

Chapter 8

Conclusion

Maximum entropy methods can combine constraints from different information sources into one unified framework. Maximum entropy models are accurate, smooth and efficient in size. In this dissertation, we have shown these advantages of maximum entropy methods via three examples of language modeling. Substantial improvement in speech recognition has been achieved by using maximum entropy models with both collocational and long-distance dependencies. The intensive computational load in estimating the parameters of maximum entropy models has also been decreased. In this chapter, we summarize the major contributions, both theoretical and experimental, and discuss the future extensions and other potential applications of this work.

8.1 Contributions and Summary

8.1.1 Accuracy

We show that maximum entropy models improve accuracy in two different large vocabulary speech recognition tasks, Switchboard and the Broadcast News. The maximum entropy models with non-local dependencies achieve a substantial improvement compared to the baseline back-off models. We have presented several kinds of maximum entropy language model and compared them with corresponding interpolated and back-off models with the same constraints. Maximum entropy models almost al-

ways achieve slightly but consistently better results than the latter. The comparison between maximum entropy trigram models and back-off models has been presented in Sections 4.3 and 4.5. The comparison between maximum entropy topic-dependent models and corresponding interpolated models has been presented in Sections 4.4.1 and 4.5.2, and that between maximum entropy syntactic models and corresponding interpolated models in Sections 4.5.2 and 5.8.2.

We also show that the high accuracy of maximum entropy models can be obtained with insufficient training data. We trained a maximum entropy model with 800K model parameters using only 2.1 million words of training data for Switchboard, and a model with 5M parameters using only 14 million words of data for Broadcast News. Maximum entropy methods avoid potential data sparseness problems by using only reliable (marginal) counts.

The high accuracy of maximum entropy models also come from properly choosing features for the model. For instance, the part-of-speech tags and syntactic non-terminal labels have the similar form. However, the latter are grammatically more meaningful than the former and comparatively will result in more improvement in speech recognition.

8.1.2 Efficiency

We demonstrate that maximum entropy models with various kinds of constraints are more space efficient than the corresponding interpolated models. We have shown in Chapter 4 that the topic-dependent maximum entropy models are only about 1/8 the size of their interpolated counterparts. The maximum entropy syntactic models also have few parameters than the corresponding interpolated models described in Chelba (2000), because the former use only marginal constraints while the latter need to store both marginal and joint counts.

We also claim that both the training and the use of maximum entropy models are efficient in time using the methods described in this dissertation. Prior to our work, the training of maximum entropy language models of large size was almost intractable. Several efficient training methods have been presented in Chapter 3 to

simplify the intensive computation in estimating model parameters. Models with only nested features, such as N-gram models, have hierarchical structures and can be trained as fast as training back-off models. Other models can also be converted to equivalent hierarchical models by hierarchization. Topic-dependent models in particular can be trained by divide-and-conquer in addition to hierarchical training; their parameters are estimated in topics and then merged. A nominal speed up of hundreds to thousands fold and a real speed-up of ten to one hundred fold have been achieved compared to the baseline unigram-caching method.

The computational load in calculating probabilities using maximum entropy models is also reduced significantly.

It is worth mentioning that the efficient training methods described about can be applied to models other than language models. We show, using the syntactic parser presented in Charniak (2000), how to train other statistical models hierarchically.

Training Charniak (2000) Parser Efficiently

The top-down parser described in Charniak (2000) explored each constituent c in a parse π by first predicting the pre-terminal tag¹ t of c , then the lexical head word h and finally the non-terminal label l . According to Equation 1 in Charniak (2000), the probability of a parse π is assigned as

$$p(\pi) = \prod_{c \in \pi} p(t|l, H)p(h|t, l, H)p(e|l, t, h, H)$$

where

$$H = l_p, t_p, l_b, l_g, h_p$$

is the history. Subscripts p , b and g in the equation above represent, respectively, the parent, the sibling and the grandparent of c . All three probabilities above can be estimated by ME models. It is apparent that the computationally intensive ME models are $p(h|t, l, H)$ and $p(e|h, t, l, H)$. We illustrate how to train these models efficiently by the example of $p(h|t, l, H)$. The training of the remaining models is similar.

¹Similar to the part-of-speech tag.

If we use the features suggested in Equation (7) in the paper, we can rewrite $p(h|t, l, H)$ as

$$p(h|t, l, H) = \frac{1}{z(t, l, H)} \alpha_h^{g(h)} \cdot \alpha_{t,h}^{g(t,h)} \cdot \alpha_{l,t,h}^{g(l,t,h)} \cdot \alpha_{l_p,l,t,h}^{g(l_p,l,t,h)} \cdot \alpha_{t_p,l_p,l,t,h}^{g(t_p,l_p,l,t,h)} \quad (8.1)$$

$$\cdot \alpha_{l_b,l_p,l,t,h}^{g(l_b,l_p,l,t,h)} \cdot \alpha_{l_g,l_p,l,t,h}^{g(l_g,l_p,l,t,h)} \cdot \alpha_{h_p,l_p,l,t,h}^{g(h_p,l_p,l,t,h)}.$$

The computational complexity of training this model using unigram-caching is considerably high (See Table 8.1). Therefore, Charniak (2000) used non-normalized models² instead. However, this approximation may cause degradation in performance. A better implementation is to train real ME models using hierarchical training methods. If the last three 5-gram features $g(l_b, l_p, l, t, h)$, $g(l_g, l_p, l, t, h)$ and $g(h_p, l_p, l, t, h)$ in Equation (8.1) are compounded into a super-feature $f(l_b, l_g, h_p, l_p, l, t, h)$, all features in (8.1) are nested, and thus the model can be trained hierarchically. Table 8.1 shows the computational load, in number of terms as described in Section 3.9.1, of unigram-caching and generalized hierarchical training, and the nominal speed-up of the latter. It is apparent that training such an ME parser will become practical (after a speed-up

Model	Unigram-caching	Hierarchical Training	Speed-up
$p(e l, t, h, H)$	$1.7 \cdot 10^9$	$2.9 \cdot 10^6$	580
$p(h l, t, H)$	$2.3 \cdot 10^9$	$2.9 \cdot 10^6$	790

Table 8.1: Nominal speed-up in training Charniak parser.

of hundreds of fold) when the generalized hierarchical training is applied, even though its computational load is intractable using unigram-caching.

8.1.3 Flexibility

We claim that the maximum entropy method has the high flexibility of incorporating constraints from different information sources in a unified framework. In this dissertation, we started from a trigram model and then augmented it by adding semantically and syntactically meaningful constraints. Finally, we combined both topic

²Assuming $Z = 1$.

constraints and syntactic constraints in one model and obtained almost additive improvement from both kinds of constraints. All these maximum entropy models have the similar exponential form and differ only in feature sets. Moreover, maximum entropy methods provide the flexibility of further expanding these models by adding new meaningful features.

8.1.4 Summary of Experimental Results

We summarize the improvement achieved in this dissertation work in Table 8.2. The table shows the relative perplexity reduction and the absolute word error rate re-

Language Model	Switchboard				Broadcast News			
	Speed-up		Reduction		Speed-up		Reduction	
	Nom.	Real	Ppl	WER	Nom.	Real	Ppl	WER
Trigram	170	30	79	38.5%	560	85	174	34.6%
Topic	400	330	-7%	-0.7%	1300	-	-12%	-0.7%
Syntactic	90	17	-6.3%	-1.0%	140	-	-7.0%	-0.8%
Composite	-	-	-13%	-1.5%	-	-	-18%	-1.2%

Table 8.2: Summary of experimental results (SWBD and 14M BN).

duction of maximum entropy models compared with the results of the baseline trigram models. The nominal speed-up and the real running time speed-up of the efficient training method based on the unigram-caching are also shown for each model. The speed-up of some models is not available since their baseline training is intractable.

Overall, the perplexity of the maximum model with topic and syntactic constraints reduces by 13% - 18% compared to the back-off trigram models in Switchboard and Broadcast News. The word error rate reduces by 1.2% - 1.5% (absolute). The gain from the topic information and the syntactic structure is almost additive, showing that these two kinds of non-local dependencies are complementary in language modeling.

8.2 Future Work

The work in this dissertation can be extended in several directions.

First, topic-dependent modeling techniques can be extended to building multilingual language models in machine translation. In a statistical machine translator, *e.g.* from English to Chinese, the most probable Chinese sentence \hat{C} given the English sentence E is sought to maximize the posterior probability $p(C|E)$, *i.e.*,

$$\hat{C} = \arg \max_C p(C|E) = \arg \max_C p(E|C) \cdot p(C).$$

The probability $p(C)$ is computed by the Chinese language model. As we have shown in this dissertation, a precise estimate for $p(C)$ should depend on the topic of C . However, this topic information is not available until C is known. Of course, we can use the first-pass translation result of C to detect the topic and then use the topic-dependent model to rescore hypotheses. However, a better way is to detect the topic directly from the English sentence E , since the English sentence and its Chinese translation must have the same topic. The topic-dependent model can thus be used in the first-pass decoding from English to Chinese.

Second, the syntactic parser used in syntactic language modeling can be trained by maximum entropy methods. In Section 5.2, the probability $p(w_i|w_1, \dots, w_{i-1})$ is computed as the product of $p(w_i|W_1^{i-1}, T_{ij})$ and $\rho(W_1^{i-1}, T_{ij})$. The former is already computed by the maximum entropy model, but the latter is still estimated by using three interpolated models: the predictor $p(w_i|h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1})$, the part-of-speech tagger $p(pos_i|w_i, nt_{i-2}, nt_{i-1})$ and the parser $p(parse_i|h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1})$, where $parse_i$ are the operations of constructing the parse tree, such as *joining the left tree to construct a noun phrase* or *joining the right tree to construct a verb phrase*. Details of estimating $\rho(W_1^{i-1}, T_{ij})$ are described in Chapter 2 of Chelba (2000). It is apparent that all three probability models used in the parser can be estimated by the maximum entropy method. The algorithm of estimating the parser parameters using maximum entropy methods is outlined below.

Initial step: Parse all sentences in the training data using the current parser described in Chelba (2000), or other state-of-the-art parsers.

Iteration step 1: Collect counts such as $\#[h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1}, w_i]$, $\#[w_i, nt_{i-2}, nt_{i-1}, pos_i]$ and $\#[h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1}, parse_i]$ as described

in Chapter 2 of Chelba (2000).

Iteration step 2: Train probability models $p(w_i|h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1})$, $p(pos_i|w_i, nt_{i-2}, nt_{i-1})$ and $p(parse_i|h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1})$ using the maximum entropy method. (Tuples and their counts obtained from the step above are training samples.)

Iteration step 3: Check the perplexity on held-out data. Stop if the perplexity has converged. Otherwise parse the training data using the new model and go to **Iteration step 1**.

Two challenges must be overcome in implementing the algorithm above. First, iteration steps 2 and 3 are computationally intensive. Furthermore, there is another iteration loop inside step 2, which is for training maximum entropy models. To avoid confusion, we regard the iteration loop described in the algorithm above as the *outer* loop and that inside step 2 as the *inner* loop. Second, the constraints in maximum entropy models may change from iteration to iteration. For example, the tuple $\langle h_{i-2}, h_{i-1}, nt_{i-2}, nt_{i-1}, w_i \rangle$ with a high initial count may obtain quite a low count after some iterations and thus should be removed from the feature set. Of course, some new constraints are also added after each iteration. This will result in different maximum entropy models in different outer iterations. If maximum entropy models are always trained starting from uniform models, the training time, compared to that of regular syntactic models, will increase proportionally to the number of the outer iterations.

We can reduce the training time per iteration by using the generalized hierarchical training described in this dissertation. Chelba (2000) also noticed that the feature sets in two consecutive outer iterations are largely overlapped. We can take advantage of this fact and set the initial model parameters by the values from the previous outer iteration for common features. For new features, we still assign $\alpha = 1$ as initial parameters. The maximum entropy models will converge faster after the first outer iteration than in the first outer iteration.

Finally, maximum entropy modeling techniques can be used in many other applications, such as question answering and automatic summarization. In these ap-

plications, the predicting information comes from various sources, for example, the positions of particular key words, the similarity between the current sentence and the question, and the length of the sentence etc. These applications will benefit from sound models combining different kinds of constraints using maximum entropy methods.