a strong positive association or a strong negative association. Since we expect translation pairs to be positively associated, we discard any negatively associated word pairs by requiring that $p(t, s) > p(t) \cdot p(s)$.

To use LLR scores to obtain initial estimates for the translation probabilities of Model 1, we have to somehow transform them into numbers that range from 0 to 1, and sum to no more than 1 for all the target words associated with each source word. We know that words with high LLR scores tend to be translations, so we want high LLR scores to correspond to high probabilities, and low LLR scores to correspond to low probabilities. The simplest approach would be to divide each LLR score by the sum of the scores for the source word of the pair, which would produce a normalized conditional probability distribution for each source word.

Doing this, however, would discard one of the major advantages of using LLR scores as a measure of word association. All the LLR scores for rare words tend to be small; thus we do not put too much confidence in any of the hypothesized word associations for such words. This is exactly the property

[1] This is not the form in which the LLR statistic is usually presented, but it can easily be shown by basic algebra to be equivalent to $-\lambda$ in Dunning's paper. See Moore (2004) for details.

needed to prevent rare source words from becoming garbage collectors. To maintain this property, for each source word we compute the sum of the LLR scores over all target words, but we then divide every LLR score by the single largest of these sums. Thus the source word with the highest LLR score sum receives a conditional probability distribution over target words summing to 1, but the corresponding distribution for every other source word sums to less than 1, reserving some probability mass for target words not seen with that word, with more probability mass being reserved the rarer the word.

There is no guarantee, of course, that this is the optimal way of discounting the probabilities assigned to less frequent words. To allow a wider range of possibilities, we add one more parameter to the model by raising each LLR score to an empirically optimized exponent before summing the resulting scores and scaling them from 0 to 1 as described above. Choosing an exponent less than 1.0 decreases the degree to which low scores are discounted, and choosing an exponent greater than 1.0 increases degree of discounting.

We still have to define an initialization of the translation probabilities for the null word. We cannot make use of LLR scores because the null word occurs in every source sentence, and any word occuring in every source sentence will have an LLR score of 0 with every target word, since $p(t|s) = p(t)$ in that case. We could leave the distribution for the null word as the uniform distribution, but we know that a high proportion of the words that should align to the null word are frequently occuring function words. Hence we initialize the distribution for the null word to be the unigram distribution of target words, so that frequent function words will receive a higher probability of aligning to the null word than rare words, which tend to be content words that do have a translation. Finally, we also effectively add extra null words to every sentence in this heuristic model, by multiplying the null word probabilities by a constant, as described in Section 5.

7 Training and Evaluation

We trained and evaluated our various modifications to Model 1 on data from the bilingual word alignment workshop held at HLT-NAACL 2003 (Mihalcea and Pedersen, 2003). We used a subset of the Canadian Hansards bilingual corpus supplied for the workshop, comprising 500,000 English-French sentences pairs, including 37 sentence pairs designated as "trial" data, and 447 sentence pairs designated as test data. The trial and test data had been manually aligned at the word level, noting particular pairs of words either as "sure" or "possible" alignments, as described by Och and Ney (2003).

To limit the number of translation probabilities that we had to store, we first computed LLR association scores for all bilingual word pairs with a positive association ($p(t, s) > p(t) \cdot p(s)$), and discarded from further consideration those with an LLR score of less that 0.9, which was chosen to be just low enough to retain all the "sure" word alignments in the trial data. This resulted in 13,285,942 possible word-to-word translation pairs (plus 66,406 possible null-word-to-word pairs).

For most models, the word translation parameters are set automatically by EM. We trained each variation of each model for 20 iterations, which was enough in almost all cases to discern a clear minimum error on the 37 sentence pairs of trial data, and we chose as the preferred iteration the one with the lowest alignment error rate on the trial data. The other parameters of the various versions of Model 1 described in Sections 4–6 were optimized with respect to alignment error rate on the trial data using simple hill climbing. All the results we report for the 447 sentence pairs of test data use the parameter values set to their optimal values for the trial data.

We report results for four principal versions of Model 1, trained using English as the source language and French as the target language:

- The *standard* model is initialized using uniform distributions, and trained without smoothing using EM, for a number of iterations optimized on the trial data.
- The *smoothed* model is like the standard model, but with optimized values of the null-word weight and add-n parameter.
- The *heuristic* model simply uses the initial heuristic estimates of the translation parameter values, with an optimized LLR exponent and null-word weight, but no EM re-estimation.
- The *combined* model initializes the translation parameter values with the heuristic estimates, using the LLR exponent and null-word weight from the optimal heuristic model, and applies EM using optimized values of the null-word weight and add-n parameters. The null-word weight used during EM is optimized separately from the null-word weight used in the initial heuristic parameter estimates.

We also performed ablation experiments in which we ommitted each applicable modification in turn from each principal version of Model 1, to observe the effect on alignment error. All non-EM-trained

Model (Ablation)	Trial AER	Test AER	Test Recall	Test Precision	LLR Exp	Init NW	EM NW	Add n	EM Iter
Standard	**0.311**	**0.298**	**0.810**	**0.646**	**NA**	**NA**	**1.0**	**0.0000**	**17**
Smoothed	**0.261**	**0.271**	**0.646**	**0.798**	**NA**	**NA**	**10.0**	**0.0100**	**15**
(EM NW)	0.285	0.273	0.833	0.671	NA	NA	1.0	0.0100	20
(Add n)	0.302	0.300	0.638	0.751	NA	NA	13.0	0.0000	14
Heuristic	**0.234**	**0.255**	**0.655**	**0.844**	**1.3**	**2.4**	**NA**	**NA**	**NA**
(LLR Exp)	0.257	0.259	0.655	0.844	1.0	2.4	NA	NA	NA
(Init NW)	0.300	0.308	0.740	0.657	1.5	1.0	NA	NA	NA
Combined	**0.203**	**0.215**	**0.724**	**0.839**	**1.3**	**2.4**	**7.0**	**0.005**	**1**
(LLR Exp)	0.258	0.272	0.636	0.809	1.0	2.4	10.0	0.0035	3
(Init NW)	*0.197*	*0.209*	0.722	0.854	1.5	1.0	10.0	0.0005	1
(EM NW)	0.281	0.267	0.833	0.680	1.3	2.4	1.0	0.0080	8
(Add n)	0.208	0.221	0.724	0.826	1.3	2.4	8.0	0.0000	1

Table 1: Evaluation Results.

parameters were re-optimized on the trial data for each version of Model 1 tested, with the exception that the value of the LLR exponent and initial null-word weight in the combined model were carried over from the heuristic model.

8 Results

We report the performance of our different versions of Model 1 in terms of precision, recall, and alignment error rate (AER) as defined by Och and Ney (2003). These three performance statistics are defined as

$$\text{recall} = \frac{|A \cap S|}{|S|} \tag{5}$$

$$\text{precision} = \frac{|A \cap P|}{|A|} \tag{6}$$

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{7}$$

where S denotes the annotated set of sure alignments, P denotes the annotated set of possible alignments, and A denotes the set of alignments produced by the model under test.[2] We take AER, which is derived from F-measure, as our primary evaluation metric.

The results of our evaluation are presented in Table 1. The columns of the table present (in order) a description of the model being tested, the AER on the trial data, the AER on the test data, test data recall, and test data precision, followed by the optimal values on the trial data for the LLR exponent, the initial (heuristic model) null-word weight, the null-word weight used in EM re-estimation, the add-n parameter value used in EM re-estimation, and the number of iterations of EM. "NA" means a parameter is not applicable in a particular model.

Results for the four principal versions of Model 1 are presented in bold. For each principal version, results of the corresponding ablation experiments are presented in standard type, giving the name of each omitted modification in parentheses.[3] Probably the most striking result is that the heuristic model substantially reduces the AER compared to the standard or smoothed model, even without EM re-estimation. The combined model produces an additional substantial reduction in alignment error, using a single iteration of EM.

The ablation experiments show how important the different modifications are to the various models. It is interesting to note that the importance of a given modification varies from model to model. For example, the re-estimation null-word weight makes essentially no contribution to the smoothed model. It can be tuned to reduce the error on the trial data, but the improvement does not carry over to the test data. The smoothed model with only the null-word weight and no add-n smoothing has essentially the same error as the standard model; and the smoothed model with add-n smoothing alone has essentially the same error as the smoothed model with both the null-word weight and add-n smoothing. On the other hand, the re-estimation null-word weight is crucial to the combined model. With it, the combined model has substantially lower error than the heuristic model without re-estimation; without it, for any number of EM iterations, the combined model has higher error than the heuristic model.

[2] As is customary, alignments to the null word are not explicitly counted.

[3] Modificiations are "omitted" by setting the corresponding parameter to a value that is equivalent to removing the modification from the model.

A similar analysis shows that add-n smoothing is much less important in the combined model than the smoothed model. The probable explanation for this is that add-n smoothing is designed to address over-fitting from many iterations of EM. While the smoothed model does require many EM iterations to reach its minimum AER, the combined model, with or without add-n smoothing, is at its minimum AER with only one EM iteration.

Finally, we note that, while the initial null-word weight is crucial to the heuristic model without re-estimation, the combined model actually performs better without it. Presumably, the re-estimation null-word weight makes the inital null-word weight redundant. In fact, the combined model without the initial null word-weight has the lowest AER on both the trial and test data of any variation tested (note AERs in italics in Figure 1). The relative reduction in AER for this model is 29.9% compared to the standard model.

We tested the significance of the differences in alignment error between each pair of our principal versions of Model 1 by looking at the AER for each sentence pair in the test set using a 2-tailed paired t test. The differences between all these models were significant at a level of 10^{-7} or better, except for the difference between the standard model and the smoothed model, which was "significant" at the 0.61 level—that is, not at all significant. The reason for this is probably the very different balance between precision and recall with the standard and smoothed models, which indicates that the models make quite different sorts of errors, making statistical significance hard to establish. This conjecture is supported by considering the smoothed model omitting the re-estimation null-word weight, which has substantially the same AER as the full smoothed model, but with a precision/recall balance much closer to the standard model. The 2-tailed paired t test comparing this model to the standard model showed significance at a level of better than 10^{-10}. We also compared the combined model with and without the initial null-word weight, and found that the improvement without the weight was significant at the 0.008 level.

9 Conclusions

We have demonstrated that it is possible to improve the performance of Model 1 in terms of alignment error by about 30%, simply by changing the way its parameters are estimated. Almost half this improvement is obtained with a simple heuristic model that does not require EM re-estimation.

It is interesting to contrast our heuristic model with the heuristic models used by Och and Ney (2003) as baselines in their comparative study of alignment models. The major difference between our model and theirs is that they base theirs on the Dice coefficient, which is computed by the formula[4]

$$\frac{2 \cdot C(t,s)}{C(t) + C(s)} \tag{8}$$

while we use the log-likelihood-ratio statistic defined in Section 6. Och and Ney find that the standard version of Model 1 produces more accurate alignments after only one iteration of EM than either of the heuristic models they consider, while we find that our heuristic model outperforms the standard version of Model 1, even with an optimal number of iterations of EM.

While the Dice coefficient is simple and intuitive—the value is 0 for words never found together, and 1 for words always found together—it lacks the important property of the LLR statistic that scores for rare words are discounted; thus it does not address the over-fitting problem for rare words.

The list of applications of IBM word-alignment Model 1 given in Section 1 should be sufficient to convince anyone of the relevance of improving the model. However, it is not clear that AER as defined by Och and Ney (2003) is always the appropriate way to evaluate the quality of the model, since the Viterbi word alignment that AER is based on is seldom used in applications of Model 1.[5] Moreover, it is notable that while the versions of Model 1 having the lowest AER have dramatically higher precision than the standard version, they also have quite a bit lower recall. If AER does not reflect the optimal balance between precision and recall for a particular application, then optimizing AER may not produce the best task-based performance for that application. Thus the next step in this research must be to test whether the improvements in AER we have demonstrated for Model 1 lead to improvements on task-based performance measures.

[4] Och and Ney give a different formula in their paper, in which the addition in the denominator is replaced by a multiplication. According to Och (personal communication), however, this is merely a typographical error in the publication, and the results reported are for the standard definition of the Dice coefficient.

[5] A possible exception is suggested by the results of Koehn et al. (2003), which show that phrase translations extracted from Model 1 alignments can perform almost as well in a phrase-based statistical translation system as those extracted from more sophisticated alignment models, provided enough training data is used.

References

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

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Meredith J. Goldsmith, Jan Hajic, Robert L. Mercer, and Surya Mohanty. 1993b. But dictionaries are data too. In *Proceedings of the ARPA Workshop on Human Language Technology*, pp. 202–205, Plainsboro, New Jersey, USA.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 310–318, Santa Cruz, California, USA.

Yuan Ding, Daniel Gildea, and Martha Palmer. 2003. An algorithm for word-level alignment of parallel dependency trees. In *Proceedings of the Ninth Machine Translation Summit*, pp. 95–101, New Orleans, Louisiana, USA.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pp. 127–133, Edmonton, Alberta, Canada.

I. Dan Melamed. 2000. Models of Translational Equivalence. *Computational Linguistics*, 26(2):221–249.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop, Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pp. 1–6, Edmonton, Alberta, Canada.

Robert C. Moore. 2001. Towards a simple and accurate statistical approach to learning translation relationships among words. In *Proceedings of the Workshop Data-driven Machine Translation at the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 79–86, Toulouse, France.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In S. Richardson (ed.), *Machine Translation: From Research to Real Users* (Proceedings, 5th Conference of the Association for Machine Translation in the Americas, Tiburon, California), pp. 135–244, Springer-Verlag, Heidelberg, Germany.

Robert C. Moore. 2004. On log-likelihood-ratios and the significance of rare events. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

Dragos S. Munteanu, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, pp. 265–272, Boston, Massachusetts, USA.

Francisco Nevado, Francisco Casacuberta, and Enrique Vidal. 2003. Parallel corpora segmentation using anchor words. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resources and tools for building MT*, pp. 33–40, Budapest, Hungary.

Franz Joseph Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och et al. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, pp. 161–168, Boston, Massachusetts, USA.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 319–326, Sapporo, Japan.