Evaluating a Maximum Entropy Translation Model

George Foster

March 29, 1999

Paper ID Code:
Submission Type: General Session
Topic Areas or Theme ID: Statistical Modeling, Maximum Entropy, Machine Translation
Word Count: 2742
Under consideration for other conferences (specify)? None

Abstract

I present empirical comparisons between a standard statistical translation model and an equivalent Maximum Entropy model. Results show that the Maximum Entropy model is promising, but highly sensitive to the method of feature selection.

1 Introduction

The technique of Maximum Entropy (ME) statistical modeling is a fairly recent and increasingly popular addition to the empirical toolkit of computational linguists. It has been successfully applied to a wide variety of problems, including language modeling (Beeferman et al., 1997a; Berger and Printz, 1998; Rosenfeld, 1996), machine translation (Berger et al., 1996), parsing (Ratnaparkhi, 1997; Skut and Brants, 1998), sentence boundary identification (Reynar and Ratnaparkhi, 1997), part-of-speech tagging (Ratnaparkhi, 1996), and text segmentation (Beeferman et al., 1997b).

The basic ME concept is easy to grasp and intuitively appealing: the modeler picks certain characteristics of the data which are considered to reflect the true distribution, then automatically derives the highest entropy model which possesses those characteristics. The idea is to avoid unwarranted assumptions by using the most "random" of the distributions that conform to the

known facts. As a means of creating a statistical model of natural language, this seems more elegant and less constraining than traditional techniques based on independence hypotheses.

I am interested in the problem of creating a statistical translation model for use in an interactive machine translation system (Foster et al., 1997). The main challenge is to combine two very different sources of predictive information about upcoming target text: the source text under translation and the part of the target text which has already been established. The classical solution to this problem is a noisy channel setup involving separate translation and language models to capture each of these sources of information (Brown et al., 1993). One drawback of a noisy channel is that it makes searching for the most likely target text complicated and expensive[1]—for my application, where real-time performance is crucial, this is highly undesirable.

ME provides an attractive solution to this problem, due to its ability to combine disparate sources of information in a principled way within a single homogeneous framework. My ultimate goal is to evaluate a ME replacement for the noisy channel model. The work described in this paper is a first step toward this goal: I evaluate a ME translation model which is directly comparable to the IBM model 1 (Brown et al., 1993). To my knowledge this is the first implementation of a full-scale ME translation model, and the first attempt at a rigorous comparison between a ME model and an equivalent non-ME model based on exactly the same sources of information.[2]

[1]Basically, this is because the translation model is used "backward" to predict source text given target text. This precludes efficient dynamic-programming search techniques unless special measures are taken, eg (Wu, 1996; Tillmann et al., 1997).

[2]Rosenfeld (1996) reports a greater perplexity reduction (23% versus 10%) over a baseline trigram language model due to long-distance word-pair predictions in a ME framework compared with the same predictions incorporated using linear interpolation. However, since the two models tested apparently differed in other aspects, it is impossible to determine how much of this gain can be attributed to the use of ME.

2 Description of Models

In this section I describe three basic translation models used in my experiments.

2.1 IBM Model 1

The first model is the IBM model 1 (IBM1) (Brown et al., 1993). This gives an estimate of the probability $p(\mathbf{t}|\mathbf{s})$ that a segment of source text $\mathbf{s}$ in one language will translate to a segment of target text $\mathbf{t}$ in another language. A major assumption of this model is that tokens in the target text are independent, ie that $p(\mathbf{t}|\mathbf{s})$ can be expressed as a product of terms $p(t|\mathbf{s})$ which give the probability that the word t will appear somewhere in the translation of $\mathbf{s}$.[3] The latter distribution is the one I will focus on in this paper. IBM1 models it using a two step generative process: 1) pick a token s at random in $\mathbf{s}$; 2) choose t according to a word-for-word translation probability $p(t|s)$. Summing over all choices for s gives the complete model:

$$p(t|\mathbf{s}) = \sum_{i=1}^{n} p(t|s_i)/n \tag{1}$$

where s_i is the ith token in $\mathbf{s}$.[4] The translation parameters $p(t|s)$ can be estimated from a bilingual corpus of aligned segment pairs using the EM algorithm, as described in (Brown et al., 1993).

[3] Ignoring the term needed to ensure that $p(\mathbf{t}|\mathbf{s})$ is normalized over all lengths of $\mathbf{t}$, which is irrelevant here.

[4] IBM1 as originally defined includes a "null" source token to account for target words with no clear 1-1 link to the source text. I have omitted this from my version for the sake of simplicity.

2.2 An Empirical Baseline Model

As a baseline for comparison, I used a model (E1) whose structure is identical to that of IBM1, but whose parameters were derived directly from co-occurrence counts in the training corpus instead of being estimated iteratively with the EM algorithm. The conditional probability of each word pair was calculated from its joint relative frequency among the token pairs generated by taking the Cartesion product over each aligned segment pair in the corpus. The theoretical properties of E1 are dubious, but it is useful as a baseline because any reasonable model ought to be able to do better.

2.3 A Maximum Entropy Translation Model

Since the ME method has been well described elsewhere, eg (Berger et al., 1996), I will assume a basic familiarity on the part of the reader and not reproduce all the details here.

The starting point for any ME model is a set of *features*: functions over the event space of interest whose expected values with respect to the true distribution are known. To create ME models based on the same information as IBM1, I used features which pick out occurrences of bilingual word pairs. For a particular word pair st, the corresponding feature is:

$$f_{st}(t', \mathbf{s}) = \begin{cases} \#\text{ of times } s \text{ occurs in } \mathbf{s}, & t = t' \\ 0, & \text{else} \end{cases}$$

This directly mirrors the way IBM1 uses a contribution from each source token to predict a target word in (1).

The next step in the modeling process is to determine the "true" expected value for each

feature. Following standard practice, I derived these from the training corpus. Each aligned segment pair $\mathbf{t}$,$\mathbf{s}$ in the corpus was split into pairs of the form $t, \mathbf{s}$, one for each target token t in $\mathbf{t}$. Defining the empirical probability $\tilde{p}(t, \mathbf{s})$ to be the relative frequency of $t, \mathbf{s}$ among all such pairs, the expected value of feature f_{st} is:

$$\tilde{p}(f_{st}) = \sum_{t', \mathbf{s}} \tilde{p}(t', \mathbf{s}) f_{st}(t', \mathbf{s}).$$

The last step in creating a model is parameter estimation, to obtain the maximum entropy model for which the expected value of each feature f_{st} is $\tilde{p}(f_{st})$. According to ME theory, this takes the form:

$$p(t', \mathbf{s}) = \exp(\sum_{st} \lambda_{st} f_{st}(t', \mathbf{s}))/Z \tag{2}$$

where Z is a normalization constant (the sum over all $t', \mathbf{s}$ of the numerator), and each λ_{st} is a parameter which gives a weight for the corresponding feature. To find the parameter settings for which the model has the desired expected values, I used the IIS algorithm (Berger et al., 1996), along with the usual marginal constraint $p(\mathbf{s}) = \tilde{p}(\mathbf{s}),\ \forall \mathbf{s}$, to avoid having to sum over all $t, \mathbf{s}$ when calculating Z.

The final translation model (ME1) is the conditional distribution derived from (2):

$$p(t|\mathbf{s}) = \exp(\sum_{s \in \mathbf{s}} \lambda_{st})/Z(\mathbf{s}) \tag{3}$$

where the sum is over all tokens in $\mathbf{s}$, with the convention that $\lambda st = 0$ if no feature f_{st} exists. $Z(\mathbf{s})$ is the analog of Z for the joint distribution, fortunately a much more tractable sum over just the words in the target vocabulary. It is interesting to compare equations (1) and (3)—if

normalization is ignored, IBM1 and ME1 have identical structure except for the fact that ME uses its sum over word-pair scores as an exponent, which will tend to amplify the effect of both high and low scores.

3 Experiments

I conducted a series of experiments with the three basic models described in the previous section on the Canadian Hansard corpus, with English as the source language and French as the target language. After tokenization and segmentation into sentences, the corpus was aligned using the method described in (Simard et al., 1992), then, to improve its quality, filtered to retain only 1-1 sentence alignments containing 40 or fewer words per side. The results comprised about 30M words in English and 33M words in French. Three separate Hansard files containing a total of approximately 70k words in each language were reserved for testing.

To evaluate the models' performance, I used the *perplexity* measure common in speech recognition. This is a geometric average over tokens, $P^{-1/T}$, where P is the total probability the model assigns to a text containing T tokens. Perplexity is a good indicator of performance for my interactive translation application, where the task is to predict upcoming target text.

To control for out-of-vocabulary words, each model uses a finite vocabulary (in each language) consisting only of words which appeared in its training corpus, plus one special unknown word *UNK*. During training, any word with frequency 1 in the corpus was mapped to UNK. During testing, any word not found in the vocabulary was mapped to UNK, and the probability of each UNK token was divided by the total number of out-of-vocabulary words in the text. To deal with a few rare zero-probability tokens, IBM1 and E1 were smoothed by interpolating them with a uniform distribution, with a coefficient of .9999; this did not significantly alter the

results on texts which did *not* contain zero-probability tokens.

3.1 Tests with "Full" Models

In section 2.3 it was left unspecified which bilingual word pairs st would have corresponding features f_{st} in ME1. For exact parity with IBM1 the obvious choice is to use all pairs which co-occur within aligned segments in the training corpus. Unfortunately, the resulting model takes much too long to train on any realistic-sized corpus; on a Sparc Ultra, I estimate my current implementation would take about 50 days to train on the whole Hansard corpus, compared with less than one day for IBM1. It is not clear that this can be improved upon significantly. The basic problem is that there are too many features active for each target token; this defeats techniques like cluster expansion (Lafferty and Suhm, 1995) for speeding up IIS. It may be possible to factor out computations common to many source segments, but the algorithm to do this is not obvious.

Because of this difficulty, I limited my tests of the "full" ME1 model to a small corpus, as shown in table 1. The large discrepancy between the scores assigned to the training and test texts shown here is a classic symptom of overtraining, which is corroborated by the fact that the weakest model, E1, does best on the test set. This is hardly surprising given the size of the training set. What is interesting is that ME1 does significantly better on the training set than IBM1. Since these two models have exactly the same number of parameters, it is quite likely that ME1 is in fact a more powerful model, and that, given a sufficiently large training corpus, its performance on the test set would be better than that of IBM1. Figure 1 contains a plot of performance versus corpus size for all three models. Unfortunately, in the very small range of corpus sizes tested here, no significant difference appears between IBM1 and ME1.

model	training corpus	test corpus
E1	184	361
IBM1	70	367
ME1	48	764

Table 1: Perplexities for "full" models trained on 10-file-pair corpus (token counts: 352k English, 391k French).

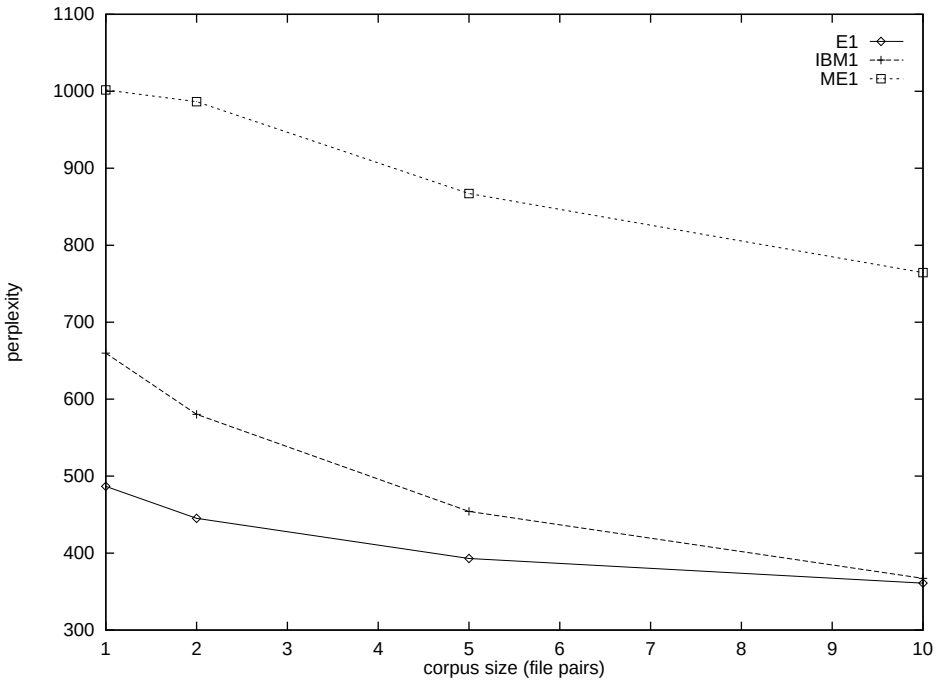

Figure 1: Test set perplexities versus training corpus size for "full" models.

3.2 Tests with Reduced Models

The version of ME1 described in the previous section is an unusual ME model in that it includes all available features rather than relying a small subset of particularly informative ones. To compare a more typical ME model, as well as to be able to use all of the training text available, I ran a second set of tests using two versions of ME1 with reduced feature sets.

The usual approach to feature selection involves a heuristic search over the feature space (Berger and Printz, 1998). Instead of doing this, I simply chose the most likely translations for each source word according to IBM1. As table 2 shows, these translations are fairly good, so intuitively this strategy is a reasonable one. For two different values of a threhold τ—.05 and .02—I created ME1 models whose feature sets were limited to word pairs st for which $p(t|s) \geq \tau$ according to IBM1. This resulted in average numbers of target words per source word of 5.1 and 12.5 respectively, compared to 681.1 for an unfiltered model. Both models were then trained on the whole training corpus. To ensure a fair comparison, I created parallel versions of E1 and IBM1 with renormalized distributions limited to these sets of word pairs.

The results are shown in table 3, with results from the unfiltered E1 and IBM1 models in the last column for comparison. Although the restrictions to highly probable translations hurt all three models, ME1's performance suffered most. This is somewhat surprising in light of the fact that it was the only model specifically trained with the reduced feature sets (rather than just being truncated and renormalized); even more surprising is the fact that performance on the test set was actually worse than the version from the previous section despite the fact that the latter was trained on a corpus that is almost 100 times smaller.

the	le/0.16 la/0.14 de/0.11 l'/0.07 ./0.06
walk	se/0.08 marcher/0.07 promener/0.04 pied/0.03 aller/.03
same	même/0.59 mêmes/0.11 chose/0.04 la/0.03 aussi/0.02
car	voiture/0.27 automobile/0.14 automobiles/0.07 voitures/0.05 d'/0.04
canada	canada/0.618526 du/0.10212 au/0.0969796 le/0.0443898 canadienne/0.0170081

Table 2: Top 5 translations according to IBM1 for randomly selected source words.

model	threshold		
	.05	.02	0
E1	1877	710	287
IBM1	1629	578	121
ME1	2037	1269	-

Table 3: Test corpus perplexities for reduced models. The *threshold* headings indicate the minimum probability thresholds used with IBM1 to compile lists of valid translations for each source word.

4 Discussion

The results in section 3.2 are striking. I have not yet had time to analyze the cause of ME1's poor performance in detail, but it seems most likely to be due to a bad choice of features. In fact, excluding the possibility of a bug, the only other alternative is that the ME method is inherently weak on this problem, which the results in section 3.1 would seem to contradict. Nevertheless, the size of the discrepancy between ME1 and both TM1 and E1 is surprising. The lesson to be learned seems to be that ME performance is very sensitive to the choice of features, at least for translation models. In future research, I plan to investigate this using a more standard method of feature selection.

5 Conclusion

I have presented a comparison between the IBM model 1 and an equivalent ME translation model for two different sets of word-pair parameters. On the full set of parameters normally used by model 1, the ME model assigned significantly lower perplexity to a small training

corpus, which indicates that it may be a more powerful model than model 1, despite assigning significantly *higher* perplexity to a test corpus. On a reduced set of parameters obtained by truncating model 1 with a probability threshold, the ME model trained on a large corpus performed much worse than model 1. Taken together with the previous result, this seems to indicate that the ME method is very sensitive to feature selection.

References

1997. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid, Spain, July.

Doug Beeferman, Adam Berger, and John Lafferty. 1997a. A model of lexical attraction and repulsion. In ACL-35 (ACL, 1997).

Doug Beeferman, Adam Berger, and John Lafferty. 1997b. Text segmentation using exponential models. In EMNLP-2 (EMN, 1997).

Adam Berger and Harry Printz. 1998. A comparison of criteria for maximum entropy / minimum divergence feature selection. In *Proceedings of the 3nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 97–106, Granada, Spain.

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.

1997. *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Providence, Rhode Island.

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

John D. Lafferty and Bernhard Suhm. 1995. Cluster expansions and iterative scaling for maximum entropy language models. In K. Hanson and R. Silver, editors, *Maximum Entropy and Bayesian Methods*. Kluwer.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 133–142.

Adwait Ratnaparkhi. 1997. A linear observed time statistical parser based on maximum entropy models. In EMNLP-2 (EMN, 1997).

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, Washinton, D.C.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech and Language*, 10:187–228.

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.

Wojciech Skut and Thorsten Brants. 1998. A Maximum Entropy partial parser for unrestricted text. In *Proceedings of the 6th ACL Workshop on Very Large Corpora (WVLC)*, pages 143–151, Montréal, Canada.

Christoph Tillmann, Stephan Vogel, Hermann Ney, and Alex Zubiaga. 1997. A dp-based search using monotone alignments in statistical translation. In ACL-35 (ACL, 1997), pages 289–296.

Dekai Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 152–158, Santa Cruz, California, September.