In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 37–42, Lisbon, Portugal, 2000.

Incorporating Position Information into a Maximum Entropy/Minimum Divergence Translation Model

George Foster
RALI, Université de Montréal
foster@iro.umontreal.ca

Abstract

I describe two methods for incorporating information about the relative positions of bilingual word pairs into a Maximum Entropy/Minimum Divergence translation model. The better of the two achieves over 40% lower test corpus perplexity than an equivalent combination of a trigram language model and the classical IBM translation model 2.

1 Introduction

Statistical Machine Translation (SMT) systems use a model of $p(\mathbf{t}|\mathbf{s})$, the probability that a text $\mathbf{s}$ in the source language will translate into a text $\mathbf{t}$ in the target language, to determine the best translation for a given source text. A straightforward way of modeling this distribution is to apply a chain-rule expansion of the form:

$$p(\mathbf{t}|\mathbf{s}) = \prod_{i=1}^{|\mathbf{t}|} p(t_i|t_1 \dots t_{i-1}, \mathbf{s}), \qquad (1)$$

where t_i denotes the ith token in $\mathbf{t}$.[1] The objects to be modeled in this case belong to the family of conditional distributions $p(w|\mathbf{h}_i, \mathbf{s})$, the probability of the ith word in $\mathbf{t}$, given the tokens which precede it and the source text.

The main motivation for modeling $p(\mathbf{t}|\mathbf{s})$ in terms of $p(w|\mathbf{h}_i, \mathbf{s})$ is that it simplifies the "decoding" problem of finding the most likely target text. In particular, if $\mathbf{h}_i$ is known, finding the best word at the current position requires only a straightforward search through the target vocabulary, and efficient dynamic-programming based heuristics can be used to extend this to sequences of words. This is very important for applications such as TransType (Foster et al., 1997; Langlais et al., 2000), where the task is to make real-time predictions of the text a human translator will type next, based on the source text under translation and some prefix of the target text that has already been typed. The standard "noisy channel" approach used in SMT, where $p(\mathbf{t}|\mathbf{s}) \propto p(\mathbf{t})p(\mathbf{s}|\mathbf{t})$, is generally too expensive for such applications because it does not permit direct calculation of the probability of a word or sequence of words beginning at the current position. Complex and expensive search strategies are required to find the best target text in this approach (García-Varea et al., 1998; Niessen et al., 1998; Och et al., 1999; Wang and Waibel, 1998).

The challenge in modeling $p(w|\mathbf{h}_i, \mathbf{s})$ is to combine two disparate sources of conditioning information in an effective way. One obvious strategy is to use a linear combination of separate language and translation components, of the form:

$$p(w|\mathbf{h}_i, \mathbf{s}) = \lambda p(w|\mathbf{h}_i) + (1-\lambda)p(w|i, \mathbf{s}). \qquad (2)$$

where $p(w|\mathbf{h}_i)$ is a language model, $p(w|i, \mathbf{s})$ is a translation model, and $\lambda \in [0, 1]$ is a combining weight. However, this appears to be a weak technique (Langlais and Foster, 2000), even when λ is allowed to depend on various features of the context $(\mathbf{h}_i, \mathbf{s})$.

In previous work (Foster, 2000), I described a Maximum Entropy/Minimum Divergence (MEMD) model (Berger et al., 1996) for $p(w|\mathbf{h}_i, \mathbf{s})$ which incorporates a trigram language model and a translation component which is an analog of the well-known IBM translation model 1 (Brown et al., 1993). This model

[1] This ignores the issue of normalization over target texts of all possible lengths, which can be easily enforced when desired by using a stop token or a prior distribution over lengths.

significantly outperforms an equivalent linear combination of a trigram and model 1 in test-corpus perplexity, despite using several orders of magnitude fewer translation parameters. Like model 1, its translation component is based only on the occurrences in $\mathbf{s}$ of words which are potential translations for w, and does not take into account the positions of these words relative to w. An obvious enhancement is to incorporate such positional information into the MEMD model, thereby making its translation component analogous to the IBM model 2. This is the problem I address in this paper.

2 Models

2.1 Linear Model

As a baseline for comparison I used a linear combination as in (2) of a standard interpolated trigram language model and the IBM translation model 2 (IBM2), with the combining weight λ optimized using the EM algorithm. IBM2 is derived as follows:[2]

$$\begin{aligned} p(w|i,\mathbf{s}) &= \sum_{j=0}^{l} p(w,j|i,\mathbf{s}) \\ &\approx \sum_{j=0}^{l} p(w|s_j)p(j|i,l) \end{aligned}$$

where $l = |\mathbf{s}|$, and the hidden variable j gives the position in $\mathbf{s}$ of the (single) source token s_j assumed to give rise to w, or 0 if there is none. The model consists of a set of word-pair parameters $p(t|s)$ and position parameters $p(j|i,l)$; in model 1 (IBM1) the latter are fixed at $1/(l+1)$, as each position, including the empty position 0, is considered equally likely to contain a translation for w. Maximum likelihood estimates for these parameters can be obtained with the EM algorithm over a bilingual training corpus, as described in (Brown et al., 1993).

2.2 MEMD Model 1

A MEMD model for $p(w|\mathbf{h}_i,\mathbf{s})$ has the general form:

$$p(w|\mathbf{h}_i,\mathbf{s}) = \frac{q(w|\mathbf{h}_i,\mathbf{s})\exp(\vec{\alpha}\cdot\mathbf{f}(w,\mathbf{h}_i,\mathbf{s}))}{Z(\mathbf{h}_i,\mathbf{s})},$$

where $q(w|\mathbf{h}_i,\mathbf{s})$ is a reference distribution, $\mathbf{f}(w,\mathbf{h}_i,\mathbf{s})$ maps $(w,\mathbf{h}_i,\mathbf{s})$ into an n-dimensional feature vector, $\vec{\alpha}$ is a corresponding vector of feature weights (the parameters of the model), and $Z(\mathbf{h}_i,\mathbf{s}) = \sum_w q(w|\mathbf{h}_i,\mathbf{s})\exp(\vec{\alpha}\cdot\mathbf{f}(w,\mathbf{h}_i))$ is a normalizing factor. For a given choice of q and $\mathbf{f}$, the IIS algorithm (Berger et al., 1996) can be used to find maximum likelihood values for the parameters $\vec{\alpha}$. It can be shown (Della Pietra et al., 1995) that these are the also the values which minimize the Kullback-Liebler divergence $D(p||q)$ between the model and the reference distribution under the constraint that the expectations of the features (ie, the components of $\mathbf{f}$) with respect to the model must equal their expectations with respect to the empirical distribution derived from the training corpus. Thus the reference distribution serves as a kind of prior, and should reflect some initial knowledge about the true distribution; and the use of any feature is justified to the extent that its empirical expectation is accurate.

In the present context, the natural choice for the reference distribution q is a trigram language model. To create a MEMD analog to IBM model 1 (MEMD1), I used boolean features corresponding to bilingual word pairs:

$$f_{st}(w,\mathbf{s}) = \begin{cases} 1, & s \in \mathbf{s} \text{ and } t = w \\ 0, & \text{else} \end{cases}$$

where (s,t) is a (source,target) word pair. Using the notational convention that α_{st} is 0 whenever the corresponding feature f_{st} does not exist in the model, MEMD1 can be written compactly as:

$$p(w|\mathbf{h}_i,\mathbf{s}) = q(w|\mathbf{h}_i)\exp(\sum_{s\in\mathbf{s}}\alpha_{sw})/Z(\mathbf{h}_i,\mathbf{s}).$$

Due to the theoretical properties of MEMD outlined above, it is necessary to select a subset of all possible features f_{st} to avoid overfitting the training corpus. Using a reduced feature set is also computationally advantageous, since the time taken to calculate the normalization constant $Z(\mathbf{h}_i,\mathbf{s})$ grows linearly with the expected number of features which are active per source word $s \in \mathbf{s}$. This is in contrast to IBM1, where use of all available word-pair parameters $p(t|s)$ is standard, and engenders only a very slight overfitting effect. In (Foster, 2000) I describe an

[2] Model 2 was originally formulated for $p(\mathbf{t}|\mathbf{s})$, but since target words are predicted independently it can also be used for $p(w|\mathbf{h}_i,\mathbf{s})$. The only necessary modification in this case is that the position parameters can no longer be conditioned on $|\mathbf{t}|$.

effective technique for selecting MEMD word-pair features.

2.3 MEMD Model 2

IBM2 incorporates position information by introducing a hidden position variable and making independence hypotheses. This approach is not applicable to MEMD models, whose features must capture events which are directly observable in the training corpus.[3] It would be possible to use pure position features of the form f_{ijl}, which capture the presence of *any* word pair at position (i, j, l) and are superficially similar to IBM2's position parameters, but these would add almost no information to MEMD1. On the other hand, features like f_{stijl}, indicating the presence of a specific pair (s, t) at position (i, j, l), would cause severe data sparseness problems.

Encoding Positions as Feature Values

A simple solution to this dilemma is to let the value of a word-pair feature reflect the current position of the pair rather just its presence or absence. A reasonable choice for this is the value of the corresponding IBM2 position parameter $p(j|i, l)$:

$$f_{st}(w, i, \mathbf{s}) = \begin{cases} p(\hat{\jmath}_s|i, l), & s \in \mathbf{s} \text{ and } t = w \\ 0, & \text{else} \end{cases}$$

where $\hat{\jmath}_s$ is the position of s in $\mathbf{s}$, or the most likely position according to IBM2 if it occurs more than once: $\hat{\jmath}_s = \text{argmax}_{j:s_j=s}\, p(j|i, l)$. Using the same convention as in the previous section, the resulting model (MEMD2R) can be written:

$$p(w|\mathbf{h}_i, \mathbf{s}) = \frac{q(w|\mathbf{h}_i) \exp(\sum_{s \in \mathbf{s}} \alpha_{sw} p(\hat{\jmath}_s|i, l))}{Z(\mathbf{h}_i, \mathbf{s})}.$$

MEMD2R is simple and compact but poses a technical difficulty due to its use of real-valued features, in that the IIS training algorithm requires integer or boolean features for efficient implemention. Since likelihood is a concave function of $\vec{\alpha}$, any hillclimbing method such as gradient ascent[4] is guaranteed to find maximum likelihood parameter values, but convergence is slower than IIS and requires tuning a gradient step parameter. Unfortunately, apart from this problem, MEMD2R also turns out to perform slightly worse than MEMD1, as described below.

Using Class-based Position Features

Since the basic problem with incorporating position information is one of insufficient data, a natural solution is to try to group word pair and position combinations with similar behaviour into classes such that the frequency of each class in the training corpus is high enough for reliable estimation. To do this, I made two preliminary assumptions: 1) word pairs with similar MEMD1 weights should be grouped together; and 2) position configurations with similar IBM2 probabilities should be grouped together. This converts the problem from one of finding classes in the five-dimensional space (s, t, i, j, l) to one of identifying rectangular areas on a 2-dimensional grid where one axis contains position configurations (i, j, l), ordered by $p(j|i, l)$; and the other contains word pairs (s, t), ordered by α_{st}. To simplify further, I partitioned both axes so as to approximately balance the total corpus frequency of all word pairs or position configurations within each partition. Thus the only parameters required to completely specify a classification are the number of position and word-pair partitions. Each combination of a position partition and a word pair partition corresponds to a class, and all classes can be expected to have roughly the same empirical counts.

The model (MEMD2B) based on this scheme has one feature for each class; if A designates the set of triples (i, j, l) in a position partition and B designates the set of pairs (s, t) in a word-pair partition, then for all A, B there is a feature:

$$\begin{aligned} f_{A,B}(w, i, \mathbf{s}) \;\; = \sum_{j=1}^{l} \;\; & \delta[(i, j, l) \in A \;\wedge \\ & (s_j, w) \in B \;\wedge \\ & j = \hat{\jmath}_{s_j}], \end{aligned}$$

where $\delta[X]$ is 1 when X is true and 0 otherwise. For robustness, I used these position features along with pure MEMD1-style word-pair features f_{st}. The weights $\alpha_{A,B}$ on the position features can thus be interpreted as correction terms for the pure word-pair weights $\alpha_{s,t}$ which

[3] Although it is possible to extend the basic framework to allow for embedded Hidden Markov Models (Lafferty, 1995).

[4] I found that the "stochastic" variant of this algorithm, in which model parameters are updated after each training example, gave the best performance.

segment	file pairs	sentence pairs	English tokens	French tokens
train	922	1,639,250	29,547,936	31,826,112
held-out 1	30	54,758	978,394	1,082,350
held-out 2	30	59,435	1,111,454	1,241,581
test	30	53,676	984,809	1,103,320

Table 1: Corpus segmentation. The *train* segment was the main training corpus; the *held-out 1* segment was used for combining weights for the trigram and the overall linear model; and the *held-out 2* segment was used for the MEMD2B partition search.

reflect the proximity of the words in the pair. The model is:

$$p(w|\mathbf{h}_i, \mathbf{s}) = \frac{q(w|\mathbf{h}_i)\exp(\sum_{s\in\mathbf{s}}\alpha_{sw} + \alpha_{A(i,\hat{\jmath}_s,l),B(s,t)})}{Z(\mathbf{h}_i,\mathbf{s})},$$

where $A(i, \hat{\jmath}_s, l)$ gives the partition for the current position, $B(s,t)$ gives the partition for the current word pair, and following the usual convention, $\alpha_{A(i,\hat{\jmath}_s,l),B(s,t)}$ is zero if these are undefined.

To find the optimal number of position partitions m and word-pair partitions n, I performed a greedy search, beginning at a small initial point (m, n) and at each iteration training two MEMD2B models characterized by (km, n) and (m, kn), where $k > 1$ is a scaling factor (note that both these models contain kmn position features). The model which gives the best performance on a validation corpus is used as the starting point for the next iteration. Since training MEMD models is very expensive, to speed up the search I relaxed the convergence criterion from a training corpus perplexity[5] drop of $< .1\%$ (requiring 20-30 IIS iterations) to $< .6\%$ (requiring approximately 10 IIS iterations). I stopped the search when the best model's performance on the validation corpus did not decrease significantly from that of the model at the previous step, indicating that overtraining was beginning to occur.

3 Results

I tested the models on the Canadian Hansard corpus, with English as the source language and French as the target language. After sentence alignment using the method described in (Simard et al., 1992), the corpus was split into disjoint segments as shown in table 1. To evaluate performance, I used perplexity: $p(\mathcal{T}|\mathcal{S})^{-1/|\mathcal{T}|}$, where p is the model being evaluated, and $(\mathcal{S}, \mathcal{T})$ is the test corpus, considered to be a set of statistically independent sentence pairs $(\mathbf{s}, \mathbf{t})$. Perplexity is a good indicator of performance for the TransType application described in the introduction, and it has also been used in the evaluation of full-fledged SMT systems (Al-Onaizan et al., 1999). To ensure a fair comparison, all models used the same target vocabulary. For all MEMD models, I used 20,000 word-pair features selected using the method described in (Foster, 2000); this is suboptimal but gives reasonably good performance and facilitates experimentation.

Figures 1 and 2 show, respectively, the path taken by the MEMD2B partition search, and the validation corpus perplexities of each model tested during the search. As shown in figure 1, the search consisted of 6 iterations. Since on all previous iterations no increase in position partitions beyond the initial value of 10 was selected, on the 5th iteration I tried decreasing the number of position partitions to 5. This model was not selected either, so on the final step only the number of word-pair partitions was augmented, yielding an optimal combination of 10 position partitions and 4000 word-pair partitions.

Table 2 gives the final results for all models. The IBM models tested here incorporate a reduced set of 1M word-pair parameters, selected using the method described in (Foster, 2000), which gives slightly better test-corpus performance than the unrestricted set of all 35M word pairs which cooccur within aligned sentence pairs in the training corpus.

The basic MEMD1 model (without position parameters) attains about 30% lower perplexity than the model 2 baseline, and MEMD2B with an optimal-sized set of position parameters achieves in a further drop of over 10%. Interestingly, the difference between IBM1 and

[5] Defined in the next section

model	word-pair parameters	position parameters	perplexity	improvement over baseline
trigram	0	0	61.0	—
trigram + IBM1	1,000,000	0	43.2	—
trigram + IBM2	1,000,000	115,568	35.2	0%
MEMD1	20,000	0	24.5	30.4%
MEMD2R	20,000	0	28.4	19.3%
MEMD2B	20,000	10×10	22.1	37.2%
MEMD2B	20,000	10×4000	20.2	42.6%

Table 2: Model performances. Linear interpolation is designated with a + sign; and the MEMD2B position parameters are given as $m\times$, where m and n are the numbers of position partitions and word-pair partitions respectively.

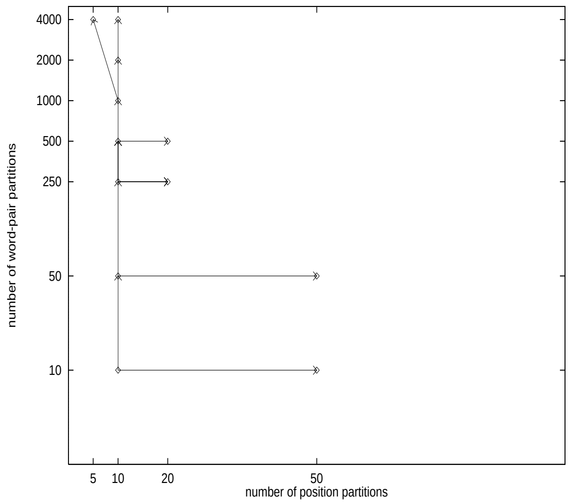

Figure 1: MEMD2B partition search path, beginning at the point $(10, 10)$. Arrows out of each point show the configurations tested at each iteration.

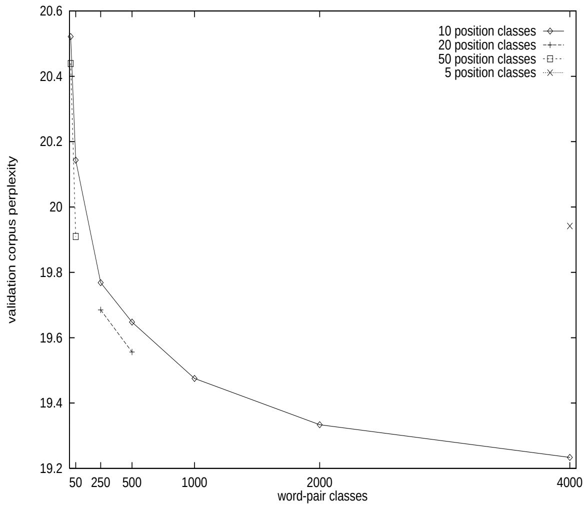

Figure 2: Validation corpus perplexities for various MEMD2B models. Each connected line in this graph corresponds to a vertical column of search points in figure 1.

IBM2's performance (18.5% lower perplexity for IBM2) is about the same as the difference between MEMD1 and MEMD2B (17.6% lower for MEMD2B).

4 Conclusion

This paper deals with the problem of incorporating information about the positions of bilingual word pairs into a MEMD model which is analogous to the classical IBM model 1, thereby creating a MEMD analog to the IBM model 2. I proposed and evaluated two methods for accomplishing this: using IBM2 position parameter probabilities as MEMD feature values, which was unsuccessful; and adding features which capture the occurrence of a word-pair with a MEMD1 weight that falls into a specific range of values at a position to which IBM2 assigns a probability in a certain range. The second model achieved over 40% lower test perplexity than a linear combination of a trigram and IBM2, despite using several orders of magnitude fewer parameters.

This work represents a novel approach to translation modeling which is most appropriate for applications like TransType which need to make rapid predictions of upcoming text. However, it is not inconceivable that it could also be used for full-fledged MT. One partial impediment to this is that the MEMD framework lacks a mechanism equivalant to the EM algorithm for estimating probabilities associated with hidden variables. The solution I have proposed here can be seen as a first step to investigating ways of getting around this problem.

Acknowledgements

This work was carried out as part of the TransType project at RALI, funded by the Natural Sciences and Engineering Research Council of Canada.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation: Final report, JHU workshop 1999. Technical report, The Center for Language and Speech Processing, The Johns Hopkins University, www.clsp.jhu.edu/ws99/projects/mt/final_report.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993. The mathematics of Machine Translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June.

S. Della Pietra, V. Della Pietra, and J. Lafferty. 1995. Inducing features of random fields. Technical Report CMU-CS-95-144, CMU.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194.

George Foster. 2000. A Maximum Entropy / Minimum Divergence translation model. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-38)*, Hong Kong, October.

Ismael García-Varea, Francisco Casacuberta, and Hermann Ney. 1998. An iterative, DP-based search algorithm for statistical machine translation. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP) 1998*, Sydney, Australia, December. pages 1135–1138.

John D. Lafferty. 1995. Gibbs-markov models. In *Computing Science and Statistics: Proceedings of the 27th Symposium on the Interface*. Interface Foundation.

Ph. Langlais and G. Foster. 2000. Using context-dependent interpolation to combine statistical language and translation models for interactive MT. In *Content-Based Multimedia Information Access (RIAO)*, Paris, France, April.

Ph. Langlais, G. Foster, and G. Lapalme. 2000. Unit completion for a computer-aided translation typing system. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP-5)*, Seattle, Washington, May.

S. Niessen, S. Vogel, H. Ney, and C. Tillmann. 1998. A DP based search algorithm for statistical machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics (COLING) 1998*, pages 960–967, Montréal, Canada, August.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the 4nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, College Park, Maryland.

Michel Simard, George F. Foster, and Pierre Isabelle. 1992. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Montréal, Québec.

Ye-yi Wang and Alex Waibel. 1998. Fast decoding for statistical machine translation. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP) 1998*, Sydney, Australia, December, pages 2775–2778.