A Maximum Entropy Method for Language Modelling

Oscar Chan, Roberto Togneri
School of Electrical, Electronic and Computer Engineering
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009
Email: oscar@ee.uwa.edu.au, roberto@ee.uwa.edu.au

Abstract—The language models used for automatic speech recognition (ASR) are often based on very simple Markov models. This paper presents an overview of a more powerful modelling technique, Maximum Entropy (ME), and its application in language modelling. Preliminary results indicate that ME models are viable for this task, and perform slightly better than the traditional models.

Index Terms—language modelling, maximum entropy

I. INTRODUCTION

A statistical language model defines a probability distribution over word sequences. The goal of language modelling is to develop models which capture relevant aspects of natural language. They have a variety of uses in natural language processing (NLP) tasks, including spelling and grammar checking, optical character recognition, continuous speech recognition, and language understanding. In the case of continuous speech recognition, the task of the recogniser is to assign a sequence of labels, or words, to an acoustic input. Because speech recognisers model units at the phoneme level, the language model is required for providing higher level linguistic knowledge. More specifically, the purpose of the language model is to assign a priori probabilities to sequences of words. Language models can be used in two stages of the recognition process: they can be built directly in the recogniser component to constrain search hypotheses, or they can be used to rank a set of outputs generated by the recogniser. A typical multi-pass recogniser will employ a simple language model integrated into the recognition process to generate a list of hypotheses, which are subsequently ranked by a more complex model. This paper deals with the latter approach, using ME language models to rescore the output of a speech recogniser.

II. PERPLEXITY AS AN EVALUATION METRIC

In order to evaluate and compare language models, an objective measure of performance is required. In the case of language models used for speech recognition, the Word Error Rate (WER) would be the most appropriate measure. However, it is often beneficial to be able to gauge the quality of a language model without the need for a recogniser, and this is necessary if the intended application of the language model is not speech recognition. An evaluation metric independent of the recogniser can be obtained from the field of information theory. If a speaker is considered to be an information source generating a sequence of discrete symbols (words), $w_1, \ldots, w_n$, according to some model $p_m$, of a language, the cross entropy of the unknown true distribution, $p$, with respect to $p_m$ is given by:

$$H = - \lim_{x \to \infty} \frac{1}{n} \sum w p(w_1, \ldots, w_n) \log p_m(w_1, \ldots, w_n)$$

(1)

Assuming ergodicity, and for large $n$, this can be approximated by:

$$H \approx - \frac{1}{n} \log p_m(w_1, \ldots, w_n)$$

(2)

This gives an indicator of how accurately $p_m$ models $p$. Cross entropy forms the basis for the metric of perplexity, which is defined as:

$$PP = 2^H$$

(3)

and can be interpreted as the average number of equiprobable choices the model encounters when presented with a sample of the language. A lower perplexity generally represents a superior language model, and is correlated with lower recognition errors [1].

III. N-GRAM LANGUAGE MODELS

One of the most commonly used types of LMs is the $n$-gram. These are $n - 1$ order Markov models, which determine word probabilities based on a limited observed history. A trigram (or 3-gram) model, for example, calculates the probability of a word as:

$$Pr(w_i|h) \triangleq Pr(w_i|w_1, \ldots, w_{i-1}) \approx Pr(w_i|w_{i-2}, w_{i-1})$$

(4)

These probabilities are determined by counting occurrences of word sequences in a training text. There are two reasons for the simplifying assumption. The first is that the correlation between words decreases as the distance between them becomes larger. Using long distance dependencies may not be of benefit, and can in fact hinder performance if the model makes inaccurate inferences from such statistics. The other reason is that, in practice, there will be insufficient training data to provide accurate counts of all possible word sequences. This problem is referred to as data sparsity, and becomes even more prominent as the value of $n$ increases. Many smoothing techniques have been developed to deal with this problem, such as backing off to lower order models [2], discounting schemes [3], [4], or clustering the vocabulary into classes [5].
IV. MAXIMUM ENTROPY MODELS

The simplicity of n-gram models has led to their widespread use, and they have been largely proven to be effective. However, these models are inherently restricted by their Markov assumption. Usually, the value of n is no higher than three. Clearly, assuming the occurrence of a word is dependent solely on the previous two words is incorrect, however data sparsity issues necessarily restrict the model to such short histories. Therefore, any information contained outside this local context is lost by n-grams.

Rosenfeld [6] proposed the use of ME models for language modelling. In the ME framework, knowledge is formulated as feature functions, or features. Binary valued features are most commonly used, and are simply indicator functions of the form:

\[ f(w, h) = \begin{cases} 
1 & : (w, h) \in S \\
0 & : \text{otherwise}
\end{cases} \]

where \( S \) is an arbitrary subset of \( W \times H \). This gives ME models great flexibility, as any information consistent with this form can subsequently be used. Features are incorporated into a model, \( p \), by constraining the model’s expectation of these features to equal to a target value, \( K \), using the constraint equation \( E_p[f] = K \). Typically, the target value is derived from a training sample such that \( K_i \) is the expectation of the \( i \)th feature \( f_i \) with respect to the empirical distribution \( \tilde{p} \). With this assumption, the constraint equation then becomes:

\[ E_p[f_i] = \sum_{w, h} \tilde{p}(w, h) f_i(w, h) \]

(5)

The appeal of ME models is their ability to model the given information without making unjustified assumptions. That is, from the set of all models consistent with the provided constraints, the ME model is the one which is closest, in terms of the Kullback-Leibler divergence [7], to the uniform distribution. Often, an initial distribution, \( p_0 \), is also given, and the distance from this distribution is minimised rather than from the uniform distribution. The resulting model, the Minimum Discrimination Information (MDI) solution, will not be the ME solution unless \( p_0 \) is uniform, although it is conventional to refer to it as such. It has been shown [8] that the MDI solution, \( p_s(s) \), has an exponential form. Additionally, if the \( K_i \)’s are the empirical expectations, the MDI solution is also the model in the exponential family which maximises the likelihood of the training sample [9].

A. Conditional Models

Early work in the area of ME language modelling used conditional exponential models of the form:

\[ p_s(w|h) = \frac{1}{Z(h)} p_0(w|h) \exp \left( \sum_i \lambda_i f_i(w, h) \right) \]

(6)

where the \( \lambda_i \)’s are Lagrange multipliers, and \( Z(h) \) is a normalising term.

While this work showed positive results, with a ME model employing triggers achieving as much as a 39% perplexity reduction [6], they also highlighted some disadvantages with ME models. The most significant of these is that the computational requirements of training are quite severe, since \( Z(h) \) needs to be calculated for all \((w, h)\). Additionally, some characteristics of language, such as semantics, grammar and prosody, are difficult to model in a conditional framework. Conditional maximum entropy models therefore suffer from similar limitations to standard n-gram models in this regard.

B. Joint Models

Joint exponential models of the form:

\[ p(s) = \frac{1}{Z} p_0(s) \exp \left( \sum_i \lambda_i f_i(s) \right) \]

(7)

have also been used for language modelling. While these were introduced as Whole-Sentence models [10], the unit being modelled, \( s \), need not be a sentence. Modelling paragraphs or even whole documents is valid under this framework.

Joint ME models address the two major issues arising from conditional ME models, namely their computational complexity and inability to effectively model large span phenomena, however they introduce a different problem. Although \( Z \) is now a true constant, it cannot be directly computed, since that would require a summation over all possible sentences. Calculating the feature expectations \( E_p[f_i] \), requires a similar summation, so these values need to be approximated via sampling. One such method is importance sampling, where the feature expectation is given by:

\[ E_p[f] = \frac{\sum_j \tilde{p}(s_j) f_i(s_j)}{\sum_j \tilde{p}(s_j) q(s_j)} \]

(8)

In this approach, a value is sampled \( \tilde{p}(s) \) times to compensate for sampling from an instrumental distribution \( q(s) \) rather than the true distribution \( p(s) \).

C. Parameter Estimation

Given an empirical distribution, \( \tilde{p} \) and a set of constraints, the goal is to find the model, \( p \), which minimises the Kullback-Leibler divergence between \( p \) and \( \tilde{p} \):

\[ \arg \min_{p \in \mathcal{P}} D(\tilde{p} \parallel p) = \arg \min_s \sum_s \tilde{p}(s) \log \frac{\tilde{p}(s)}{p(s)} = \arg \max_{p \in \mathcal{P}} \sum_s \tilde{p}(s) \log p(s) \]

(9)

where \( \mathcal{P} \) is the set of all models having the form (7). (9) is also the log-likelihood, \( L_{\tilde{p}}(p) \), for the exponential model, and is a concave function over the parameters \( \{ \lambda_i \} \). This has no closed form solution, so it is solved either iteratively or with standard constrained optimisation techniques. It is worthwhile noting that the derivative of the log-likelihood function is:

\[ \frac{\partial}{\partial \lambda_i} L_{\tilde{p}}(p) = K_i - E_p[f_i] \]

(10)

\( E_p[f] \) is required for both the iterative scaling and constrained optimisation methods, and \( K \) is a constant. Therefore, there is little additional computational cost associated with using a
D. Smoothing

ME models, like other maximum likelihood models, are subject to over-training. As the number of features used in the model increase, the model becomes more tightly constrained to the training sample and loses its ability to generalise. In order to alleviate this, some of the probability mass can be redistributed in a process called smoothing. One technique for smoothing ME models is to apply a Gaussian prior over the model [12], and perform maximum a posteriori optimisations performed. This effectively changes the objective function to:

$$F(p) = \sum_s \tilde{p}(s) \log p(s) - \frac{1}{2\sigma^2} \sum_i \lambda_i^2$$

where the second term penalises models which diverge from the initial model, and $\frac{1}{2\sigma^2}$ acts as a weight used to trade off between well-fitted and smooth models.

V. EXPERIMENTAL WORK

This work makes use of the Text Modeller toolkit [13], which, to date, is the only freely available toolkit designed to fit joint maximum entropy models. This toolkit uses gradient based methods for optimisation, and currently has implementations for the Conjugate Gradient (CG), Broyden-Fletcher-Goldfarb-Shanno (BFGS) and Powell algorithms. No functionality for smoothing was provided by the toolkit, so the Gaussian prior technique described in Section IV-D was implemented to perform this.

The experiments were carried out on a subset of the Boston University Radio Speech Corpus [14], which consists of professionally read radio speech data. This subset comprised of 422 utterances containing 32289 words in 1567 sentences. 85% of the utterances were selected at random to form a training set, 5% for development testing and the remaining 10% made up the test set. A trigram model using Good-Turing discounting was trained on this data, and serves as the baseline.

The models were constructed using n-gram features of the form:

$$f_\alpha(s) = \begin{cases} 1 & \text{if } n\text{-gram } \alpha \text{ occurs } m \text{ times in } s \\ 0 & \text{otherwise} \end{cases}$$

for all $m$, and the baseline trigram was used as the initial distribution. These are somewhat different from standard n-gram features which are used in conditional models. Consider the bigram stock market. In a conditional model, a feature would be derived based on $p(w_i = \text{market}|w_{i-1} = \text{stock})$. This is a measure of the relative frequency with which the word market follows the word stock. But in a joint model, sentences, rather than words, are treated as atomic units. In order to cater for this framework, the information is formulated as $p(\text{stock market} \text{ occurs in sentence } s)$. However, while this provides knowledge on the co-occurrence of the two words, it does not fully capture the frequency in which the n-gram may occur. Suppose stock market occurs twice in a particular sentence. This feature would fail to distinguish between such a sentence and one in which stock market occurred only once. Subsequently, the model underestimates the true frequency of the word pair, the effects of which are most apparent with low order n-grams. Therefore, it is also necessary to model how often an n-gram occurs in s, leading to features of the form given by (12).

In the experiments performed here, the CG method was used for model optimisation. Model expectations, $E_p[f]$, were estimated using importance sampling and $10^5$ samples generated from the initial distribution, while feature target values, $K$, were derived from counting their occurrences in the training set. To obtain reliable estimates of $K$, it is desirable to introduce cutoffs such that features occurring in the training data fewer times than the cutoff value are excluded. Table I shows the test set perplexity and relative improvement over the baseline of the ME models for various cutoff values. Models were smoothed with the Gaussian prior method, where the value of $\frac{1}{2\sigma^2}$ was optimised on the development test set. All of these models achieved slightly better results than the baseline trigram, with the greatest perplexity reduction obtained using a cutoff of 3. It is likely that the performance of those models using lower cutoffs was hampered by the inclusion of too many unreliable features.

These language models were also applied to the rescoring of n-best lists generated by a recogniser. The speech data was parameterised into feature vectors consisting of 13 MFCCs with delta and acceleration, and acoustic models used 3-state triphone HMMs with 5 Gaussian mixtures per state. A 100-best list was generated using the recogniser and a trigram language model, which was then rescored using the ME model with the lowest perplexity. This resulted in an increase in word accuracy from 58.33% to 58.85%, a relative improvement of 0.9%. While this is a modest gain at best, it is consistent with expectations since it was achieved using the same information available to the n-gram model. In addition, it is possible that recognition performance may be hampered by the use of n-best lists. Because these lists are generated with the aid of the existing language model, the coverage of the hypotheses is often quite limited. From the partial n-best list shown in Table II, it is evident that the same errors are propagated throughout the entire list. It would be preferable to have more alternatives in order the increase the changes of the correct

<table>
<thead>
<tr>
<th>Cutoff</th>
<th>Perplexity</th>
<th>% Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>trigram</td>
<td>44.23</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>43.02</td>
<td>2.70</td>
</tr>
<tr>
<td>2</td>
<td>43.11</td>
<td>2.48</td>
</tr>
<tr>
<td>3</td>
<td>42.87</td>
<td>3.06</td>
</tr>
<tr>
<td>4</td>
<td>42.94</td>
<td>2.88</td>
</tr>
<tr>
<td>5</td>
<td>43.10</td>
<td>2.51</td>
</tr>
<tr>
<td>6</td>
<td>43.33</td>
<td>1.99</td>
</tr>
</tbody>
</table>

TABLE I

Test set perplexity of ME models smoothed with a Gaussian prior. These results were obtained using the optimal value of $\frac{1}{2\sigma^2}$ for each model.
transcription being present. This can be achieved by either increasing the size of the list or by using a word hypothesis graph (WHG), which can compactly represent a large number of transcriptions.

VI. Conclusions and Future Work

Maximum entropy language models were trained on a portion of the Boston University Radio Speech Corpus using n-gram type features, and were shown to provide a small improvement over a standard trigram model in both perplexity and word recognition results. On top of demonstrating the validity of this approach, the flexibility of the ME framework for incorporating knowledge opens up a number of avenues for further exploration.

The direction this work will follow is to investigate the use of prosodic features, such as intonation and stress. The suprasegmental nature of these features makes them difficult to use in traditional models, however the joint ME approach is well suited to the task. The radio speech data being used is one of the few corpora designed for research into prosody, and has been annotated according the Tones and Break Indices (ToBI) scheme. Language models making use of these prosodic annotations are currently being developed.

References