Research Article

Unsupervised Adaptation of Statistical Language Models for Speech Recognition

T Niesler[a] D Willett[b]

[a]*Department of Electronic Engineering, University of Stellenbosch, Stellenbosch, South Africa,* trn@dsp.sun.ac.za
[b]*Speech Open Lab, NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan.* willett@cslab.kecl.ntt.co.jp
(Now with TEMIC SDS GmbH, Ulm, Germany).

Abstract

It has been demonstrated repeatedly that the acoustic models of a speaker-independent speech recognition system can benefit substantially from the application of unsupervised adaptation methods as a means of speaker enrollment. Unsupervised adaptation has however not yet been applied to the statistical language model component of the recognition system. We investigate two techniques with which a first-pass recognition transcription is used to adapt the parameters of the n-gram language model that is used in the recognition search. It is found that best results are achieved when both methods are employed in conjunction with each other. The performance of the adaptation methods were determined experimentally by application to the transcription of a set of lecture speeches. Improvements both in terms of language model perplexity as well as recognition word error-rate were achieved.

Keywords: *Statistical language modeling, unsupervised adaptation, speech recognition.*
Computing Review Categories: *I.2.7.*

1 Introduction

The task of a speech recognition system is to automatically produce a text transcript of a passage of human speech. State-of-the-art systems take a probabilistic view of this problem and search for that sequence of words $\mathbf{w}$ that is most likely given $\mathbf{x}$, the acoustic signal obtained from the microphone. In particular, a speech recogniser finds that word sequence $\mathbf{w}$ which maximises the conditional probability $P(\mathbf{w}|\mathbf{x})$. Bayes rule shows that this is equivalent to determining:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \left\{ P(\mathbf{x}\,|\,\mathbf{w}) \cdot P(\mathbf{w}) \right\}$$

From this equation we see that two probabilistic components are employed in the recognition search: an *acoustic model* and a *language model*. The former estimates the likelihood $P(\mathbf{x}|\mathbf{w})$ of the speech signal, given a hypothesis $\mathbf{w}$ of the uttered word sequence. Normally the acoustic model consists of a set of hidden Markov models (HMMs) representing the phonemes of the language in question. In contrast, the language model will estimate the likelihood $P(\mathbf{w})$ of this hypothesised word sequence without regard for the acoustic evidence. Hence this second component reflects the linguistic patterns of the language.

Both the acoustic model and the language model are normally exposed to a great variety of data during training to ensure subsequent good performance under a wide variety of test conditions. For example, the training data will contain speech gathered from many different speakers to afford the system robustness to changes in voice quality, accent and speaking style. Such speaker-independent models can be expected to fare well across a wide variety of test speakers. Better performance can however be achieved for any particular test speaker by using an acoustic model better tuned to the individual in question. Ideally this model would be trained on data obtained exclusively from this speaker, but this is rarely possible since several hours of recorded and transcribed speech are normally required for the training process.

Although the test conditions may be unknown in advance, they normally remain constant for some significant length of time. During this period both the acoustic and the language model can benefit from *adaptation*. For example, the speaker and the topic of discussion may remain unchanged for the length of a conversation. Hence the acoustic models can be adapted to more closely match the characteristics of the speakers voice, and the language model adapted to better model the subject and style of the conversation.

If the correct transcription of the adaptation data is available and used to accomplish this, the adaptation is *supervised*. This style of adaptation has been shown to be successful both when applied to the acoustic model as well as to the language model. If instead the speech recogniser is employed to produce a (generally errorful) transcription of the adaptation data, and this is subsequently used to update the models, the adaptation is *unsupervised*. Supervised

adaptation generally is more effective than unsupervised adaptation. Indeed, due to the presence of recognition errors in the automatically determined transcription there is no guarantee that unsupervised adaptation will not be detrimental to model quality. However the need for a manually obtained correct transcription of the adaptation data may be inconvenient in practice. Hence unsupervised adaptation may remain an attractive option.

This paper deals with the experimental evaluation of unsupervised language model adaptation using two approaches that have been shown to perform well for supervised adaptation. The individual elements of this strategy have mostly already been reported elsewhere in the literature, and this is indicated at the pertinent points. This work considers a novel combination of adaptation methods and includes a complete evaluation in terms of both recognition word-error rates and language model perplexities.

2 Language modelling

Consider a word sequence $\mathbf{w}$ consisting of L words:

$$\mathbf{w}(1,L) \quad = \quad \{w(1),w(2),\ldots,w(L)\}$$

The language model estimates the prior probability $P(\mathbf{w})$ of this sequence. This joint probability may be decomposed into a product of conditionals as follows:

$$P(\mathbf{w}(1,L)) \quad = \quad \prod_{i=1}^{L} P(w(i)|\mathbf{w}(1,i-1))$$

In practice the conditional probability distributions $P(w(i)\,|\,\mathbf{w}(1,i-1)$ must be estimated for each conditioning context $\mathbf{w}(1,i-1)$ from a finite data corpus of example text. The number of different word sequences $\mathbf{w}(0,i-1)$ with i larger than 2 or 3 is impractically large, rendering such estimation infeasible. For this reason these conditioning contexts are normally approximated by the most recent $n-1$ words:

$$P(w(i)\,|\,\mathbf{w}(1,i-1)) \quad \approx \quad P(w(i)\,|\,\mathbf{w}(i-n+1,i-1))$$

Such models are termed n-gram models. In practice, n is normally limited to 2 (bigram) or 3 (trigram). This is the type of model used in most state-of-the-art recognition systems, and that we will consider in this work. Nevertheless, alternatives to this basic n-gram model are available. For example, word classes may be used instead of individual words in the n-gram probability estimates [10]. By grouping words according to some meaningful measure (such as their grammatical function) more robust probability estimates can be obtained from limited data. Alternatively, *cache-* or *trigger-based* models have been proposed as a means of capturing probabilistic dependencies between distant words [8], [11]. Due to their dependence on n-tuples, n-gram language models cannot by themselves capture patterns spanning more than n consecutive words.

3 The Task

The language model adaptation methods will be evaluated by applying them to a recognition system for recorded Japanese lecture speeches. The speech data for this task and their transcriptions were provided by the Japanese national research project on Spontaneous Speech [14]. All the lecture speeches were recorded at conferences on speech, acoustics, linguistics and the Japanese language, and hence the variety of topics under discussion is highly constrained. Language model adaptation for this task can be expected to be difficult due to the high degree of uniformity of the subject matter among the training speeches.

Our speech corpus consisted of a total of 158 speeches, spoken by both male and female speakers and with an approximate average lecture length of 15 minutes. We have set aside 7 speeches as a development test set (dev-test hereafter) and another 7 as an evaluation test set (eval-test hereafter). The development test set will be used to optimise the various parameters of the adaptation process, while the evaluation test set will be kept aside for later independent evaluation. The specification of these sets is given in Table 1.

Data set	#Speeches	Total length	#Words
Development	7	2.0h	22,576
Evaluation	7	3.2h	35,476
Training	144	38h	413,484

Table 1: Development, evaluation and training data.

The acoustic models are constructed using the data in the training set, and are not altered during the subsequent language model adaptation process. A baseline language model is obtained from the same training corpus and is used to obtain a first-pass recognition result. This is used together with the training set to adapt the baseline language model, which is then used in a second recognition pass. This process of language model adaptation can be iterated. The procedure is illustrated in Figure 1.

Note that the transcriptions of the 144 training speeches were the only source of language modeling data used in this work.

4 Language model adaptation

Language model adaptation is most often performed in a supervised manner. This assumes the availability of a well-trained *background* language model together with a relatively small amount of adaptation text from the target domain. The goal is to use this text to adapt the background model so that it will exhibit better performance on further material from the target domain. Good results have been

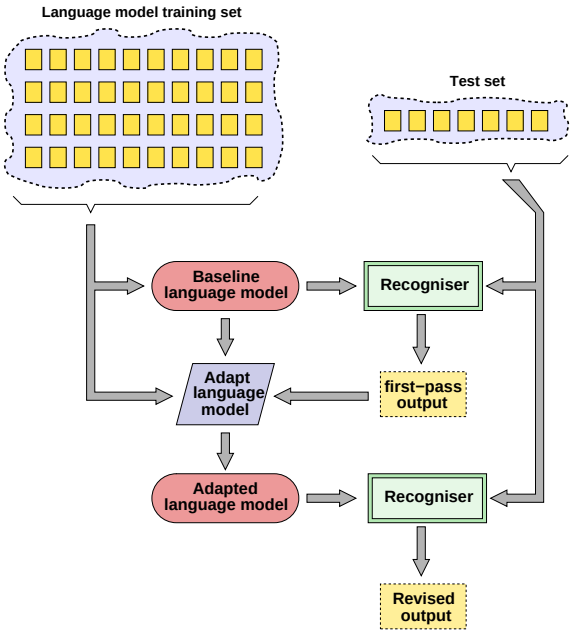

Figure 1: The language model adaptation process.

achieved for this mode of adaptation using for example Bayes and MAP adaptation [2], linear interpolation [7] and minimum discriminative estimation [6].

Supervised adaptation requires the target domain and adaptation text to be available *a-priori*. In situations where this is not possible, unsupervised adaptation becomes attractive. As explained in section 1, this mode of adaptation treats the output of a previous recognition pass as adaptation text data from the target domain. Sections 4.1 and 4.2 describe the techniques used in this work.

4.1 Text selection

In order to obtain a language model more focused on the target domain, we may try to identify a subset of the training material that is in some sense closest to the target domain, and then adapt the background language model using this subset. We will achieve this by selecting from the 144 speeches in the training corpus a set of speeches judged most similar in character to the current recognition hypothesis. In order to measure the similarity between two speeches, we use an information retrieval measure known as *term frequency inverse document frequency* (tf-idf) [12]. Let there be D speeches (documents) in the training set. Denote the words of the training set vocabulary by $\{w_1, w_2, \ldots, w_V\}$, where V is the size of the vocabulary. Define the *term frequency* $tf(d_i, w_j)$ as the number of times the word w_j occurs in document d_i. Finally define the *inverse document frequency* $idf(w_j)$ to be:

$$idf(w_j) \quad = \quad \frac{D}{\text{number of documents containing } w_j}$$

Hence the inverse document frequency is large when the word w_j occurs in few documents. The tf-idf $\mathcal{T}(d_i, w_j)$ of document d_i and word w_j is defined by:

$$\mathcal{T}(d_i, w_j) \quad = \quad tf(d_i, w_j) \cdot \log(idf(w_j))$$

The term frequency is large for frequent words, while the inverse document frequency is large for words occurring in few documents. Hence $\mathcal{T}(d_i, w_j)$ will be large when w_j occurs often in d_i but does not occur in many other documents. Such words may be expected to be good discriminating characteristics of the document d_i.

A measure of similarity $\mathcal{S}(d_i, d_k)$ of two documents d_i and d_k can now be defined:

$$\mathcal{S}(d_i, d_k) \quad = \quad \frac{\sum_{j=1}^{V} (\mathcal{T}(d_i, w_j) \cdot \mathcal{T}(d_k, w_j))}{\sqrt{\left(\sum_{j=1}^{V} \mathcal{T}(d_i, w_j)^2\right) \cdot \left(\sum_{j=1}^{V} \mathcal{T}(d_k, w_j)^2\right)}}$$

If we define the vector $\mathbf{t}$ as follows:

$$\mathbf{t}(d_i) = \{\mathcal{T}(d_i, w_1), \mathcal{T}(d_i, w_2), \ldots, \mathcal{T}(d_i, w_V)\}$$

we see that the similarity is the cosine of the angle between the vectors $\mathbf{t}(d_i)$ and $\mathbf{t}(d_k)$:

$$\mathcal{S}(d_i, d_k) \quad = \quad \frac{\mathbf{t}(d_i) \bullet \mathbf{t}(d_k)}{||\mathbf{t}(d_i)|| \cdot ||\mathbf{t}(d_k)||}$$

where the "$\bullet$" operator in the numerator is the vector dot product. Two documents will therefore be judged similar when corresponding words exhibit a high tf-idf. For such documents the vectors $\mathbf{t}(d_i)$ and $\mathbf{t}(d_k)$ will be directed in a similar direction, and hence the cosine of the angle between them will be close to 1.

Since $\mathcal{T}(d_i, w_j)$ is positive or zero, $\mathcal{S}(d_i, d_k)$ varies between 0 (for unrelated documents) and 1 (for highly related documents).

In order to identify the documents most closely related to the recognition hypothesis d_x, the similarity $\mathcal{S}(d_i, d_x)$ is calculated for each document d_i, $i = 1, 2, \ldots, D$. All documents for which:

$$\mathcal{S}(d_i, d_x) > \gamma \cdot S_{max} \tag{1}$$

are selected as adaptation material for the language model, where

$$S_{max} \quad = \quad \max_i \mathcal{S}(d_i, d_x)$$

and $0 \leq \gamma \leq 1$. When $\gamma < 1$, at least one document will be selected for use as adaptation material.

The tf-idf measure is employed in a similar fashion in [9] to identify relevant documents from a much larger training corpus for supervised language model adaptation. A related text-similarity measure is employed in [5] to optimise the training set of a language model for a particular target domain.

Linear interpolation

Once a subset of the training set has been selected by means of the tf-idf measure, this data must be used to adapt the background language model. This was achieved by building an *n*-gram language model from the adaptation data, and then forming a linear interpolation.

$$P_a(w|h) \quad = \quad \lambda(h) \cdot P_b(w|h) + (1-\lambda(h))\hat{P}_a(w|h)$$

In the above equation w indicates the word for which the probability is sought, and h the context upon which the language model will base its estimate of the probability. Then $P_b(w|h)$ represents the background model, $\hat{P}_a(w|h)$ the language model obtained from the adaptation data, and $P_a(w|h)$ the adapted language model. Each unique context h has an associated interpolation weight $\lambda(h)$ since some contexts in $\hat{P}_a(w\,|\,h)$ may be expected to be better trained than others. The interpolation parameters $\lambda(h)$ were determined iteratively by means of the EM algorithm [7] as follows:

$$\hat{\lambda}(h) \quad = \quad \frac{1}{N_w(h)} \sum_{\forall w \in *|h} \frac{\lambda(h) \cdot P_b(w|h)}{\lambda(h) \cdot P_b(w|h) + (1-\lambda(h)) \cdot \hat{P}_a(w|h)}$$

Here $\hat{\lambda}(h)$ is the updated interpolation parameter and $N_w(h)$ is the number of words in the adaptation set occurring in the context h. The summation is over all such words in the adaptation set, i.e.:

$$N_w(h) \quad = \quad \sum_{\forall w \in *|h} 1$$

Since there are generally few adaptation data, it is not possible to train interpolation parameters for each context h. Hence the set of histories were clustered according to their occurrence counts in a fashion similar to that presented in [2]. Starting from the lowest occurrence counts, contexts are merged successively into clusters so that each cluster is seen at least a threshold number of times in the adaptation data. This threshold was determined empirically by optimising the perplexity of interpolated models on the development test set. A value of 10 was found to yield good results, but the performance of the interpolated models was seen to be weakly dependent on the precise value. This clustering procedure ensured that contexts seen a larger number of times were the sole occupants of their cluster, while contexts seen too few times for the EM optimisation to work reliably were grouped with other contexts appearing a similar number of times.

This form of linear interpolation has been shown to be a well-performing variant of MAP adaptation [2].

A number of authors have proposed clustering the training corpus according to topic, and then adapting the language model by linear interpolation using data from the most relevant clusters [13] or from all clusters [7], [3].

4.2 MDE adaptation

Minimum discriminant estimation (MDE) has been applied successfully to supervised language model adaptation in [6]. First the adaptation data is used to estimate a unigram distribution $P_a(w)$. The MDE method then finds an adapted language model $P_a(w|h)$ with the smallest Kullback-Leibler distance to the background language model $P_b(w|h)$ while maintaining $P_a(w)$ as its marginal distribution, i.e.:

$$\sum_h P_a(w|h) \cdot P_a(h) \quad = \quad P_a(w) \quad \forall w$$

Since a closed-form solution to this problem is not available, it is normally determined iteratively by means of the Generalised Iterative Scaling (GIS) algorithm. This algorithm is very numerically intensive, so we employ the approximate solution presented in [6]:

$$P_a(w|h) \quad = \quad \frac{\alpha(w) \cdot P_b(w|h)}{\sum\limits_w \alpha(w) \cdot P_b(w|h)}$$

where

$$\alpha(w) \quad = \quad \left(\frac{P_a(w)}{P_b(w)}\right)^{\beta}$$

This can be shown to correspond to an approximate single iteration of the GIS algorithm. A value of $\beta = 0.5$ was taken for all our experiments, as recommended in [6].

The unigram probabilities were estimated using absolute discounting and a minimum *n*-gram count of zero [1].

MDE is well-suited to adaptation in situations where there is very little adaptation data, since the technique requires only the estimation of a unigram distribution of the target domain, and no higher-order *n*-gram distributions.

5 Experimental evaluation

The two techniques described in section 4 were applied to the lecture speech task introduced in section 3. This section describes the experimental setup and presents perplexity and word error-rate results.

5.1 Baseline language model

The 413K words present in the reference transcription of the 144 training speeches were used to train a backoff tri-

gram language model [4] using the CMU language modelling toolkit [1]. All experiments employed a closed vocabulary comprising the approximately 13K distinct words found in all 158 transcriptions in the lecture speech task. The trigram language model included all bigrams but excluded trigrams occurring only once. A minimum count of 12 was specified for unigrams. These parameters were determined by approximately optimising the perplexity measured on the reference transcription of the development-test (dev-test) set. The resulting model contains 109K bigrams and 45K trigrams, and gives perplexities of 130.14 and 122.68 on the dev-test and eval-test set reference transcriptions respectively.

5.2 Acoustic models

A preexisting set of acoustic models that had been trained on Japanese read speech was available for the purposes of this research. Baseline acoustic models were obtained by retraining these models on the 144 speeches in the training set. This resulted in a set of tree-based state-clustered speaker-independent cross-word triphone models with 2000 states, 16 Gaussian mixtures per state and diagonal covariance matrices. The acoustic parameterisation consisted of 12 MFCCs, energy, and deltas, resulting in 26-dimensional feature vectors.

5.3 Speech recognition engine

Decoding was performed with a time-synchronous beam-search decoder that performs the Token-Passing procedure [16] in a composition of Weighted Finite State Transducers [15]. The search is performed on each complete lecture speech in a single time-synchronous Viterbi-decoding run without incorporation of other means of segmentation.

The decoder makes use of a precompiled search network that includes the HMM structure, dictionary and the baseline unigram language model. The respective trigram deviation language models are composed on-the-fly. In this respect, on-line transducer composition offers a convenient approach to decoding with modified language models that does not require expensive precomputation of the resulting transducer composition [15].

5.4 Adaptation by text selection

In order to evaluate language model adaptation by text selection, the algorithm described in section 4.1 was used to identify lecture speeches in the training set similar to the recognition hypothesis dev-rec0 obtained by decoding the dev-test set with the baseline language model LM0. The speeches identified in this way were used to adapt the baseline language model LM0 by means of linear interpolation, resulting in an adapted language model LM1. A second iteration was then performed in which a new set of lecture speeches are identified using the recognition hypothesis dev-rec1 obtained by decoding the dev-test set with the already adapted language model LM1. These speeches were once again used to adapt LM1 by means of linear interpolation to yield LM2. The process is illustrated in Figure 2.

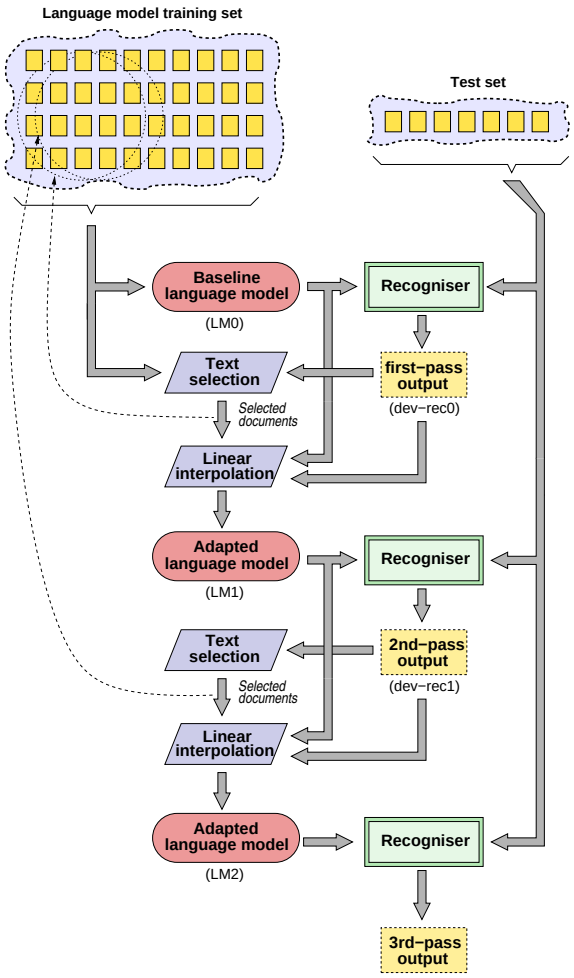

Figure 2: Language model adaptation by text selection.

Table 2 shows the perplexity of the adapted language model LM1 measured both on the development-test reference transcription (dev-ref) as well as the recognition hypothesis (dev-rec0) for a number of different choices of the parameter γ used in equation 1. The table shows a minimum at $\gamma = 0.35$ for the perplexity measured both on dev-ref and on dev-rec0. This strong correlation is remarkable, particularly since the high word error-rate implies that dev-ref and dev-rec0 differ significantly. The minimum is quite shallow and therefore the exact value of γ does not appear to be critical.

Threshold γ	Avg. selected documents	Perplexity	
		dev-ref	dev-rec0
0.1	89	124.98	102.21
0.15	66	123.70	101.59
0.25	38	122.43	101.02
0.35	24	**121.83**	**100.73**
0.50	10	122.78	101.05

Table 2: Optimisation of the threshold γ.

Table 2 also indicates the average number of documents selected by the test-selection procedure for each of the 7 speeches in the test set.

Language model	Perplexity			WER
	dev-ref	dev-rec0	dev-rec1	%
LM0	130.14	107.16	-	33.5
LM1	121.83	100.73	100.04	32.8
LM2	121.49	-	99.14	32.7

Table 3: Adaptation by text selection (dev-test).

Table 3 shows the recognition results for both iterations of adaptation using $\gamma = 0.35$ as determined from Table 2. Adaptation has lead to a 2.4% relative reduction in word-error rate and a 6.6% relative reduction in perplexity measured on the reference transcription (dev-ref).

In order to determine how robust the text selection measure is to the relatively high number of recognition errors in the transcription, adaptation was also carried out using the reference transcription (dev-ref) instead of the recognition outputs (dev-rec0 and dev-rec1), as illustrated in Figure 3.

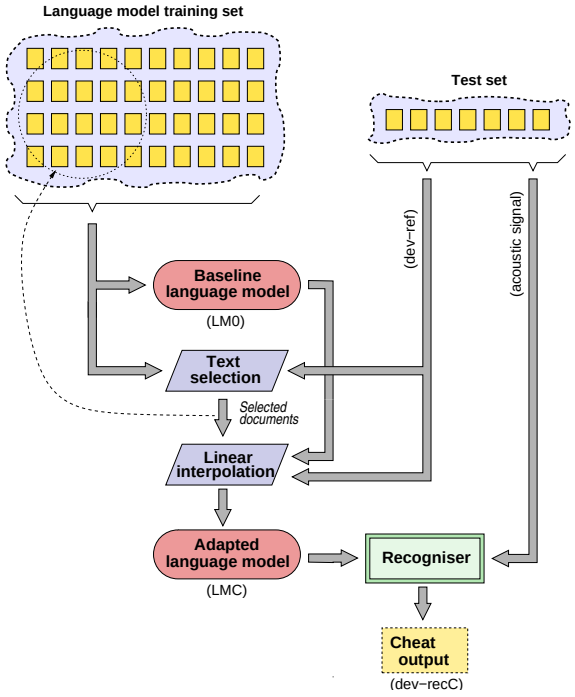

Figure 3: Language model adaptation by cheated text selection.

This experimental setup amounts to cheating, since the reference transcription is employed during adaptation. Its purpose is to measure an upper bound on the performance that can be achieved by text selection as a method of adaptation. The results are shown in Table 4.

Language model	Perplexity	WER
	dev-ref	%
LM0	130.14	33.5
LMC	119.01	32.7

Table 4: Adaptation by cheated text selection (dev-test).

From these results we see that the recognition error rate of 32.7% achieved when cheating is the same as that achieved after 2 iterations of unsupervised text selection (Table 3), and very close to that achieved after a single iteration of the same. This suggests that the text-selection measure is a highly robust to recognition errors, making it a particularly good choice for unsupervised adaptation.

5.5 MDE adaptation

In this case, the baseline language model LM0 is adapted by MDE as described in section 4.2 using as adaptation material the recognition hypothesis dev-rec0 obtained from the recognition pass with LM0. This results in a new language model LM3. A further recognition experiment using LM3 yields a new recognition hypothesis dev-rec2 which is used to perform a second iteration of MDE to produce LM4. The results of this process are presented in Table 5 and the process illustrated in Figure 4.

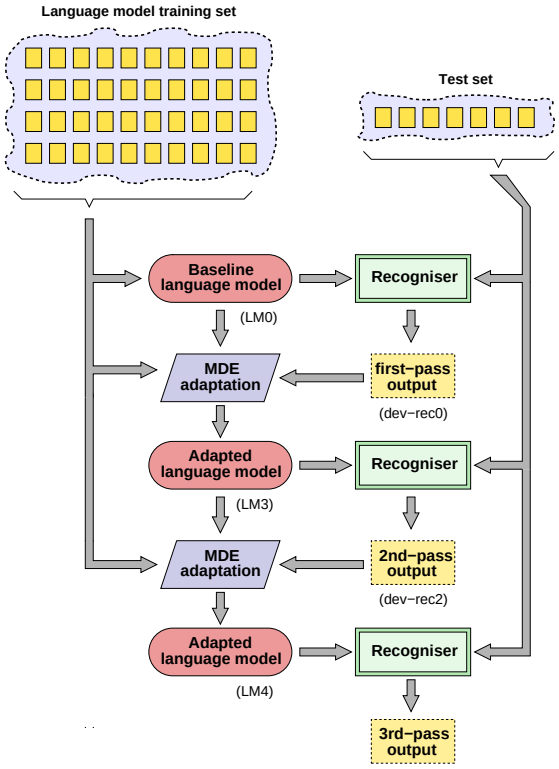

Figure 4: Language model adaptation by MDE.

Language model	Perplexity			WER
	dev-ref	dev-rec0	dev-rec2	%
LM0	130.14	107.16	-	33.5
LM3	89.12	66.96	65.41	31.8
LM4	87.95	67.26	63.27	31.7

Table 5: Adaptation by MDE (dev-test).

From Table 5 we see that a single iteration of MDE achieves a 5.1% relative decrease in the word error-rate and a 31.5% relative decrease in perplexity measured on the reference transcription (dev-ref). Hence the improvements are much larger than for text selection as presented in sec-

tion 5.4. The second iteration of MDE adaptation achieves much smaller improvements.

5.6 Combined adaptation

Table 6 presents perplexity and recognition results when performing text selection and MDE adaptation in succession. Text selection is performed first to update the baseline language model LM0 using the recognition hypothesis dev-rec0, as in Table 3. The resultant language model LM1 is then adapted by MDE, again using dev-rec0, to yield a new language model LM5. The perplexity of 86.30 and word error-rate of 31.7% are slightly better than those achieved in Tables 3 and 5 by applying just one of the adaptation methods.

Language model	Perplexity				WER
	dev-ref	dev-rec0	dev-rec3	dev-rec4	%
LM0	130.14	107.16	-		33.5
LM1	121.83	100.73	-	-	32.8
LM5	86.30	64.78	64.52	-	31.7
LM6	86.28	-	64.50	-	-
LM7	79.18	-	54.36	55.18	31.2
LM8	79.06	-	-	55.18	-
LM9	78.22	-	-	51.28	31.2

Table 6: Adaptation by text-selection and MDE (dev-test).

Another two iterations of combined adaptation were performed, and the results are included in Table 6 (refer to Table 8 for a key to the abbreviations used). The second iteration of text-selection followed by MDE leads to significant further improvements, while the third iteration shows no significant further gains. Overall the development-test word error rate has been improved by 6.9% relative. The first two iterations of the adaptation process are illustrated in Figure 5.

Finally, Table 7 shows the corresponding set of experiments applied to the evaluation-test set. Improvements are smaller than for the development-test set but show a similar tendency.

Language model	Perplexity				WER
	eval-ref	eval-rec0	eval-rec3	eval-rec4	%
LM0	122.68	92.11	-		36.9
LM10	113.62	86.54	-	-	-
LM11	89.85	62.40	62.14	-	35.8
LM12	89.72	-	62.12	-	-
LM13	86.62	-	55.23	56.23	35.7
LM14	86.30	-	-	56.16	-
LM15	88.18	-	-	53.34	35.7

Table 7: Adaptation by text-selection and MDE (eval-test).

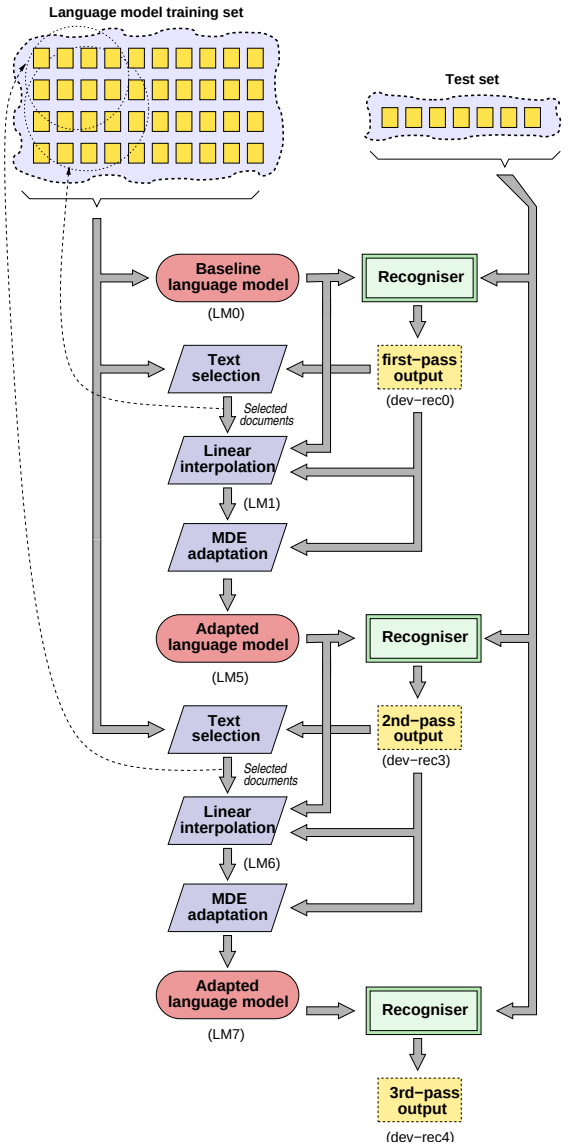

Figure 5: Language model adaptation by text selection and MDE.

6 Summary and conclusions

We have evaluated two methods of unsupervised language model adaptation. Both methods were able to reduce language model perplexity as well as the recognition word error-rate for a Japanese large vocabulary transcription task. When used in conjunction with one another, further improvements were achieved. The text-selection distance metric in particular has been demonstrated to be highly robust to speech recognition errors.

These results are promising, especially in view of the small amount of language model training data that was available and its highly constrained nature. They demonstrate the successful adaptation of the language model to the topic and style of each speaker in an unsupervised manner. The extension of these methods to larger text corpora, the incorporation of confidence measures and the combination with unsupervised acoustic model adaptation remains the subject of ongoing work.

Label	Description
LM0	Baseline (background) trigram language model.
dev-ref	Reference transcription for dev-test set.
dev-rec0	Dev recognition hypothesis using LM0.
LM1	LM0 adapted by text selection on dev-rec0.
dev-rec1	Dev recognition hypothesis using LM1.
LM2	LM0 adapted by text selection on dev-rec1.
LM3	LM0 adapted by MDE on dev-rec0.
dev-rec2	Dev recognition hypothesis using LM3.
LM4	LM3 adapted by MDE on dev-rec2.
LM5	LM1 adapted by MDE on dev-rec0.
dev-rec3	Dev recognition hypothesis using LM5.
LM6	LM5 adapted by text selection on dev-rec3.
LM7	LM6 adapted by MDE on dev-rec3.
dev-rec4	Dev Recognition hypothesis using LM7.
LM8	LM7 adapted by text selection on dev-rec4.
LM9	LM8 adapted by MDE on dev-rec4.
eval-ref	Reference transcription for eval-test set.
eval-rec0	Eval recognition hypothesis using LM0.
LM10	LM0 adapted by text selection on eval-rec0.
LM11	LM10 adapted by MDE on eval-rec0.
eval-rec2	Eval recognition hypothesis using LM11.
LM12	LM11 adapted by text selection on eval-rec2.
LM13	LM12 adapted by MDE on eval-rec2.
eval-rec3	Eval recognition hypothesis using LM13.
LM14	LM13 adapted by text selection on eval-rec3.
LM15	LM14 adapted by MDE on eval-rec3.

Table 8: Legend for labels used in Tables 3 to 7.

Acknowledgment

We thank Dr. Shigeru Katagiri for supporting the authors' corporative work. The lecture speech data and transcription were provided by the Japanese Science and Technology Agency Priority Program "Spontaneous Speech: Corpus and Processing Technology".

References

[1] P. Clarkson and R. Rosenfeld. Statistical language modelling using the CMU-Cambridge toolkit. In *Proc. Eurospeech*, pages 2707–2710, Rodos, Greece, 1997.

[2] M. Federico. Bayesian estimation methods for *n*-gram language models. In *Proc. ICSLP*, pages 240–243, Philadelphia, 1996.

[3] A. Kalai, S. Chen, A. Blum, and R. Rosenfeld. Online algorithms for combining language models. In *Proc. ICASSP*, Phoenix, Arizona, 1999.

[4] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Trans. ASSP*, 35(3):400–401, March 1987.

[5] D. Klakow. Selecting articles from the language model training corpus. In *Proc. ICASSP*, Istanbul, Turkey, 2000.

[6] R. Kneser, J. Peters, and D. Klakow. Language model adaptation using dynamic marginals. In *Proc. Eurospeech*, pages 1971–1974, Rodos, Greece, 1997.

[7] R. Kneser and V. Steinbiss. On the dynamic adaptation of stochastic language models. In *Proc. ICASSP*, pages 586–589, Minneapolis, 1993.

[8] R. Kuhn and R. de Mori. A cache-based natural language model for speech recognition. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 12(6):570–583, June 1990.

[9] M. Mahajan, D. Beeferman, and X.D. Huang. Improved topic-dependent language modelling using information retrieval techniques. In *Proc. ICASSP*, Phoenix, Arizona, 1999.

[10] T.R. Niesler and P.C. Woodland. Variable-length category-based statistical language models. *Computer, Speech and Language*, 13:99–124, January 1999.

[11] R. Rosenfeld. *Adaptive statistical language modelling: a maximum entropy approach.* Ph.D. Dissertation CMU-CS-94-138, School of Computer Science, Carnegie Mellon University, April 1994.

[12] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[13] K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In *Proc. Eurospeech*, Rodos, Greece, 1997.

[14] T. Shinozaki, C. Hori, and S. Furui. Toward automatic transcription of spontaneous speech. In *Proc. Eurospeech*, pages 491–494, Aalborg, Denmark, 2001.

[15] D. Willett and S. Katagiri. Recent advances in efficient decoding combining online transducer composition and smoothed language model incorporation. In *Proc. ICASSP*, Orlando, 2002.

[16] S.J. Young. *Token passing, a simple conceptual model for connected speech recognition systems*. Technical Report TR38, Cambridge University, 1989.