

Temporal Episodic Memory Model: An Evolution of MINERVA2

Viktoria Maier, Roger K. Moore

Department of Speech and Hearing
University of Sheffield, Sheffield, United Kingdom

V.Maier@dcs.shef.ac.uk, r.k.moore@dcs.shef.ac.uk

Abstract

This paper introduces a new model for automatic speech recognition (ASR) called TEMM - Temporal Episodic Memory Model. TEMM is derived from a simulation of human episodic memory called MINERVA2, and it not only overcomes the inability of MINERVA2 to use temporal sequence for recognition flexibly, but it also employs a prediction mechanism as an additional source of information. The performance of TEMM on an ASR task is compared to state-of-the-art HMM/GMM baseline systems, and a first analysis shows both promising results and a need to further stabilise the consistency of the output of the new model.

Index Terms: Episodic memory, exemplar-based ASR, MINERVA2, prediction

1. Introduction

It has become apparent that the performance of state-of-the-art automatic speech recognition (ASR) systems is asymptoting at a level that falls short of that which is desirable for many advanced applications [1] let alone being comparable with the capabilities of a human listener [2]. As a consequence, a number of researchers are exploring the field of human speech recognition (HSR) in order to better understand the nature of speech, and to investigate the possibility that a simulation of the human speech recognition system might lead to more competitive and robust ASR [3]. In particular, there is growing interest in the possible implications of ‘episodic’ memory for perceptual tasks, and a number of HSR researchers are investigating an ‘exemplar based’ approach [4][5][6]. The main reason for this rise in interest is that the flexibility and robustness exhibited by HSR is not able to be modelled adequately with an architecture based on pre-abstracted representations. An exemplar-based approach (such as [7]) offers a mechanism for retaining and accessing the ‘fine phonetic detail’ [8] that is discarded in purely abstract representations such as hidden Markov models (HMMs).

In a previous paper [9] the authors presented a vowel recognition system based on Hintzman’s computational multiple-trace (episodic) memory model known as MINERVA2 [10]. MINERVA2 is interesting because of its capacity to retain detailed traces together with its ability to abstract away from this detailed information to new and unseen input. The model performed very well in comparison to support-vector-machine (SVM), Gaussian mixture model (GMM) and k-nearest-neighbour classifiers on the Peterson & Barney dataset. However, this was a simple speech-related task, involving *no temporal information* that had been chosen specifically because MINERVA2 was unable to model temporal structure.

This paper presents a new ‘temporal episodic memory model’ (TEMM) which, although based on MINERVA2, incorporates both temporal structure and a mechanism for

prediction (another important property of the human processing mechanism [11]). TEMM has been assessed in comparison to a standard HMM classifier on the TI-ALPHA isolated word database [12].

2. Temporal episodic memory model

As explained in [9], in response to a *probe*, MINERVA2 constructs an *echo* by weighting all samples in the training data - see Figure 1:

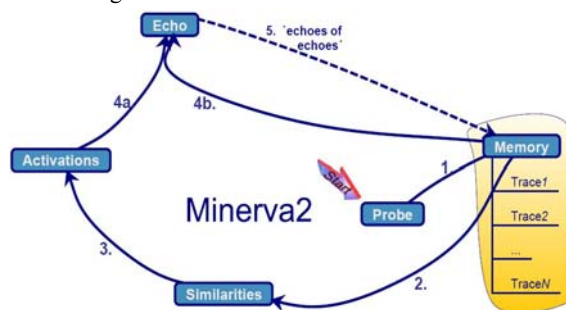


Figure 1: Schematic diagram of MINERVA2.

While MINERVA2 showed competitive results on a single-frame task (where a frame is the result of an analysis window applied to the speech wave), a big disadvantage of the model is that it is unable to model temporal information since the output that is created is constructed by combining all traces in the training data. Clearly the temporal evolution of speech is crucial. Therefore, a model that aspires to use fine phonetic detail in the speech signal needs to be able to exploit sequence information. However, the only way of adding sequence to MINERVA2 is by supplying it explicitly in the features used (where features refers to the chosen signal representation, such as MFCC), for example by using derivatives or by encoding traces that span more than one frame. Unfortunately, these options have the severe disadvantage of being both hard-wired and limited in scope. Therefore, further development of MINERVA2 is needed in order to overcome its temporal-modelling limitations while at the same time preserving its positive characteristics and appealing simplicity.

TEMM is thus an attempt to overcome such limitations by expanding the MINERVA2 framework to accommodate temporal information. In addition, TEMM extends the temporal framework to accommodate *prediction*, a behavioural feature that has been posited as a key aspect of intelligence [11].

As the base operation, TEMM follows the principles of MINERVA2 as laid out in the schematic in Figure 1. In so doing, the system acquires knowledge about how well each trace (i.e a memory representation containing features and their classification) in the database fits the current input data. By acknowledging that speech frames are in fact not

10.21437/Interspeech.2007-319

independently and identically distributed (as assumed by the HMM framework) but instead correlated, it is plausible to assume that the current fit of data has some correlation with the next input frame. This assumed relationship is used to derive a prediction of the features corresponding to the next input frame.

2.1. TEMM and feature prediction

Feature prediction is a central part of TEMM. The fit of the predictions to the input data and how discriminating those predictions are with respect to the next best class provides an indication of (a) the goodness of fit of previous decisions (i.e. future decisions can influence past decisions), and (b) the goodness of fit of current data to future data.

The prediction step fits neatly into the overall TEMM framework; by using the acquired similarity, or activation, of traces to input frames, it is possible to construct predictions for the features of the next input frame. Since it is speech recognition that is of interest, the competition between different classes is of primary importance. So, predictions are constructed for the features of each possible class. Figure 2 illustrates this process for a two-class problem:

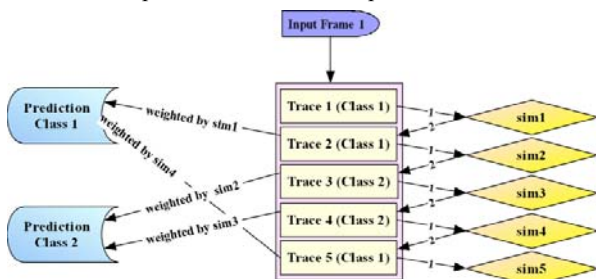


Figure 2: Schematic diagram of the process of feature prediction. Arrows numbered (1) are the normal process of computing similarities. Arrows numbered (2) show the new assignment of the similarity data for prediction computation.

The prediction step serves as a method to generalise from training data. It also means that the model has an in-built immediate assessment of the outcome of the previous step. If the predictions fit the next input frame well, it adds credibility to the adequate use of the training data for the assessment of the input data that created the similarity/activation measures. On the other hand, if the prediction does not fit the input data at time $t+1$, then it may be necessary to question the data's assessment of classification of input frame t as well. This has further implications which are discussed later.

It should be pointed out that feature predictions can and possibly should model context-dependent predictions (depending on the data structure in the database used). For example, in a phone-based recognition task there would be merit in creating different feature predictions for different class contexts. This means that, for the two-class problem illustrated in Figure 2, there would be maximally four ('number of classes'^{squared}) different predictions (i.e. 1. Prediction 'Class 1 to Class 1'; 2. Prediction 'Class 1 to Class 2'; 3. Prediction 'Class 2 to Class 2'; 4. Prediction 'Class 2 to Class 1') instead of the two predictions that were used for the example in Figure 2. There is also the possibility that not all possible combinations are present in the database, in which case the "prior expectation" for this particular transition is zero.

As a consequence, the prediction step allows the model to keep track of how likely it is that the next input frame is going to belong to a particular class. This information is the same as the "intensity" of a prediction (corresponding to the summed activations that led to the prediction). I.e. a prediction's intensity corresponds to a prior expectation that the next frame belongs to the same class. The prediction intensity is used when updating activations (see Figure 4).

2.2. Use of temporal structure in TEMM

Temporal information in TEMM is introduced using the concept of a "trace unit" - a sequence of successive traces from the database. The database stores traces in *sequence*. So, the trace that follows any one trace in the database holds the frame that followed the previous frame in the speech signal. This means that trace units are blocks of traces (i.e. frame values and class information). These trace units hold an expanding context which, due to the fact that they preserve an accurate account of sequence in the original speech signal, contains the fine temporal information. Trace units expand as a function of the confidence associated with the classification of the input frames.

An important issue is when such trace units are formed. In section 2.1 it was argued that the fit of the prediction to the predicted input can be interpreted not only as a confidence measure for the previous decision, but also as an indication of how well the database with the current trace activations represents the new input data. If it seems that the database with the current trace activations represent the input data sufficiently well (see 2.3), then it is reasonable to allow continued use of the accumulated information. In this case, information is preserved by forming trace units that in effect are updated (trace) activations whose scope no longer only hold one single trace in the database, but a step-by-step (over time) growing sequence of frames.

A second issue is how these temporal units are formed and used. With each new input frame a prediction is computed for the features in each class. In the event that the prediction matches the predicted input frame well (enough), the scope of the trace units is expanded by one frame.

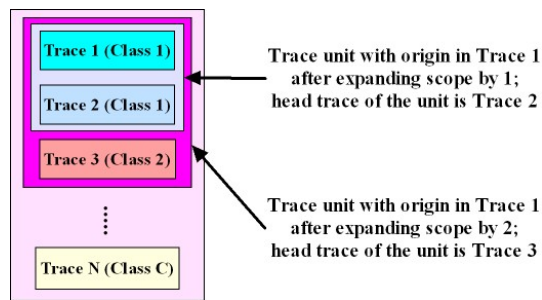


Figure 3: Schematic diagram of the process of forming trace units.

Expansion of a trace unit means updating the activation of the current trace (called the trace unit's "head") in a way that preserves information of the path that led to a particular trace unit's "head". The trace unit's "head" is the actual trace that the activation is attached to. A trace unit is however not just a variable-length version of a context trace (as can be used to introduce static context into MINERVA2). The difference lies not only in how the information is stored and hence accessed, but also how the activation is updated. Increasing the scope of a "trace unit" has been inspired by the HMM framework. In fact, it uses the form of an HMM with one state-per-frame.

To explain a trace unit in HMM’s terminology: the number of “states” in each “HMM” is increased by one with each extension of the trace unit. This means that a trace unit also uses “transition probabilities” from one frame to another. This “transition probability” is the likelihood of transiting from the class of the frame belonging to state x into the class of the frame belonging to state $x+1$. This value is computed at the time of prediction and is equal to the sum of all activation values used to compute the corresponding class prediction. In other words, the “transition probability” is a factor shared by all trace units whose head (and previous frame) is a member of the same class. The “transition probabilities” are a measure for how likely it is to go from class of trace X to the class of trace $X+1$ at a particular time.

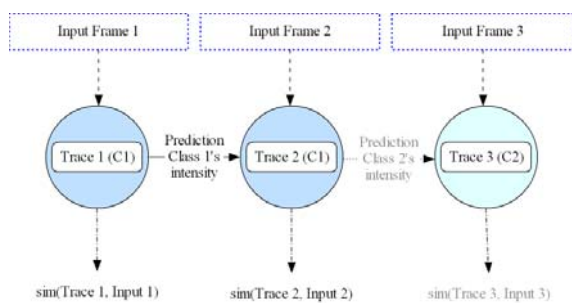


Figure 4: Schematic diagram of the application of a HMM-like topology to trace units.

By adding “transition probabilities” as an information source for updating the activations, trace units that belong to more likely classes are rewarded. This means that all trace units keep an indirect memory of how well the classes associated with its frames fit the assumed classes of the input. Class information is thus included into the trace units.

Trace units have a scope that is not limited by class labels, they can cover one frame but, as they expand it is theoretically possible that the trace units can cover words and even sentences. How many different classes are covered in each trace unit is dependent on the underlying classes of the traces, and varies from trace unit to trace unit. As a result, variable “units” (i.e. phones, words, etc.) can compete directly with each other - a concept also included in Grossberg’s ARTWORD model of human speech perception [13].

2.3. Classification in TEMM

TEMM performs Minerva2-type classification when it has no history to work from, i.e. either when the first input frame is submitted, or when a trace unit’s prediction has been disregarded as not a good fit to the data.

MINERVA2-type classification means that the current class is judged on the overall activation intensity of traces belonging to each class. When, however, “history” is available, the system has much more information available, e.g. activation values for trace units which cover a larger frame context (except for the first iteration). Further, the system predicts features for each class (context) available. The predictions are computed using the trace unit’s larger context incorporating history. So it can be argued that the predictions are a class-dependent summary of all the information available to the system at the time a prediction is made. Hence, it makes sense to use such summaries for further decisions. This means that when predictions for

current input frames are available, classification decisions are based on them.

To be precise, the classification decision is based on a comparison of the predictions to the actual predicted frame. In the example used in Figure 2 the predictions were derived using activations derived from input frame 1, thus the predictions relate to the features of frame 2. Therefore, in this case, the predictions are compared to the actual features of input frame 2. The best fitting prediction can be seen as the “most likely class at this point in time given the information available to the system”. However, it also makes sense to encode a threshold in the system which can forestall such a decision in the face of uncertainty, e.g. if the best two matching predictions are equally close to the actual features. In this case, a “backing off” to MINERVA2-style computation is encoded into the model. In this case, formed trace units are lost and the process begins anew from the beginning. Already classified input frames, however, are retained (a constraint which may be removed in future versions of TEMM).

2.4. Changes to the adapted MINERVA2 model

In this first version of TEMM it was felt that the MINERVA2 parameter features should be encoded as closely as possible to the adaptation of MINERVA2 for ASR outlined in [9]. However, it was necessary to adapt the similarity computation:

$$sim_{I,t} = 1 - (ED_{I,t} / \max(ED_{I,t})) \quad (1)$$

where $ED_{I,t}$ is the Euclidean Distance between the input vector and trace t .

This equation is dependent on the maximum ED value found at each input frame. Since the new model extends its scope of trace units to more than one time frame, the derived values at each input frame would not be based on the same scale and hence would lead to a variable weighting of the various comparison steps. This is highly undesirable (at least in such a random fashion) and hence in TEMM the similarity equation was changed to offer a more stable framework:

$$sim_{I,t} = 1 / (ED_{I,t} + \epsilon) \quad (2)$$

where $ED_{I,t}$ is the Euclidean Distance between the input vector and trace t , and ϵ is a factor added to avoid division by 0.

Except for this change, all equations in TEMM that relate to MINERVA2 remained the same.

3. TI-ALPHA experiments with TEMM

The database chosen for this investigation was the TI-ALPHA isolated word corpus because of the high confusability of the words set and its consequent high sensitivity to alternative recognition approaches. The data used consisted of two speakers, (one male (M1) and one female (F1)), uttering two letters of the orthographic alphabet – “S” and “J”. The training set consisted of 20 utterances per speaker and the test set consisted of 16 utterances per speaker. All experiments were conducted using MFCC features and a 25ms frame was taken every 10ms. Only one feature was used (for both TEMM and HMM) in order to minimize the influence of the distance measure used in TEMM. The classes corresponded to whole-word labels, and speaker-independent (SI) and speaker-dependent (SD) experiments were conducted.

Baseline results were obtained using a standard whole-word left-to-right HMM with three states per model. Since

TEMM is tied to one distribution estimation per feature, a single-state single Gaussian HMM was also computed for direct comparison. All HMM models were trained by incremental mixture splitting. The number of components per mixture was optimized for best performance. All references to the number of states in an HMM refer to emitting states only. In order to investigate the influence of different features on the recognition results, and their suitability for the different models, two distinct experiment conditions were set up, each using a different MFCC feature to represent the data. Condition 1 used C0 and condition 2 used C2 as features. The results are shown in Tables 1 and 2:

Table 1. Recognition results using the C0 feature.

Model	FER
SD: TEMM (p=2)	33.15 %
SD: HMM 1 State (single Gaussian)	40.04 %
SD: HMM 3 States (single Gaussian)	28.81 %
SD: HMM 3 States (GMM 2)	22.99 %
SI: TEMM (p=1)	40.63 %
SI: HMM 1 State (single Gaussian)	46.38 %
SI: HMM 3 States (single Gaussian)	40.59 %
SI: HMM 3 States (GMM 120)	37.20 %

Table 2. Recognition results using the C2 feature.

Model	FER
SD: TEMM (p=2)	32.25 %
SD: HMM 1 State (single Gaussian)	32.61 %
SD: HMM 3 States (single Gaussian)	36.26 %
SD: HMM 3 States (GMM 60)	36.20 %
SI: TEMM (p=1)	33.34 %
SI: HMM 1 State (single Gaussian)	72.31 %
SI: HMM 3 States (single Gaussian)	69.28 %
SI: HMM 3 States (GMM 120)	61.98 %

The recognition results are rather interesting. HMMs were able to perform significantly better than TEMM when using the C0 feature. It can be argued that this may be due to the fact that C0 models overall energy in the signal. As such, neither model has much opportunity to retain fine details of the speech signal. Hence TEMM is unable to use such information to its advantage. This interpretation is supported by the recognition results using C2. Here, TEMM outperforms the 3-state HMM in the SD condition as well as in the SI condition (the difference is statistically significant only in the SI condition).

A closer analysis of the results showed that HMM output for all test conditions tended to remain in one model for a relatively long time, and hence gave rise to rather smooth recognition results that seldom changed model within a test utterance. This led to recognised words with long, (in this case) more realistic durations, and often to single-word recognition of the utterance, thereby allowing the HMM an unfair advantage to use more information on which to base its decision. This was the same even for single-Gaussian one-state HMM models. A further investigation of the model's parameters showed that the HMM's transition probabilities favoured self-transitions (i.e. transitions into the same state instead of the next), and are thus the most probable cause for this output smoothing. TEMM currently has no such smoothing mechanism, and it was observed that the

recognized class often changed from one frame to another. This is due to the fact that the classification decision in the current TEMM architecture is based solely on the similarity of the input data to its predicted feature values.

4. Conclusion

A new episodic trace model (TEMM) has been introduced that incorporates temporal sequence information. TEMM has shown some promising results in direct comparison with a conventional HMM-based classifier on data extracted from the TI-ALPHA corpus. Surprisingly, as the results on C2 features in the SI condition indicate, it seems that TEMM may be more robust to non-matching training/test data than HMMs. However, HMMs seem to have an advantage arising from the use of transition probabilities which favour self-transitions thereby giving rise to smoother recognition results. Currently, there is no equivalent constraint in TEMM, and indeed its output is rather inconsistent. Current research is investigating this particular behaviour. A further point that is receiving attention is the identification of the best feature representation to use with TEMM. It is hypothesised that once these two factors have been addressed satisfactorily, TEMM will be able to consistently outperform HMMs in a range of automatic speech recognition tasks.

5. References

- [1] Lippmann, R., "Speech Recognition by Machines and Humans", J. Speech Communication, 22, 1-15, Elsevier, 1997.
- [2] Moore R. K., "A comparison of the Data Requirements of Automatic Speech Recognition Systems and Human Listeners", Proc. Eurospeech, 2582-2584, 2003.
- [3] Moore, R. K. and Cutler, A., "Constraints on Theories of Human vs. Machine Recognition of Speech", Proc. SPRAAC Workshop on Human Speech Recognition as Pattern Classification, Max-Planck-Institute for Psycholinguistics, Nijmegen, 2001.
- [4] Bybee, J., Phonology and Language Use, Cambridge University Press, 2001.
- [5] Tulving, E., "Episodic Memory: from Mind to Brain", Annu. Rev. Psychol. 53, 1-25, 2002.
- [6] Luce, P. A. and Lyons, E. A., "Specificity of Memory Representations for Spoken Words", Memory and Cognition, 26(4): 708-715, 1998.
- [7] De Wachter, M., Demuynck, K., van Compernelle, D., Wambacq, P., 2003. Data Driven Example Based Continuous Speech Recognition. Proc. Eurospeech, 1133-1136.
- [8] Hawkins, S., and Smith, R., "Polysp: a polysystemic, phonetically-rich approach to speech understanding". Italian Journal of Linguistics - Rivista di Linguistica, 2001.
- [9] Maier, V., Moore, R. K., "An Investigation into a Simulation of Episodic Memory for Automatic Speech Recognition", Proc. Interspeech, 2005.
- [10] Hintzman, D. L., "Schema-Abstraction in a Multiple-Trace Memory Model", Psychological Review, 93: 411-427, 1986.
- [11] Hawkins J., Blakeslee S., On Intelligence, Henry Holt, New York, 2004.
- [12] Liberman et al, TI46-Word, LDC Catalog No. LDC 93S9, 1993.
- [13] Grossberg S, "Resonant neural dynamics of speech perception", Journal of Phonetics, 31: 423-445, 2003.