An investigation into a Simulation of Episodic Memory for Automatic Speech Recognition

Viktoria Maier & Roger K. Moore

Department of Speech and Hearing
University of Sheffield, Sheffield, United Kingdom
Viktoria@dcs.shef.ac.uk, r.k.moore@dcs.shef.ac.uk

Abstract
This paper investigates a simulation of episodic memory known in the literature as ‘MINERVA 2’. MINERVA 2 is a computational multiple-trace memory model that successfully predicts basic findings from the schema-abstraction literature. This model has been implemented and tested on a simple ASR task using vowel formant data taken from the Peterson & Barney database. Recognition results are compared to a number of state-of-the-art pattern classifiers, and it is shown that the episodic model achieves the best performance.

1. Introduction
Automatic speech recognition (ASR) is now a long established research field. From the early years of template-based isolated digit recognition in the 1950s up to the latest large-vocabulary continuous speech recognition (LVCSR) systems, the field has benefited from the introduction of many new ideas and has thus made significant progress [1]. As is well known, most contemporary state-of-the-art systems are based on the use of hidden Markov models (HMMs); a formal statistical framework that is used to capture the immense variability in speech by training context-dependent sub-word models on substantial quantities of recorded speech data.

It can be argued that the introduction in the 1980s of HMMs has been the main catalyst for the steady improvements in recognition accuracy that have been achieved over the past twenty years. However, HMMs are not without their shortcomings; many assumptions are made about the nature and structure of speech signals, and some of these (e.g. it is assumed that speech is generated as the result of a 1st-order Markov process) are patently false. Nevertheless, the advantages of using statistics to model our lack of knowledge about the detailed structure and dependencies in speech, currently far outweigh the disadvantages arising from poor approximations to reality.

However, in recent times it has become apparent that, not only is ASR performance an order of magnitude worse than human speech recognition (HSR) [2], but the incremental improvements in state-of-the-art systems are also asymptoting to a level of performance that is well short of that required for many practical applications [3]. The consequence of this is that a number of researchers are beginning to explore the relationship between ASR and HSR in more detail [4]. At the same time, research in the field of HSR has seen the introduction of computational models based on search algorithms that have strong overlaps with the techniques used in contemporary ASR [5]. Unfortunately, such models typically use a phonetic representation as input, and do not link directly with real speech signals. An attempt by Scharenborg et al to combine a state-of-the-art HSR model with a state-of-the-art ASR front-end resulted in performance comparable to a conventional ASR system [6]. There is thus considerable interest in finding an approach in which the benefits of ASR and HSR combine to produce performance that is in advance of conventional ASR.

One area in which the conventional approach to ASR may be questioned is in the use of statistical distributions. Clearly probability density functions (pdfs) constitute an essential element of hidden Markov modelling and provide a powerful method of generalising from seen to unseen data. However, the use of pdfs does represent a potential loss of information; the detail that is present in individual data samples is sacrificed in order to pool information in a controlled fashion. This debate between the storage of individual tokens versus the derivation of an abstract representation based on a set of tokens is a familiar one in the psychological literature relating to human memory [7][8][9] (and to HSR [10][11][12][13]), and there is considerable interest in the topic of ‘episodic memory’ in which there is evidence that individual memory ‘traces’ are stored. The potential relevance of this to ASR has already been realised by a few researchers, and some early work is moving in the direction of using template-based rather than HMM-based approaches for LVCSR [14][15].

This paper introduces a simulation of episodic memory known as ‘MINERVA 2’ [16]. MINERVA 2 is a computational multiple-trace memory model that successfully predicts basic findings from the schema-abstraction literature. This model has been implemented and tested on a simple ASR task using vowel formant data taken from the Peterson & Barney database [17]. Recognition results are compared to a number of state-of-the-art pattern classifiers, and it is shown that the episodic model achieves the best performance.

2. MINERVA2
MINERVA 2 simulates episodic memory by first storing ‘traces’ (records of memory experiences or episodes). Inputs to the system - ‘probes’ - are compared to all of the traces in memory, and the retrieved ‘echo’ (essentially a weighted composite of the stored traces) contains more knowledge about the input, e.g. its class. The weights that are used depend on the similarity between the input and each trace. Hintzman has also shown that the model is able to create abstract representations of stored data. Further, by probing repetitively with the abstracted representations (a process referred to as ‘echoes of echoes’), it is possible to refine the response and exploit the relationships between stored traces.

The main parameters of the model are (i) the feature representations, (ii) the similarity function, (iii) the weighting function (also called activation function), and (iv) the echo retrieval function.
2.1. Feature Representations

The feature values in MINERVA 2 vary between 1, -1 and 0, where ‘1’ represents excitation of a property, ‘-1’ represents inhibition of a property and ‘0’ represents indeterminate information. Classification labels are stored as blocks of features. For example, in a simulation of a three class problem, Hintzman used ten values for the category name.

Note that the feature value ‘0’ can be used not only to represent those features that are not known (for example, the class label to which an unknown input pattern belongs), but also for modelling the loss of information in ‘forgetting’. It also has implications within the main functions of the model.

2.2. Similarity Measure

The restricted feature values (1, -1 and 0) mean that it is possible to calculate the similarity \( \text{sim}_{i,j} \) using the following expression:

\[
\text{sim}_{i,j} = \left( \frac{1}{r} \sum_{j=1}^{N} I t_j \right)^p \quad (2.1)
\]

... where \( I_i \) is the \( i^{th} \) feature of the input vector \( t_i \), \( I_t \) is the \( i^{th} \) feature of the trace \( t \), \( N \) is the total number of features and \( r \) is the number of features relevant to the comparison (i.e. where either \( I_i \) or \( t_i \) is ≠ 0).

2.3. Activation Function

To gain the final weighting \( w \) of the traces with respect to input \( I \), the similarity measure is raised to the power of \( p \). This, as Hintzman said, in effect increases the signal-to-noise ratio in the signal, i.e. it gives more weight to the most similar traces and less to those traces that are not similar.

\[
w_{I,j} = \text{sim}_{i,j}^p \quad (2.2)
\]

Hintzman sets the value of \( p \) to three. He states however, that other values are permissible as long as the sign of \( \text{sim}_{i,j} \) is retained. Thus in MINERVA 2, \( p \) is restricted to odd values.

2.4. Echo Intensity

Echo intensity is a measure of how much activation has been triggered. The more traces that match the input, and the more similar they are to the input, the greater the value of \( I \). Echo intensity can be used to determine the judgement of frequency and familiarity; it is defined as follows:

\[
\text{int}_I = \sum_{j=1}^{T} w_{I,j} \quad (2.3)
\]

... where \( I \) is the input, \( T \) is the total number of traces stored. The intensity is a value that may or may not play a role, depending on the task.

2.5. Echo Retrieval

The echo is the abstraction of the stored traces as a response to the input. This is accomplished by deriving a weighted sum of all traces in memory. The echo then becomes:

\[
\text{echo}_I = \sum_{i=1}^{T} w_{I,j} \cdot \text{trace}_i \quad (2.4)
\]

... where \( w_{I,j} \) is the weight on trace \( t \) for input \( I \), and \( T \) is the number of stored traces.

2.6. Minerva Classification

Classification in MINERVA 2 faces a problem called ‘ambiguous recall’, which means that the probe derived may not be a perfect match to the class. The problem is solved by passing a normalized echo back into the system as a probe. However, since it is computationally expensive, usually the classification decision is based on the ‘most likely’ class.

2.7. Relationship with K-Nearest-Neighbour

Clearly MINERVA 2 would appear to share many similarities with the well known pattern classification technique known as ‘k-nearest-neighbour’ (KNN). Essentially, a KNN classifier labels input \( I \) with the label of the majority of the \( k \) nearest neighbours.

\[
\text{knnClass}_I = \sum_{k=1}^{N} s_{I,k} \cdot \text{cont}_{I,k} \quad (2.5)
\]

... where \( s_{I,k} \) is the similarity measure between input \( I \) and stored exemplar \( k \in \text{knn} \) nearest neighbour and

\[
\text{cont}_{I,k} = \begin{cases} 1 & \text{if } k \in c \\ 0 & \text{if } k \notin c \end{cases} \quad (2.6)
\]

These expressions are the same as equation (2.4) when: \( k = T \) (i.e. all traces are used), \( s_{I,k} = w_{I,k} \) and \( \text{cont}_{I,k} = \text{trace}_k \). Hence, KNN can be seen as a special case of MINERVA 2.

3. Experiments and Results

3.1. Peterson & Barney Vowel Data

The Peterson & Barney data [17] was selected for the recognition experiments because it is a well studied dataset in the pattern recognition literature, and the fact that it is based solely on the analysis of single frames of speech. The database comprises values for the first three formants and the fundamental frequency for a set of ten vowels spoken by 33 male, 28 female and 15 children. Each talker pronounced each vowel twice, giving a total of 1520 tokens.

3.2. Experimental Setup

All experiments were conducted using ERB-scaled frequencies [18] in order to give appropriate importance to each of the frequency parameters. Training/testing employed a ‘leave one out’ regime in which each system was trained on all data, apart from the vector being used for test. Results were obtained using standard GMM, KNN and SVM classifiers, as well as a modified version of MINERVA 2.

3.3. Modifications to MINERVA 2

In order to use MINERVA 2 for the classification task described above, it was necessary to modify certain aspects of the system. For example, the restricted value representation in MINERVA 2 had to be extended to handle the numeric values deriving from the input frequency features, as follows.
3.3.1. Feature Representation

In the implementation discussed here, the information used in the feature vector consisted of: the fundamental frequency and the three formant frequency values, the phoneme class affiliation and the gender information.

The frequency feature set and the two classification feature sets have to be encoded differently, as they represent two different types of information and have different requirements. For example, it is desirable for the classification features (i) to return some measure of certainty/confusion between classes and (ii) to use the same numerical distance function as the rest of the features. In order to satisfy (i), it is necessary to encode the classification features by occupying one feature per class. Thus the classification features were encoded by ‘place’ within this block of features rather than as a numeric value.

This means that for example, if it is necessary to store gender information using three different classes (male, female, child), then this would be stored as a three-valued vector, where the true gender X would be set to a value greater than zero, and the other two values would be set to zero.

\[
\begin{array}{ccc}
\text{male} & \text{female} & \text{child} \\
X & 0 & 0 \\
0 & X & 0
\end{array}
\]

Figure 1: Two examples for encoding the classification feature. In the first row, the features encode a male voice, in the second one it is a female. X is an arbitrary positive value.

Note that if every gender feature value set (e.g. [X,0,0] or [0,X,0]) stores the same values (even if in different order), it would not change the derived similarity ordering of the traces. However, it does change the derived similarity values. In particular, because of the normalization of the similarity values (see 3.3.2), the bigger the value for \( X \), the smaller the spacing of the derived values. Hence, in this implementation, \( X \) is set to 1 unless stated otherwise. By setting \( X \) to 1, and having one cell per class, it also possible to derive a confusion matrix and hence a basis for a confidence measure.

3.3.2. Similarity Measure

Since feature values are numeric, it was necessary to compute the similarity using an intermediate step in which the distance between the input and stored traces is computed. Currently, the distance measure used is the Euclidean Distance (ED):

\[
ED_{I,t} = \sqrt{\sum_{i=1}^{n} (|I_i - t_i|)^2}
\]

(2.7)

… where \( I_i \) is the \( i \)th feature of the input vector and \( t_i \) is the \( i \)th feature of the trace \( t \).

The similarity between the input \( I \) and the trace \( t \) is then computed by:

\[
sim_{I,t} = 1 - \left( \frac{ED_{I,t}}{\max(ED_I)} \right)
\]

(2.8)

… where \( ED_{I,t} \) is the vector of length \( n \), with \( n \) equal to the number of features, and \( \max(ED_I) \) is the maximum value in the vector. It is necessary to normalize \( ED \) in order to ensure that the range of \( \text{sim}_{I,t} \) is between 0 and 1.

In MINERVA 2, if a feature value is set to ‘0’ in either the input or the trace, then that feature is not considered for calculating similarity (effectively annihilating any effect of that feature). To maintain this behaviour, if no value is specified for an information block in the input, then this feature is not used to calculate the similarity.

3.3.3. Weighting Function

The weighting function \( w \) is implemented as stated in equation (2.2). The value of \( p \) has been determined empirically. Note that in the current implementation, all values of \( \text{sim}_{I,t} \) are positive and therefore the restriction to odd values of \( p \) was not necessary.

3.3.4. Echo Function

Since the features are numerical in the modified version of MINERVA 2, it also became necessary to normalize the acquired echo values. The echo then becomes:

\[
\text{echo}_t = \left( \sum_{i=1}^{n} w_{I,t} \cdot \text{trace} \right) \text{int}_t
\]

(2.9)

… where \( w_{I,t} \) is the weight on trace \( t \) for input \( I \), and \( T \) corresponds to the number of stored traces.

3.3.5. Echoes of Echoes

Due to the limited numeric representation used in MINERVA 2 (0, -1 and 1), the ‘echoes of echoes’ process simply involved using the retrieved echoes as probes as is (although after normalization). When the same principle were applied to the extended range of numeric values used in the modified version of MINERVA 2, it was found that the process often failed to converge. Therefore, the new information derived from a first-pass (i.e. the classification features) are appended to the original frequency features.

3.4. Results

The results of the main recognition experiments are presented in Table 1. The best GMM result was obtained with five Gaussian mixture components, the best KNN result was found using \( k = 8 \), the best SVM result was obtained using a standard SVM with a spherically normalised polynomial kernel of order 2 and a regularisation parameter of 10000 and the modified version of MINERVA 2 performed best with \( p = 37 \) (although it was not very sensitive to the actual value of \( p \)).

As can be seen from Table 1, the episodic model achieves the best overall result.

Table 1: Recognition results.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Mixture Model (n=5)</td>
<td>11.78%</td>
</tr>
<tr>
<td>K-Nearest-Neighbour (k=8)</td>
<td>11.58%</td>
</tr>
<tr>
<td>Support Vector Machine (o=2)</td>
<td>10.72%</td>
</tr>
<tr>
<td>Episodic Model (p=37)</td>
<td>10.53%</td>
</tr>
</tbody>
</table>

The result for the episodic model shown in Table 1 was derived using an implementation where the distance measure
was computed using only the frequency features, and not the
classification features. When all of the features are used,
some of the similarity value range (from 0 to 1) is taken up by
this numerically static information, and it was expected that
this would have a negative impact on performance. This was
confirmed - the error rate dropped to 10.59% ($p = 120$).

The results using the ‘echoes of echoes’ process are
shown in Table 2 (for three values of gender category and
with the phoneme classification variable set to 1). The
performance is marginally worse than that shown in Table 1.

5. Conclusion

This paper reports on the investigation of a simulation of
episodic memory known as MINERVA 2 - a computational
multiple-trace memory model that successfully predicts basic
findings from the schema-abstraction literature. A modified
version of MINERVA 2 has been implemented and tested on a
simple ASR task using vowel formant data taken from the
Peterson & Barney database. Recognition results have been
compared to state-of-the-art GMM, KNN and SVM pattern
classifiers, and it is shown that the episodic model achieves
the best performance (although the differences are not
statistically significant).

Although this research is only a first step towards the
challenges posed by ASR, it has nevertheless shown that it is
possible to achieve a positive result using a psychologically-
motivated approach based on episodic memory. The next
steps in the research are (i) to move to a multi-frame
environment, (ii) to use ‘standard’ (MFCC) features, and (iii)
to further develop the episodic model.

6. Acknowledgements

The authors would like to thank Vincent Wan for running the
SVM machine experiments on his system.

7. References

[2] Lippmann, R., “Speech Recognition by Machines and
Automatic Speech Recognition Systems and Human
SPRAAC Workshop on Human Speech Recognition as
Pattern Classification*, Max-Planck-Institute for
Continuous Speech Recognition”, *Cognition*, 52(3):163-
Recognition using Automatic Speech Recognition
Paradigms in SpeM”, *Proc. Eurospeech*, 2285-2288,
2003.
R., “Episodic Recognition Memory for High-
Dimensional, Human Synthetic Faces” [Abstract],
for Visual Textures: A roving probe gathers some
memory”, *Psychological Science*, In press.
Representations for Spoken Words”, *Memory and
Spoken Word Identification and Recognition Memory”,
*Journal of Experimental Psychology: Learning Memory
Theory of Lexical Access”, *Psychol Rev*, 105(2):251-79,
1998.
Learning in Speech”, *Cognitive Psychology*, 47, 204–
Markov Models with dynamic time warping for speech
recognition”, *Proc. IEEE ICASSP 2004*.
Wambaq, P., “Data Driven Example Based Continuous
[16] Hintzman, D. L., “Schema-Abstraction in a Multiple-
Trace Memory Model”, *Psychological Review*, 93: 411-
427, 1986.
in a study of the vowels”, *J. Acoust. Soc. Am.* 24, 175-
184, 1952
[18] Moore, B. C. J., *An Introduction to the Psychology of

<table>
<thead>
<tr>
<th>Gender Classification</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>X=1</td>
<td>10.66%</td>
</tr>
<tr>
<td>X=3</td>
<td>10.72%</td>
</tr>
<tr>
<td>X=10</td>
<td>11.12%</td>
</tr>
</tbody>
</table>