LEARNING ON THE JOB: THE APPLICATION OF MACHINE LEARNING WITHIN THE SPEECH DECODER

M A Huckvale Department of Phonetics and Linguistics, University College London, Gower Street, London WC1E 6BT, U.K.
G J A Hunter Department of Phonetics and Linguistics, University College London, Gower Street, London WC1E 6BT, U.K.

1. INTRODUCTION

The current approach to the training of large vocabulary continuous speech recognition (LVCSR) systems involves the use of large corpora of text and labelled audio recordings [1]. These resources are analysed and statistics extracted so that the recogniser can determine the likelihood that the observed signal would have been generated from each sentence it proposes. The analysis of corpora is performed *off-line*, using statistical language models of the text (often trigram models of words), and acoustic models of the signal (often hidden Markov models of phonemes).

To this off-line processing, recent years have seen a growth in the use of methods of adaptation in which the general statistical models are tuned to the specific characteristics of a given speaker, a given acoustic environment or a given topic. These adaptation processes modify the stored characteristics of the language model and the acoustic model to improve the probability that the correct interpretation would have given rise to the observed signal.

No one would argue that these components provide a perfect model of the true statistical distribution of words and sounds. Weaknesses in typical acoustic models include:

- crude modelling of the interdependencies between the acoustic forms of different phones
- no model of systematic pronunciation variation across different contexts or speakers
- little exploitation of durational or pitch cues
- no exploitation of knowledge of style, emotion, or physiological state of the speaker

Weaknesses in typical language models include

- restriction to short-distance dependencies within sentence (trigram models)
- little exploitation of topic or meaning or grammaticality
- poor predictive power for novel or rare events
- limited vocabulary and inability to deal with novel words

These weaknesses are, of course, opportunities for research; and much effort has been spent at looking at these.

There are also many weaknesses that can be seen within the decoder: how these statistical components are exploited in recognition. Weaknesses here include:

- arbitrary balancing of probabilities between the acoustic and language models
- ignorance of interactions between the acoustic model and the language model
- assumptions that words don't overlap in time
- inability to deal with disfluencies and restarts

These are less common areas for research.

Thus we arrive at the present situation in which work is required on many fronts, but each aspect may in itself only provide a modest improvement in performance. It is as if there are many small weaknesses rather than one significant problem. A serious consequence of this situation in speech recognition research is that workers on one small aspect do not know what effect their 'improvements' will have in combination with the work of others. We have been working in the area of morphology for speech recognition [2] but we do not know whether the improvements we've seen will show up in combination with more sophisticated language models or with state-of-the-art acoustic models.

In this paper we are looking towards a 'third way'. Rather than try to build better statistical models, or try to find ways of adapting them to the context, we seek to apply general machine learning principles within the decoder. Thus the decoder will monitor and modify its own behaviour by 'learning on the job'. This work is very much in the exploratory stage. We do not yet know whether the approach will make any significant impact. We do not yet know how it relates to other work in improving language models and acoustic models. We do not even know the best way to make it work.

Our learning decoder is able to relate the correct transcription of an utterance to the complete list of hypotheses that it generated during its attempt at decoding the signal. By looking at the correct and incorrect hypotheses over large numbers of training utterances, it tries to find features of these hypotheses that correlate with their correctness (or with their incorrectness). The aim is not to replace the language model or acoustic model, nor to act as an alternative to adaptation. Instead the machine learning should identify and compensate for common errors made during decoding. Those features that correlate with *correct* can be used to improve the score of probably correct hypotheses, and those features that correlate with *incorrect* can be used to worsen the score of probably incorrect hypotheses. We can use data held-out from training to evaluate the effect of the learning component.

In this paper, we describe how we have implemented and tested this application of machine learning within the decoder of a large vocabulary continuous speech recognition system. In section 2 we describe the mathematical framework we have adopted, while in section 3 we describe a small experiment proposed only as a proof-of-

concept. In section 4 we reflect on the promises offered by the technique and make suggestions for further investigations.

2. Supervised machine learning in the decoder

The aim of the machine learning system is to

- uncover characteristic features of sentence fragment hypotheses which correlate with the correctness of the hypothesis, and
- deliver a probability to the decoder that a sentence fragment is correct given the features that it exhibits.

We describe each of these in turn.

2.1 Selection of features

What features of a sentence fragment hypothesis would assist in determining its probability of being correct? Any features we choose should be complementary to the information provided by the acoustic model and the language model.

In terms of acoustic information, these features might be based on:

- articulation rate, tempo variations, segment durations
- fundamental frequency, voice quality
- articulatory quality
- level of background noise
- detection of speaker, accent, style, emotion or physiological state

In terms of linguistic information, these features might be based on:

- collocational information about words across whole sentences
- measures of grammaticality
- measures of semantic relationships between words

Although many of these aspects of language are likely to influence how a listener decodes an utterance, it is just very complicated to see how they can all be modelled independently and all incorporated in the decoding.

Worse, in many cases we don't know the relative importance of the different features, not how they interact. It is very hard to judge the *utility* of the information provided by a feature. We may run into the problem highlighted by Rosenfeld [3] that we will never have enough data to model rare events - because they are rare.

Thus the first task of our machine learning component will be to decide which of the very many possible features will be of use in practice. Since it is relatively easy to suggest features, but hard to know how useful they are, we leave this task up to the learning system. We simply suggest a very large number of *possible* features and let the system decide which ones to take note of. A useful measure of utility is *mutual information* [10]. For some binary feature f_i and some correctness indicator y, we can calculate the mutual information between f_i and y as:

$$MI(f_i, y) = \sum_{f=0,1} \sum_{g=0,1} p(f_i = f, y = g) \log\left(\frac{p(y = g \mid f_i = f)}{p(y = g)}\right) \quad (1)$$

We can choose features with high mutual information shown between the feature and the known correctness of a hypothesis. Features with high mutual information may be useful in predicting correctness or incorrectness and are saved for evaluation in combination.

2.2 Probability modelling of features

Given some signal S and some hypothesis W, we normally calculate the probability that a hypothesis is an interpretation of a signal using Bayes' theorem

$$p(W|S) = p(S|W).p(W)/p(S) \quad (2)$$

Where p(S|W) is the probability that the hypothesis *generated* the signal calculated by the acoustic model, and p(W) is the probability of the hypothesis itself, as calculated by the language model. The decoder seeks to find the single hypothesis that maximises p(W|S).

To incorporate knowledge about some additional features of a hypothesis F(W) not covered by the language model, we can extend the language model to incorporate the prediction of some property y indicating the correctness of the hypothesis:

$$p'(W,y) = p(W).p(y|F(W)) \quad (3)$$

assuming that the language model and the predictions from the features are independent. The probability that a hypothesis is correct given the features of the hypothesis can be expressed in terms of an *exponential model* of the form

$$p(y = correct \mid F(W)) = \frac{\exp[\sum_i l_i f_i]}{1 + \exp[\sum_i l_i f_i]} \quad (4)$$

where f_i is 1 if the feature *i* is present in the list F(W). The $\{?_i\}$ are constants found from training data. A particular benefit of this model is that the $\{?_i\}$ can be estimated using the principle of *maximum entropy*. Here the least constraining assumptions are drawn from the training data. The $\{?_i\}$ are found by maximising the entropy function

$$\Psi(l) = -\sum_x p(x) \log(1 + \exp[\sum_i l_i f_i]) + \sum_i l_i p(f_i) \quad (5)$$

where x refers to each different training pattern, p(x) is the probability that the pattern occurs in the training data, and $p(f_i)$ is the probability that feature *i* is seen. We choose to find the maximum of this function using a method of functional optimisation [4]. Other approaches can be found in [5].

3. Experiment

3.1 Materials

Text material for training and testing was selected from the British National Corpus [6]. 80M word of text was reserved for training, and the rest for testing. The corpus was pre-processed to remove all punctuation except for sentence markers, and to convert all numeric items and abbreviations to whole words. A vocabulary of 65,000 words was generated from the most common words in the training portion.

For this experiment we used 1000 spoken sentences taken from the testing portion of the BNC, 100 each from 5 male and 5 female speakers of British English. These were converted to word lattices using the Abbot system [7] with a 65,000-word pronunciation dictionary adapted from BEEP [8] and supplemented with pronunciations from a letter-to-sound system. Abbot was run with parameters provided by Steve Renals to increase the maximum number of hypotheses considered per node to 100.

A language model was constructed for the 65,000-word lexicon using the 80Mword training portion of the BNC. This was performed using the CMU-Cambridge toolkit [9] using Good-Turing discounting.

Decoding of the word lattices using the language model was performed by the UCL decoder, which is able to report node-by-node the currently considered sentence fragment hypotheses for each time step in the word lattice. These hypotheses always extend from the start of the sentence to a word that ends at the current node. They are marked with an overall log probability found during decoding from the acoustic model and the language model.

3.2 Preparation

The hypotheses produced during the decoding of the 1000 sentences were marked for correctness using the known transcription. For training and testing the maximum entropy feature model, we used only those hypotheses that originated from nodes where a correct answer was present within the top 100 hypotheses. This gave us a total of 430,000 hypotheses, of which 26,000 were correct. On average each hypothesis contained 5.65 words.

10% of the data (10 sentences) was reserved from each speaker for testing; the rest was input to the training procedure.

3.3 Feature generation

For this experiment we based our features simply on the collocational properties of word classes within the hypotheses. To do this we designed a set of 50 word classes using word frequency information generated from the training corpus. The word classes were chosen to have approximately similar frequencies in the training corpus. This was achieved by studying the relative frequency of the 50 most common words and the frequency of the 50 most common BNC word tags. We found that a combination of the 25 most common words, 24 most common tags and 1 miscellaneous class gave a

suitable mapping from each word to one of 50 classes. The list of classes is shown in table 1

Table 1 - Word Classes

Class	Word	Class	Tag	Description
1	THE	26	NN1	Singular Noun
2	<S>	27	MISC	Miscellaneous
3	OF	28	AJ0	General Adjective
4	AND	29	NN2	Plural Noun
5	TO	30	AV0	General Adverb
6	A	31	NP0	Proper Noun
7	IN	32	CRD	Cardinal Number
8	IS	33	PNP	Personal Pronoun
9	THAT	34	DT0	General Determiner
10	WAS	35	VVI	Verb Infinitive
11	FOR	36	PRP	Preposition
12	IT	37	VVN	Past Participle Verb
13	ON	38	VM0	Modal Aux. Verb
14	WITH	39	VVD	Past Tense of Verb
15	AS	40	VVG	Verb (-ing form)
16	HE	41	DPS	Possessive Determiner
17	BE	42	NN0	Noun (not number specific)
18	BY	43	CJS	Subordinating Conjunction
19	AT	44	DTQ	wh- determiner
20	I	45	VVZ	Present form (-s) of verb
21	ONE	46	AT0	"Article" determiner (a, the,an)
22	HIS	47	AJ0-NN1	Word can be noun or adjective
23	NOT	48	VBB	Present tense of verb "to be"
24	BUT	49	AVP	Adverb particle (up, off, ...)
25	FROM	50	VHD	Past tense of verb "to have"

Using these word classes, collocational features were proposed as follows: feature F(m,n) is 1 if and only if word-class m occurs in the hypothesis before word class n. Thus each hypothesis is converted to a (sparse) vector of 2500 bits.

3.4 Feature Winnowing

To determine which of the 2500 features had some potential for predicting the correctness of the hypothesis, a first ‘winnowing’ stage was implemented using a mutual information criterion as described in section 2.1.

The winnowing procedure looked only at those hypotheses that were either correct or which had a score better than the correct hypothesis on the node. The mutual information was calculated between each feature f_i and the correctness indicator y. The 50 features showing the greatest values were retained for input to the maximum entropy modelling.

3.5 Maximum entropy modelling

From the list of 50 features showing the greatest mutual information, maximum entropy models are made using a greedy algorithm (following [5]) that considers first the best model with one feature, then the best second feature that can be added to the first, the best third feature that can be added to the first two, and so on.

The maximum entropy modelling halts when the additional benefit of adding another feature falls below some threshold. A typical example of a model of 10 features is shown in table 2.

Table 2 - Example Maximum Entropy Model

No.	Feature	Lambda	Description
1	1<26	-1.55945	“the” before singular noun
2	26<26	-1.50821	singular noun before singular noun
3	27<1	-1.42796	miscellaneous before “the”
4	1<28	-1.558	“the” before general adjective
5	30<1	-1.42289	general adverb before “the”
6	30<31	-3.89828	general adverb before proper noun
7	34<1	-1.3658	general determiner before “the”
8	27<8	-1.92252	miscellaneous before “is”
9	1<27	-1.43065	“the” before miscellaneous
10	26<35	-1.56516	singular noun before infinitive

Note that all the lambda values are negative, indicating that these features reduce the likelihood of any hypothesis containing these features being correct. Features that increased the likelihood of a hypothesis being correct were found by the winnowing procedure but they did not find their way into any maximum entropy model.

At first sight these features of incorrect hypotheses do not look particularly odd. However a feature is useful if its frequency of occurrence is different in correct and incorrect hypotheses. Thus the fixation on the use of ‘the’ may simply indicate that the recogniser is hypothesising this word too often.

3.6 Evaluation

To evaluate the feature selection and maximum entropy models, the 10% of data reserved for testing was processed through the word-class mapping and feature extraction stages. The overall score for each hypothesis was then adjusted using equations (3) and (4) for each of the selected features and calculated lambda parameters found from the 90% of data used for training. The procedure was then repeated 10 times for each possible division between test and training.

To evaluate the effectiveness of the new scores for each hypothesis, we calculated the average rank of the correct answer in the list of hypotheses generated for each node. After rescoring, the hypothesis list was resorted and the average rank of the correct answer recalculated. The results are shown in table 3:

Table 3 - Change in Ranking of First Correct Hypothesis

Test Data Set	Mean Correct Ranking (Before)	Mean Correct Ranking (After)	Mean Improvement
hyp0.lst	13.15	9.94	3.21
hyp1.lst	15.63	12.38	3.25
hyp2.lst	14.96	11.99	2.97
hyp3.lst	16.19	11.89	4.30
hyp4.lst	13.96	10.59	3.37
hyp5.lst	15.19	10.12	5.07
hyp6.lst	17.04	11.06	5.98
hyp7.lst	14.51	10.56	3.95
hyp8.lst	14.21	10.40	3.81
hyp9.lst	14.26	10.16	4.10

Overall the mean ranking of the correct answer improved by 4 places, from an average rank of 14.9 to an average rank of 10.9. The results seem consistent across each rotation of data. We have not yet determined how these improvements in ranking affect word recognition score. For this experiment we simply wanted to show that the maximum entropy model made consistent changes to scores in the right direction.

4. Discussion

The experiment described above is only a first attempt at applying the idea of machine learning within the decoder, and serves only as a proof of concept that the idea holds some promise. We made many arbitrary decisions in feature analysis and in modelling and these can almost certainly be improved.

Now that we have the basic framework for experimentation we would like to look at:

1. choosing word classes on the basis of either grammatical functionality, or on the basis of how the word contributes to meaning
2. choosing other features based on the position of the word with respect to words that become before and after it
3. finding the best way to exploit the modified scores in the decoder: whether the modifications should be actually incorporated with scores from the acoustic model and language model, or whether they should be used simply to help rank hypotheses within a node.
4. determining the effect of the machine learning on word accuracy
5. determining the effect of the machine learning on sentences drawn from a different corpus spoken by speakers outside the training set.

One particular problem that might arise with this technique is that the features found in one set of data fail to be useful in another. On the other hand, the technique trawls through a large number of features to find ones that occur commonly and have the greatest effect. We are hopeful that the technique can be extended and refined to incorporate acoustic as well as linguistic features, and that a general learning framework can be established within the decoder to identify further features automatically.

Acknowledgements

Gordon Hunter is supported by a research studentship from the U.K Engineering and Physical Sciences Research Council. Some of the work reported here was conducted under the EPSRC project “Enhanced Language Modelling”, Grant No GR/L81406. Thanks to Alex Fang for help with the BNC corpus and the generation of the language models.

References

[1] S. Young. Large Vocabulary Continuous Speech Recognition: A Review. In *IEEE Signal Processing Magazine*, 13(5), 1996, 45-57.

[2] A.C. Fang, M.A. Huckvale. Enhanced Language Modelling with Phonologically Constrained Morphological Analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 5-9 June 2000, Istanbul, Turkey.

[3] R. Rosenfeld. Incorporating Linguistic Structure into Statistical Language Models. In *Philosophical Transactions of the Royal Society of London A*, 358, 2000, 1311-1324.

[4] J.A. Nelder, R. Mead. A simplex method for function minimization. In *The Computer Journal*, vol.7, 1965, The British Computer Society, 308-313.

[5] A. Berger, S. Della Pietra, V. Della Pietra. A maximum entropy approach to natural language processing. In *Computational Linguistics*, 22, 1996, pp1-36.

[6] L. Burnard. *Users Reference Guide for the British National Corpus.* Oxford University Computing Services, Oxford, 1995.

[7] M. Hochberg, S. Renals, A. Robinson. ABBOT: The CUED Hybrid Connectionist-HMM Large Vocabulary Recognition System. In *Proc. Language Technology Workshop,* Austin Texas, Jan 1995. Morgan Kaufmann.

[8] A. Robinson, BEEP Pronunciation Dictionary. Retrieved from World Wide Web: ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz

[9] P. Clarkson, R. Rosenfeld. Statistical Language Modeling using the CMU-Cambridge Toolkit. In *Proc. Eurospeech 97*, 1997.

[10] R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modelling. In *Computer, Speech and Language*, 10, 1996, 187-288.