Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D3.2 Report focussing on the results of the initial ASR experiments comparing episodic and semantic long term memory

Due date of deliverable: 2008-12-22
Actual submission date: 2008-11-15

Start date of project: 2006-12-01 Duration: 36 Months

Organisation name of lead contractor for this deliverable: Speech and Hearing Research Group, University of Sheffield

Revision: 0.1

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

VERSION DETAILS	
Version:	1.0
Date:	22 December 2008
Status:	Final

CONTRIBUTOR(S) to DELIVERABLE	
Partner	***Name***
USFD	Mark Elshaw, Viktoria Maier, Roger K Moore Guillaume Aimetti
Centre for Language and Speech Technology, Nijmegen	Louis ten Bosch, Michael Klein
Center for Processing Speech and Images, Katholieke Universiteit Leuven	Hugo Van hamme
Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology	Okko Rasanen

DOCUMENT HISTORY			
Version	***Date***	***Responsible***	***Description***
01	10 Nov 2008	Mark Elshaw	Develop first draft of memory architecture including contributions from project partners
02	21 Nov 2008	Mark Elshaw	Further extension of report and editing of report
03	26 Nov 2008	Roger Moore	Check for content and grammer
04	26 Nov 2008	Mark Elshaw	Made changes identified by Roger Moore
04	9 Dec 2008	Mark Elshaw	Make changes identified by reviewer
05	11 Dec 2008	Mark Elshaw	Make changes identified by reviewer and incorporate additional text
05	17 Dec 2008	Okko Rasanen	Checked model description
05	17 Dec 2008	Hugo Van hamme	Checked model description

DELIVERABLE REVIEW			
Version	***Date***	***Reviewed by***	***Conclusion****
03kd	8 Dec 2008	Kris Demuynck	See track-changes
04MS	11 Dec 2008	Michael Klein	See track-changes and materials

TABLE OF CONTENTS

1 Introduction

This deliverable D3.2 report considers the results of the initial automatic speech recognition experiments to compare episodic and semantic long-term memory. An ACORNS memory architecture has been devised towards the aim of developing an agent that can comprehend language and communicate based on sensory inputs in an emergent manner. The ACORNS memory architecture was developed based on inspiration and constrains from the human memory and cerebral cortex described in deliverable D3.1. Although in this report there is a focus on long-term memory in the ACORNS memory architecture, the structure of the architecture relies on an interaction between working memory and long-term memory. The ACORNS memory architecture is described in terms of the processes involved and data stores. The long-term memory representations are the stored weight structures and working memory is the activations. The architecture is also based on the hierarchical memory-prediction model which allows the agent to develop communication skills in an emergent manner.

In this report Section 2 examines the current ACORNS memory architecture; Section 3 briefly defines what is meant by long-term memory; Section 4 considers examples of semantic long-term memory models developed with in the ACORNS memory architecture; Section 5 presents an implementation of the episodic long-term memory model associated with the memory architecture; and Section 6 discusses some of the outcomes associated with these two sets of memory models.

2 Overview of the ACORNS memory architecture

The ACORNS project investigates the feasibility of the memory-prediction framework (Hawkins and Blakeslee 2004) as a basis for understanding language acquisition and communication. The memory-prediction framework is appealing, mainly because it is based on solid neuro-physiological evidence (Mountcastle 1978). Equally importantly, is the extensive literature on memory processing in psychological research (Baddeley 1992) that does not necessarily map one-to-one to the structure suggested by the memory-prediction framework. Therefore, much time and effort has been spent during the first two years of the project to design a memory architecture that at once reflects the results of decades of psychological research and the basic tenets of the memory-prediction framework.

In this memory architecture (Figure 1) the working memory unit receives an attention-gated version of the current audio and semantic (visual) feature input (from Echoic and Iconic memory), which produces a working memory a representation in the form of activations of the input. This representation is produced through learned weights that are stored in the long-term memory in the form of semantic and episodic memory (activations are in short-term memory and weights in long-term memory). These weights are updated/learned based on the activations that are produced in the working memory so new examples of audio and visual samples can be incorporated into long-term memory as well as better representations of previously stored weights. Attention mechanisms whose weights are also stored in semantic long-term memory are used to control the updating of the learned long-term memory weights for other automatic speech recognition applications. By combining the weights and the current activation patterns the devised models can perform activities such as automatic speech retrieval, representation and prediction. With semantic features in the ACORNS architecture, the complete scene can be presented to the iconic memory. Episodic and semantic long-term memories are produced and updated by changing the weight structures stored in long-term memory.

It is possible to map the echoic and iconic memory in Figure 1 onto the lowest level of the memory prediction architecture, because neither model makes hard claims with respect to the neural encoding and representation of the sensory signals. The processing going on in the working memory and stored weights in long-term memory in Figure 1 fit onto the connections that are formed and the information that flows in the

higher levels of the structure memory-prediction. Therefore, we take it that an architecture such as depicted in Figure 1 can implement the basic operations in a memory-prediction framework.

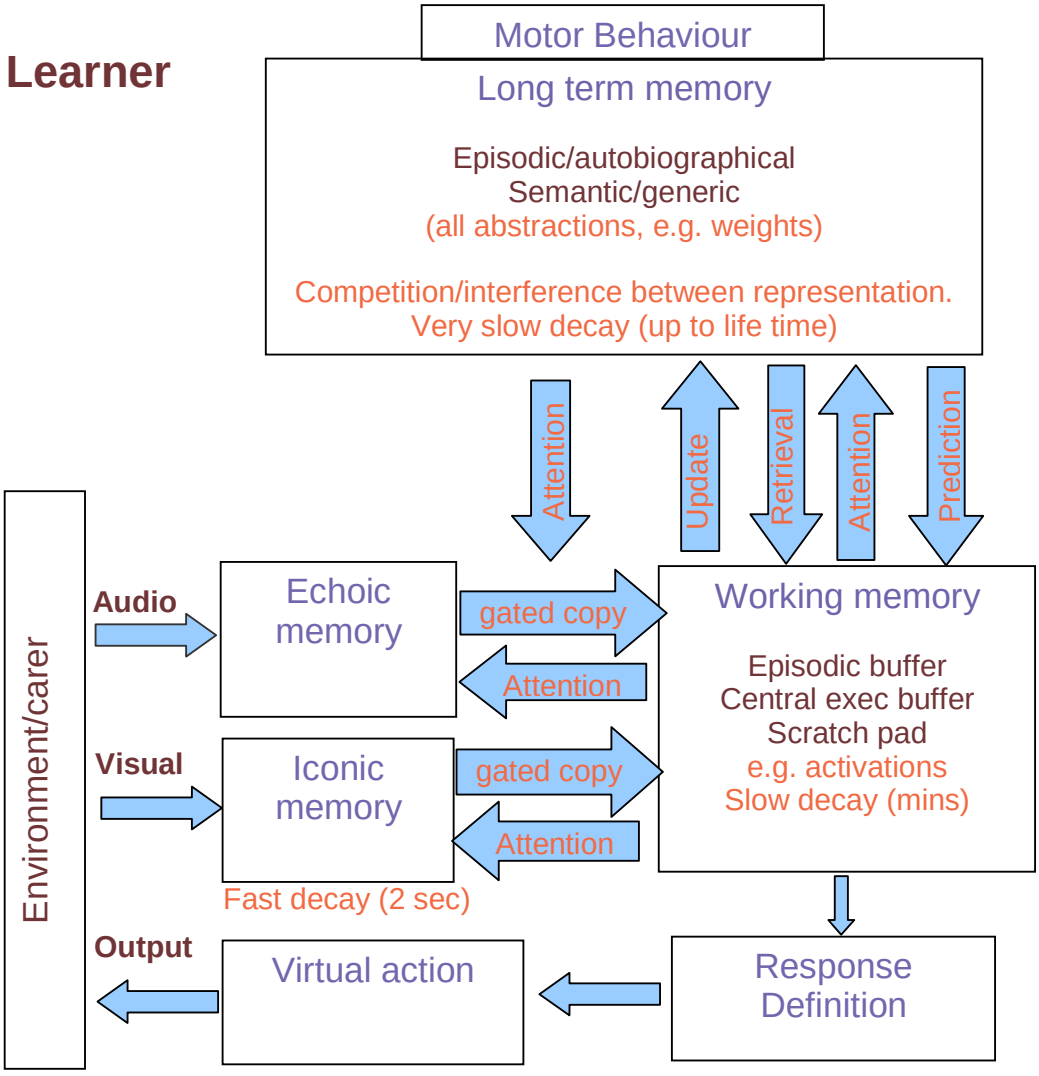


Figure 1 The ACORN memory architecture. Boxes and arrows refer to data structures and processes, respectively.

Figure 2 shows a representation of the hierarchical structure of the ACORNS memory architecture. The audio and semantic (visual) feature inputs are combined with the learned weights (semantic and episodic long-term memory) to produce representations of speech at different levels of abstraction. A_1 represents the activations (working memory) that are learned by creating and updating long-term memory W_1 based on only receiving audio input and as such are the representation of speech units. The A_2 region provides representations of words by combining semantic (visual) features of words A_s with the phone representation previous produced by the A_1 region, using learned weights W_2. The representation is based on learned weights in the upper layer of the model. The symbols A_s in Figure 2 represent the semantic activations and

are the result of processing the visual input. The A_3 region provides activation patterns that represent an utterance, based on learned weights W_3 of semantic and episodic long-term memory by combining the activations for the words A_2.

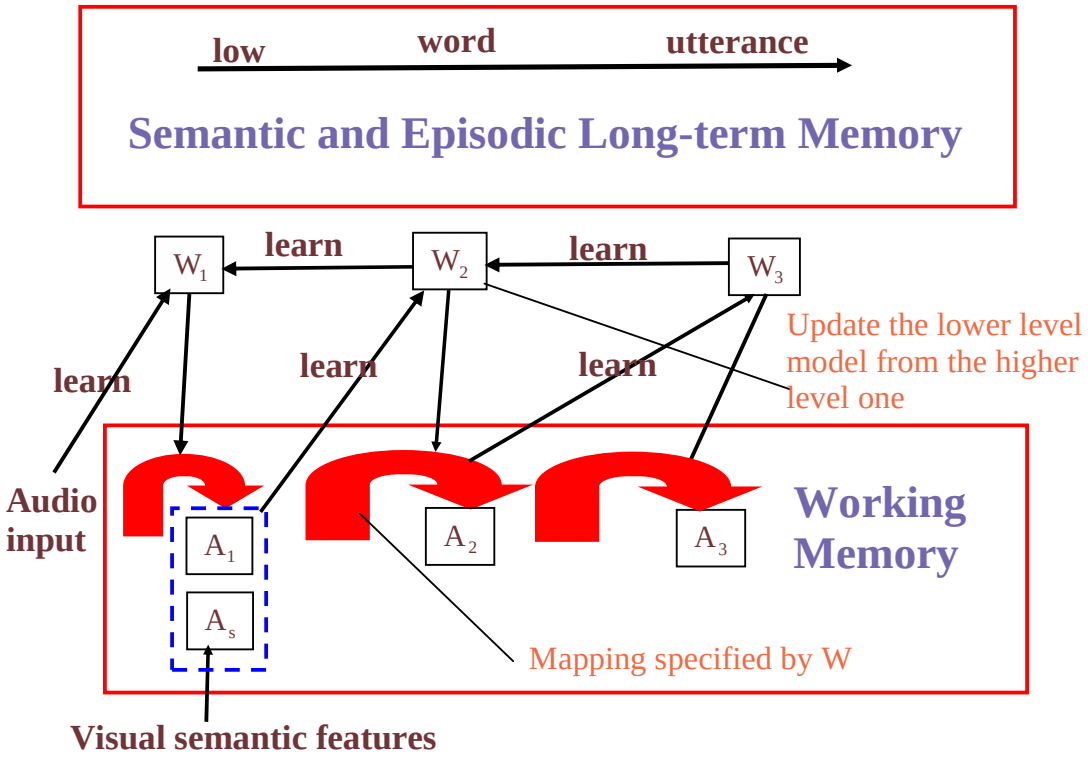


Figure 2 Hierarchical organisation of the ACORNS memory architecture.

The incorporation of complex feature based semantic (visual) features to achieve grounding into the ACORNS memory architecture works towards the development of computational models of an agent that learns to communicate. Pulvermüller (Pulvermüller 1999, Pulvermüller 2002, Pulvermüller 2003) states that semantic (visual) features play an important role in the representation of words in the cerebral cortex. He argues it is important to relate the neurons that represent the word form with those neurons associated with perception and actions that reflect the semantic information on a word. When considering content words, the semantic factors that influence the cell assemblies come from various modalities and include the complexity of activity performed, facial expression or sound, the type and number of muscles involved, the colour of the stimulus, the object complexity, movement involved, the tool used and whether the person can see herself doing this activity (Pulvermüller 2003). For objects the semantic features represented by cell assemblies typically relate to their colour, smell or shape.

3 Long-term memory

Given that this deliverable D3.2 report looks at some of the current ACORNS models from a memory architecture point-of-view with a focus on the semantic and episodic long-term memory, we will start with a brief description of what is meant by long-term memory. Long-term memory is said to contain those memories that remain for more than just a few minutes. It includes the memory of recent facts as well as the memory of older facts. The robustness of the memory is thought to be also dependent on rehearsal. While the memory of recent facts can be quite fragile, the memory of older facts is usually quite robust. The long-

term memory is usually thought of as being divided into two types of memory - explicit and implicit. Explicit memory (also called declarative memory) is that type of memory that one is aware of and can name. Implicit memory is for example motor memories, i.e. those memories that one uses to perform a certain action such as riding a bike, once it is learned. Explicit memory can further be divided into two sub categories: Episodic memory and Semantic memory.

Semantic long-term memory is thought of as a declarative system that is responsible for general factual knowledge of the world in an abstract and relational form (Baddeley 2002, Neath and Surprenant 2002). It contrasts directly with episodic memory through its lack of association with a specific moment in an individual's personal past, and lacking subjective experience (Eysenck and Keane 2005). Semantic memory is independent of the context (in terms of time and place) in which it was acquired. It is formed by a lifetime of information. It can be regarded as a form of reference material, which includes rules and concepts that let us construct a mental representation of the world without any immediate perceptions.

Episodic memory was proposed by Tulving as recently as 1972 (Tulving 1972). In contrast to semantic memory, episodic memory is thought to retain particular personal experiences at particular times and places (Tulving 1972, Eysenck and Keane 2005). The most distinctive feature of episodic memory is that not only are events memorised, but also the contexts in which they occurred. What enters into episodic memory has been shown to be dependent on attention mechanisms (Gleitman et al. 1999). From neuroscience studies the frontal and medial temporal areas are found to have a role in representing and retrieving episodic memory and the hippocampal system in representing and encoding the spatial-location memories (Hayes et al. 2004).

4 Semantic long-term memory in ACORNS memory architecture

In this section of the report we present various models that have been developed in the ACORNS project incorporate a semantic long-term memory component in the ACORNS memory architecture (see Figure 1). The example semantic long-term memory based models are related to attention based activities, keyword recognition, and word and phoneme representation and recognition.

4.1 Attention

As can be seen from Figure 1, a feature of the memory model is the use of attention mechanism at various points within the model, which makes use of learned structures that are stored in semantic long-term memory. Most commonly, attention is used to refer to 'selectivity of processing'. For Pugh et al. (1996) attention should include the capacity to switch focus from one element to another, must be maintained over a period of time and be limited in the number of elements that can be focused on at any time.

James (1890) defined attention as:

"Everyone knows what attention is. It is the taking possession of the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought. Focalisation, concentration, of consciousness are of its essence."

Various neuroscience studies have considered aspects of attention such as sustained attention, selective attention and decision and action control (Pugh et al. 1996). For sustained attention activation is found in the superior parietal area and the prefrontal areas of the right hemisphere when subjects looked for small changes in stimulus (Pugh at al. 1996, Pardo et al. 1991). Turning to selective attention the superior parietal lobule is identified to be related to removing focus from one element to the next, the superior colliculus to the movement to a new focus and the pulvinar to filtering out stimulus that are not of interest (Posner and

Presti 1987, Pugh et al. 1996). Below is an examination of examples of the attention mechanisms with focus on semantic long-term memory that have been developed within the memory architecture.

4.2 Selective attention semantic long-term model

The aim of the first semantic long-term memory based model described here is to examine whether special attentional focus to keywords in utterances facilitates word learning and recognition. The word-learning algorithm (Figure 3) uses a modified concept matrix approach (weights like structures that are stored in semantic long-term memory) to track transitional probabilities of vector quantized speech, quantization being provided by the clustering algorithm developed in workpackage 2. 2000 utterances, from a single Finnish female speaker in the corpus collect in period 1, were used as the test material.

The cornerstones of the algorithm rely on the so-called concept matrices which form an inner representation (semantic long-term memory) that associatively combines information from the auditory stream (cluster, or "phone", sequences) with other visual semantic input. In this case the visual input is simplified to being the keyword tag associated with each sentence. In other words, the algorithm concentrates solely on the modelling of auditory stimuli and the interaction between working and semantic long-term memory related processing and makes an approximation that non-auditory modality signals are provided by external processing modules instead of intertwined multimodal processing streams already at the signal level. The word-learning algorithm takes a multimodal tag and two parallel cluster sequences corresponding to a single utterance as the input. The first sequence S_1 contains a discrete sequence of segmental cluster indices of segment onsets (spectral representations created from the first 40% of segment durations) while the second sequence S_2 contains cluster indices for the remaining 60% of segment durations in a similar manner. The multimodal tag *t* represents one of the ten possible keywords in the material as an integer.

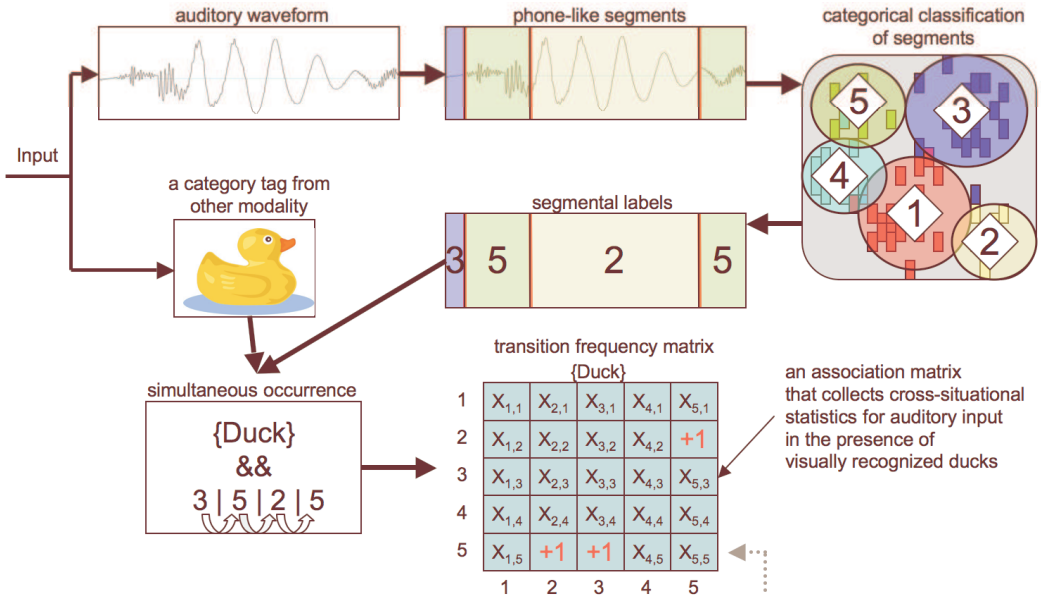


Figure 3 A schematic representation of the word-learning algorithm. Cross-situational statistics are collected for phone-like-segment transitions during presence of multimodal tags.

Whenever a new utterance is introduced, the algorithm goes through all indices in S_1 and adds all pairs of two subsequent symbols as transitions to a transition frequency matrix $\mathbf{M}_{f,t,1}$ defined by the tag associated

with the utterance. This is repeated for S_2 $\mathbf{M}_{f,t,2}$. The stochastic (normalized transition probability $\sum p(i)$ =1 from any symbol) matrices $\mathbf{M}_{p,t,1}$ and $\mathbf{M}_{p,t,2}$ are then created from both frequency matrices. These matrices now contain probability distributions for all possible symbols [x+1] from all possible symbols [x] for each concept (keyword). Similarly to transitions of subsequent symbol pairs, matrices $\mathbf{M}_{f,t,1,3}, \mathbf{M}_{f,t,2,3}, \mathbf{M}_{p,t,1,3}, \mathbf{M}_{p,t,2,3}$ are created for transitions from [x] to [x+2]. If the tag *t* has been never seen before, a new set of concept matrices (long-term memory weights) is created for both S_1 and S_2 for both [x]→[x+1] and [x]→[x+2] transitions. Otherwise, the existing frequency matrices and corresponding probability matrices in the semantic long-term memory are updated.

To recognize a word from a previously unheard utterance, sequences S_1 and S_2 corresponding to the incoming utterance are utilized. Both sequences are windowed simultaneously with a varying sized window to obtain sub-sequences B1 and B2 for all possible window locations and for all pre-defined window sizes. These sequences are then used to "activate" concept matrices, that is, [x]□[x+1] and [x]□[x+2] transitions in the sub-sequences. This process is then repeated for all concept matrices to provide a cumulative probability sum for each sub-sequence for each concept matrix. In the case of zero probability for a transition (no such transition has ever occurred before in the presence of some corresponding tag), a small penalty to the cumulative sum is introduced. The most probable combination of a sub-sequence and a concept matrix produces a word hypothesis (i.e., which concept is being activated most) and a hypothesis for the temporal location for the word in speech signal.

The procedure is to emphasize segmental transitions occurring during keywords with different scaling compared to the surrounding carrier sentences. Two basic attentional learning situations implemented in the current version of the algorithm are tested. In *the first situation* the learner has absolutely no feedback from the external world except for input consisting of spoken utterances, corresponding tags, and temporal locations of the keywords. This simulates a situation where the learner gets accurate information about the keyword location from some other process, e.g., by processing of the prosody of the input. In *the second situation*, which is a so-called reinforced learning environment, the learning agent obtains feedback for its decisions from the caretaker. After the learner made a hypothesis about which concept/keyword/tag is being referred to by the spoken utterance, the caretaker signals whether the hypothesis is correct or incorrect. In the case of a correct answer, only the sub-section of the sequence that yields the best match for a correct concept matrix is added to the matrix (note that external temporal focus is not influential in this method, but more like a self-driven focus for important content). On the other hand, if the answer is incorrect, all transitions in the entire sequence are added to the concept matrix defined by the tag in a similar manner as in the non-reinforced case using the temporal focus. This should lead to situations where the contents of concept matrices become increasingly selective to the contents of words instead of modelling entire utterances.

Four learning experiments were conducted for both the reinforced and non-reinforced cases in which the scalar value for a keyword and for the surrounding carrier sentence are varied independently. Three extreme and one intermediate situation are considered: (i) no differential scalar value at all, (ii) all scalar value on carrier sentences, (iii) all scalar value on keywords, and (iv) 100% scalar value on keywords and 50% scalar value on carriers. It turned out that the differences between overall accuracies of reinforced versus non-reinforced learning, and non-weighted versus the weighted cases are not large. Only when scaling the keyword downwards to zero without reinforced learning did poorer recognition rates occur, as is expected.

In the non-reinforced case an increase in the weight of a keyword increases the recognition confidence notably due to diminished statistical saliency of carrier sentences that are shared between several concepts. Similar trends can be detected in the reinforced case as well, although the change is much smaller in this case. However, the suppression of the 'value' of carrier sentence information, when compared to the keyword case, seemed to impair the keyword recognition rate in both cases, inferring that that the surrounding context facilitates keyword recognition. This is a reasonable outcome: at least with such a small vocabulary and simple grammatical structure, the statistical properties of carrier sentences are probably

biased to point at specific keywords. the surrounding context facilitates keyword recognition. This is a reasonable outcome: at least with such a small vocabulary and simple grammatical structure, the statistical properties of carrier sentences are probably biased to point at specific keywords.

Figure 4 shows local and cumulative recognition rates for scaled and non-scaled experiments with non-reinforced learning (top: first 300 utterances in detail, bottom: the entire material in a longer evaluation window). As can be seen, the difference in recognition accuracy is not significant, although focused keywords obtain a small overhead in the beginning. With large amounts of training material cumulative accuracies converge to similar accuracies. An interesting detail can be seen at approximately 700 and 1700 utterances, where focused keyword accuracy dips clearly due to some local variation within the speech material while non-focused accuracy remains relatively stable.

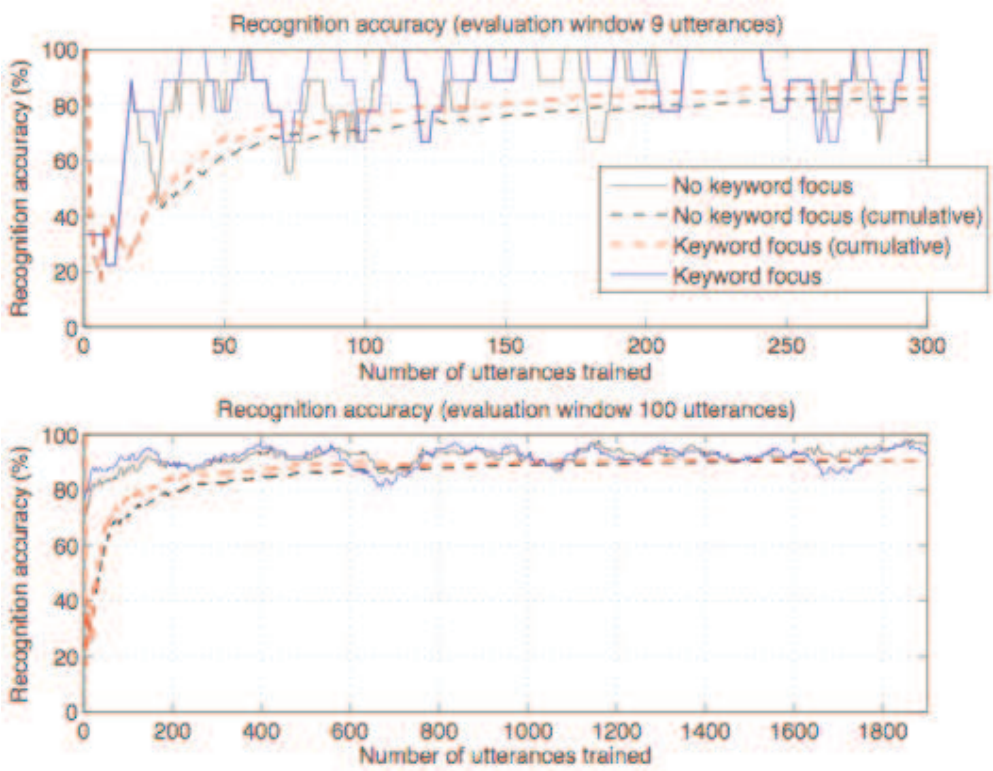


Figure 4 Word recognition accuracy, focused ($w_{keyword}$ = 1, $w_{carrier}$ = 0.5) versus non-focused attention.

When the keyword scaling is set to zero, the learner cannot detect directly any cross situational statistics (see Smith & Yu, 2008) between occurrences of similar auditory word forms and a specific concept. In a non-reinforced situation the learner can only make a guess between the concepts sharing the same characteristic carrier sentences since the learner does not 'hear' the keyword at all. In the reinforced learning case the learner can accidentally stumble upon cues for auditory word form. What happens in a reinforced non-keyword focused situation is that the carrier sentences enable correct recognition every now and then similar to the non-reinforced case, but the recognition window can overlap partially with the keyword specific segments despite that a small penalty is introduced for zero probability transitions in the recognition window. These types of transitions are then summed to the concept matrix, making a small but increasingly significant difference to other concepts sharing the same carrier sentence. This gradually increases the overlap between the recognition window and the segments associated with the keyword in the future input, increasing the selectivity and recognition accuracy near non-focused or keyword-focused learning situations on the long run.

4.3 Attention-gated mechanism for speech detection

The next attention based model in the ACORNS memory architecture that makes use of semantic long-term memory controls the type of input data that is passed from the echoic memory into the working memory. In the architecture the auditory signal in the form of a waveform is split into time slices using a moving window and introduced into the attention-gating architecture one at a time so it can learn to detect speech (Figure 5). The training samples for the attention-gated reinforcement model are auditory samples of non-speech (crowd noise) and speech (first female speaker from English the ACORNS database). The test data includes (i) different samples take from the same crowd scenes used for training; (ii) new non-speech samples from scenarios not used in training; (iii) different recordings of the training utterances by the first female speaker, (iv) the first female speaker saying different utterances from those used in the train samples; and (v) a second female speaker saying the original utterances.

The model uses an adaption of the actor-critical form of reinforcement learning (Sutton and Barto 1988) to learn and update the critic's weights w^c and the actor weights w^s (stored as semantic long-term memory) to determine if an auditory input section is speech. A sample is selected and moving window auditory section inputs are created along the full length of the auditory sample and introduced one section at a time to train this neural network model.

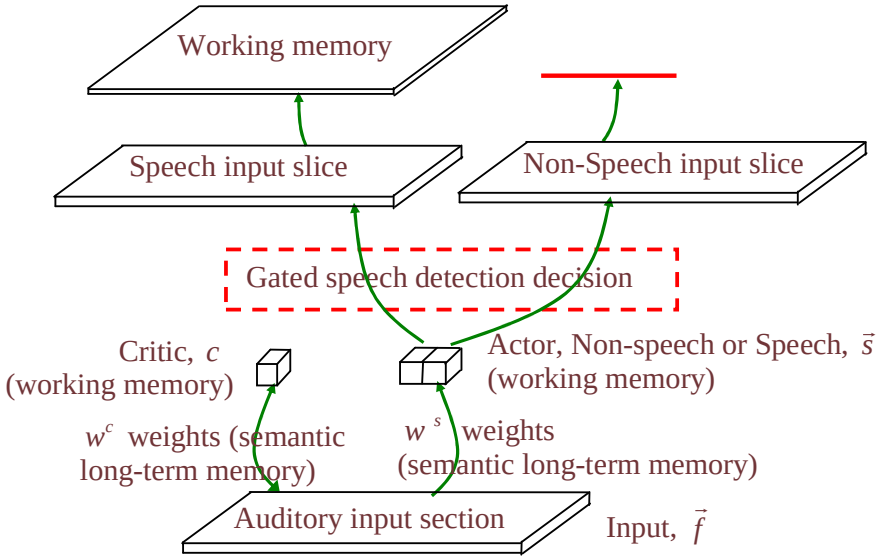


Figure 5 Reinforcement based attention-gated model to discriminate between speech and non-speech.

Until the end of the specific auditory sample the following process is repeated to train the model. The reinforcement model uses the auditory input section $\vec{f}$ and critic weights w^c (stored as semantic long-term memory) to create the activation of the critic unit c (working memory activations):

$$c = \sum w_j^{\,c} \cdot f_j \qquad (1)$$

where j is the index for the data points in the input section (1 to 30)

Once this is performed then the probability that the actor unit i is set active for the input section is calculated:

$$P(s_i=1)=\frac{a_i}{\sum a_{i'}} \text{ with } a_i=\sum w_j^s \cdot f_j \quad (2)$$

The next input section of the sample $\vec{f}'$ is then selected and the critic activation c' (working memory activation) is calculated:

$$c'=\sum w_j{}^c \cdot f_j' \quad (3)$$

A difference value between the critic value for the previous input auditory slice c and the current one c' that is determined according to:

$$\delta=(c-(c'-\gamma)) \quad (4)$$

where γ is the discount rate at 0.9.

The immediate reward IR that is used to change the weights after each auditory input section is determined using the difference value δ and the input section values f' using the Equation below. Hence this gated model makes use of the fact that it is possible to identify if the input section is correctly detected and allocates the immediate reward value:

$$IR=\begin{cases}\delta \cdot f'_j > 0 \text{ else } 0, \text{if the auditory section correctly detected}\\ \\ \text{-}2(abs(\delta \cdot f'_j)), \text{if the auditory section wrongly detected}\end{cases} \quad (5)$$

For each input section the critic weights (semantic long-term memory representation) are updated with η being the learning rate at 0.0000035:

$$w^c=w^c+(IR\cdot\eta) \quad (6)$$

The actor weights (semantic long-term memory) are updated based on the immediate reward for the actor unit *i* with the highest probability:

$$w_i{}^s=w_i{}^s+(IR\cdot s_i\cdot\eta) \quad (7)$$

The delay reward values DR_k are used to update weights at the end of the full non-speech and speech sample and are determined using the equation below. The DR value reflects the degree based on the difference value and input section values $\vec{f}'$ that the model correctly identifies the sample, and is defined as follow:

$$DR_k=\begin{cases}sum(\delta \cdot f'_j, \text{ for all auditory slices correctly detected for the full sample}\\ \\ sum(\text{-}2(abs(\delta \cdot f'_j))), \text{for all slices wrongly detected for the full sample}\end{cases} \quad (8)$$

where k is the index for delay reward values (1 – positive, 2 – negative)

At the end of the auditory sample the delayed reward values are used to update the critic weights (semantic long-term memory):

$$w_j^c = w_j^c + (DR_k \cdot \eta) \quad (9)$$

The actor weights (semantic long-term memory) at the end of the sample are also updated with the delay reward values based on the actor unit i with the highest probability at the time when the delay reward values are calculated and whether the input sections are detected correctly:

$$w_{ij}^s = w_{ij}^s + (DR_k \cdot s_i \cdot \eta) \quad (10)$$

The auditory waveform samples for both the training and test data are converted into auditory input slices using the logarithmic mel-spectrum approach. Training is performed using each full sample of speech and non-speech in the training set selected randomly and presented over 17 epochs to allow the reinforcement attention gating system to learn to differentiate between speech and non-speech.

The reinforcement based attention-gated mechanism is tested using speech and non-speech samples made up of auditory input section values one at a time along the full sample. The reinforcement gating system is able to detect correctly 93% of the new non-speech auditory input slices from the crowd scenes that are used to train the network. When considering the detection of speech the performance is 80% for the new versions of the same utterances and new utterances by the training female speaker. The incorrect detection of speech slice inputs by the gating network is due in part to periods in the speech samples where there are no speech sounds for instance between words. However, as a frame-based approach is used and given speech typically crosses multiple frames, it will be possible to use this characteristic to identify wrongly detected speech frames. This is possible in that when a frame of speech is correctly detected and then a frame representing a signal within say 200ms is also detected as speech the frames between these can also be assumed to be speech. When considering the performance on speech from a female speaker who is not used in training the reduction in performance is not that significant (76%), which indicates that the system is not specific to the training person and can be used for other female speakers.

4.4 The phoneme and word recognition within the memory architecture

An application of the memory architecture for identifying words and phones has been devised using the non-negative matrix factorization (NMF) approach. The conceptual representation related to the memory architecture is shown in Figure 6. NMF has been used in the ACORNS project to discovered pattern in an utterance. In the present NMF bottom-up approach, recognition is driven by the co-occurrence of acoustic events. In general, these events are the occurrence of specific patterns in the time-frequency plane. This leads to a vectorial representation of high but fixed dimension called ‘Histogram of Acoustic Co-occurrence’ (HAC). In HAC, the probability of acoustic events is accumulated over a graph (representing speech input). The HAC-model (histogram of acoustic co-occurrence) with its associated learning algorithm based on NMF is able to discover recurring acoustic patterns in speech both without supervision and with weak supervision. By extending this model using ‘time Histogram of Acoustic Co-occurrence’ (tHAC) it is possible to estimate at which specific times the acoustic patterns have occurred. A second option for localizing the patterns is to use a sliding window. If a pattern is detected at the current position, we know it must be within the window boundaries. By combining both ideas, an even finer time estimate is obtained. With this approach, continuous speech recognition on the TIDIGITS database was successful. There is now a bottom-up

activation-based recognizer that doesn't need a search algorithm like Hidden Markov Models (HMM) (see also Deliverable D4.1).

In Figure 6 the echoic memory contains the buffer of speech for copying exemplars and eHAC and ichonic memory buffer of semantic features for copying exemplars and eHAC. The working memory representation incorporates a gated buffer of speech and semantic features and eHAC representation of activated phones and semantic features. eHAC instead of multiplying the probability of acoustic events with the time at which they occur (tHAC), multiplies the probability of acoustic events with two or more monotonous function of time, e.g. $\exp(-\alpha t)$ to allow estimating the time at which a pattern is activated. This means that the input representation maintains cells (nodes or rows in an input vector) which are activated by an acoustic event and whose activity then decays. This is cognitively very plausible. From that NMF computes activations of discovered patterns and their activation time. Hence the output becomes activated events (patterns) which decay with time. The methods described above have been validated and are operational.

The architecture will be developed further as described below. In order to develop semantic long-term memory representations of words in terms of phone-like acoustic units the NMF approach can use a hierarchical approach. Assume hundreds of VQ-based word-sized patterns were discovered with NMF: V=W*H with W the word models (one per column) in terms of VQ-label co-occurrences. Factorizing W=Y*G should result in common units across the words, like phones, in Y and the presence/absence of each of these units in each word in G. This is an off-line training process: a discovery on the structures in memory. Hence this produces V=Y*G*H. Once Y is fixed and the phones in a language are learned, only a new entry in G is needed to extend the vocabulary with one word.

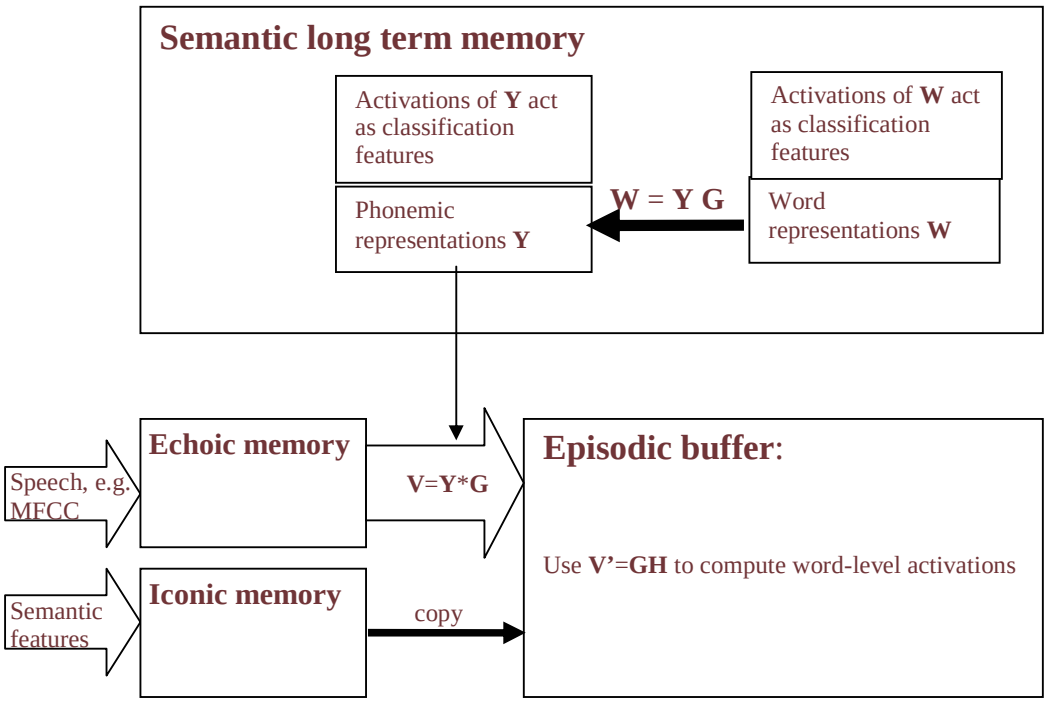


Figure 6 A conceptual representation of the NMF model for word and phone recognition related to the ACORNS memory architecture.

4.5 Semantic long-term memory models for semantic (visual) feature for word representation

In the experiments with the ACORN database collected in first period of project typically semantic (visual) information is presented to the learner in the form of rigid semantic tags, which in contrast to the noisy

speech input do not resemble real input, do not account for the variability of real world perception of objects and actions, and due not represent the ambiguities in terms of reference that language acquiring children have to face. Instead they represent a strong form of a prior knowledge that is not in accordance with the general goals of the ACORNS project. To overcome this shortcoming it was decided to use semantic features to approximate the visual input of the learner. In particular, features can be used to model (i) token variation (noise), (ii) type distance, and (iii) referential ambiguity. Token variation means that actions and objects of a category look different at different instances, either due to a change in perspective, or they might actually be different individuals. Type distance means that some semantic categories are more similar to others giving rise to confusing in the learner.

Semantic features allow the modelling of similarity and simulate phenomena like over-generalization (calling a cat "dog") that can be observed in during language acquisition. Further, referential ambiguity can be accomplished by using several object position slots. An object is defined by a set of features, depending on which categories apply to it. For example one position might be filled with the feature sets of red, furry, eats, bear, and animal, while another slot might be filled with the feature sets of round, green, apple, food. The learning system might then be exposed to an utterance such as 'The bear eats the apple' or 'The red furry animal eats the round green food'. Although the main switch from the usage of rigid tags to the use of full set of ACORNS semantic features as described in deliverable D5.4.2 for the database that combines recording made in the two periods of the project, first explorations with semantic features were already made using a features set defining the concepts of the records produced in the first period of the project. The primary issue to solve was what sort of cognitive plausible module could provide a good interface between the features and the parts of the acorns implementation which is concerned with the acquisition of the auditory representations.

The acquisition of semantic categories on the basis of these semantic (visual) features is performed using two mapping approaches: self-organising map and biased competitive layer. The weights produced by these models (see Figure 1) are stored in semantic long-term memory and activations associated with a specific semantic feature word patterns occur in working memory. A self-organising map (Kohonen 1997) is an unsupervised learning algorithm that uses an additional competitive output layer of nodes in addition to the input layer. Each unit of this output layer acquires a prototype vector. The closer the prototype vector is to the input vector the stronger it reacts to it. During learning (semantic long-term memory weights) the output vectors systematically adapt to cover the categories present in the input vectors so that the final output map captures the distance of input vectors in the topography of the output grid. Simulations are made with binary output (winner-takes-all) and probabilistic outputs.

Like a self-organising map the Biased Competitive Layer is an unsupervised learning algorithm that uses an additional competitive output layer of nodes in addition to the input layer. Also in this algorithm, each unit of this output layer acquires a prototype vector, with the closeness of the prototype vector determining the strength of its reaction. However, in contrast to the self-organising map where the distance in the output grid is supposed to represent the distance in the input space, the biased competitive layer does not use an output grid, but merely a number of output units that have no relation to each other. The algorithm uses a bias that increases if a unit is not selected to make sure that every unit is selected an adapted to the input space. A number of experiments were run to generate over-generalizations, one of the most basic behavioral findings explained by semantic features during first word acquisition. Over-generalization in this context is the application of one concept, such as [[dog]], for everything similar to a dog (like furry animals with four legs) - a behavior reported for children during language acquisition (e.g., Clark, 1973). With a 4x4 output grid (16 units for 13 concepts) the self-organising map did not lead to a unit for every concept. The reason for this can be seen in the general mechanisms of the Kohonen algorithm. It strives to map similar vectors closer to each other. Hence, it mapped similar vectors (like mama and papa) into the same unit. Choosing a higher resolution (8x8) overcame this problem, since consequently the self-organising map had more space to represent distances in input space while still using different output units for different concepts. Further, we demonstrated that the conceptual analogue of the hypernym/hyponym problem did not occur in this architecture.

Biased Competitive Layers performs as well as self-organized maps. Since this architecture did not attempt to map conceptual similarity into output space, it always uses different units for different concepts. We showed that the model develops 13 conceptual units, each of which activates the strongest when the feature set that defines the concept is presented to the model. In simulation designed to test for over-generalizations, it could be observed that the [[dog]] concept is applied to a feature set defining a cat for a number of time steps, until the concept [[cat]] is established. This result suggests that over-generalization is not necessarily caused by a representation of wrong or insufficient features, but just by the activation of the concept which is the closest match.

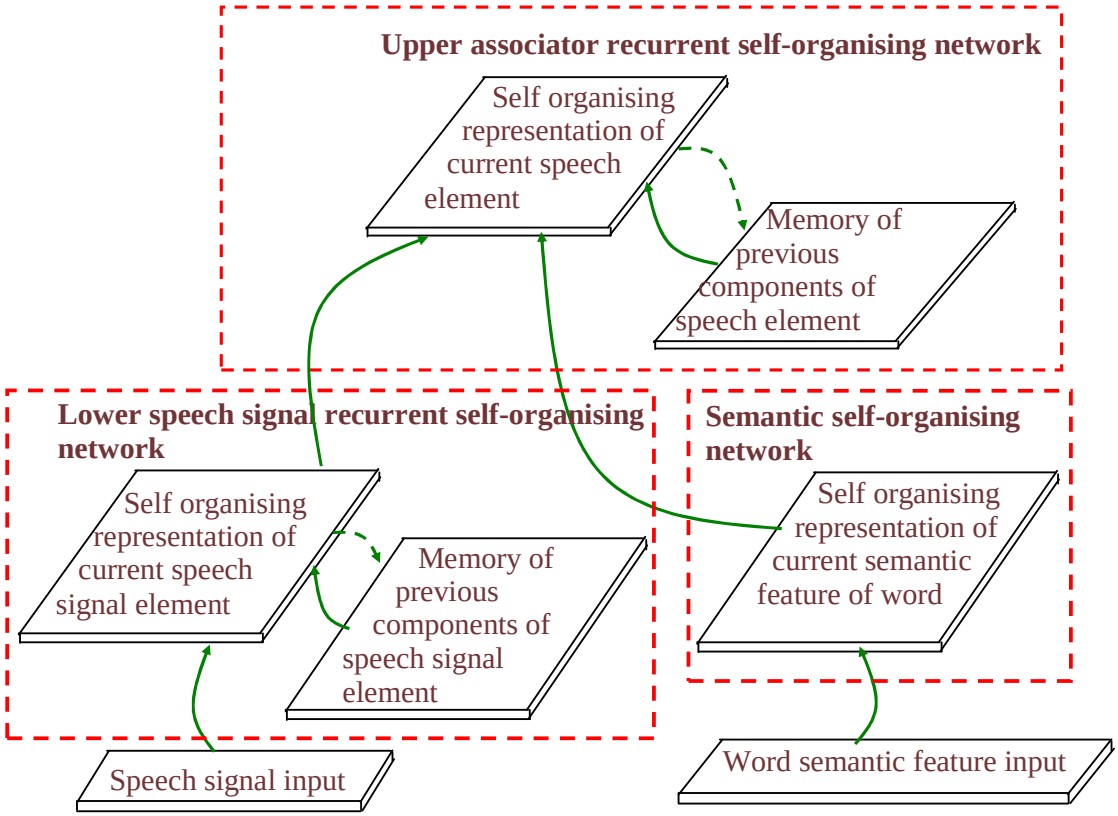


Figure 7 Representation of semantic long-term memory based recurrent self-organising memory model for emergent speech representation.

4.6 Semantic long-term memory self-organising based word representation

A further semantic long-term memory based approach within the ACORNS memory architecture uses a recurrent self-organising map model for learning an emergent representation of speech. This model combines speech signals and semantic (visual) features to develop emergent speech representation. The complete model is depicted in Figure 7. The lower speech signal recurrent self-organising model is trained using two inputs, namely the current speech window time slice representation of the speech waveform and the previous time-step activations from the recurrent self-organising network. The previous time-step activation can be seen as the abstract representation of the previous speech signal slices in working memory. Making use of finding from the semantic (visual) feature long-term memory model considered in subsection 4.5 a self-organising network is trained using the semantic (visual) features to produce an unsupervised semantic representation for specific words. It should be noted however that the semantic feature set was devised as an intermediate approach for test of concept related to the recurrent self-organising map model. In future implementations and extensions of the model the ACORNS semantic (visual) feature set (see

deliverable D5.4.2) will be used as it is now available and offer a better approach for representing the visual scene. Once training is completed for these two lower networks, the upper recurrent self-organising model is used to associate the speech signal activations with the semantic (visual) feature activations of the words overtime.

The semantic long-term memory model approach applied to emergent speech representation described here uses the basic self-organising map for semantic (visual) features representation and an adaption of the recurrent self-organising maps of Voegtlin (2002). In the lower speech signal recurrent self-organising model (Figure 8) the speech slices making up the word representations are introduced one at a time with the previous activation from the self-organising network acting as the working memory information. In the model a set of weights (semantic long-term memory model) are trained so they are associated with the current speech input slice and another set of weights are trained so they relate to the recurrent self-organising activations at previous time step. The upper associator recurrent self-organisation network structure is depicted in Figure 9. This network is trained to produce speech representations using the activations of the lower speech signal recurrent self-organising network for each speech time slice, the activation for the upper associator recurrent self-organising map at the previous time step and the activation for the appropriate word from the semantic feature self-organising network. For the full word input semantic (visual) feature representation with the semantic self-organising network and each time speech slice for the lower speech signal recurrent self-organising network and the upper associator recurrent self-organising network a best matching unit with the lowest activation on the output layer is identified.

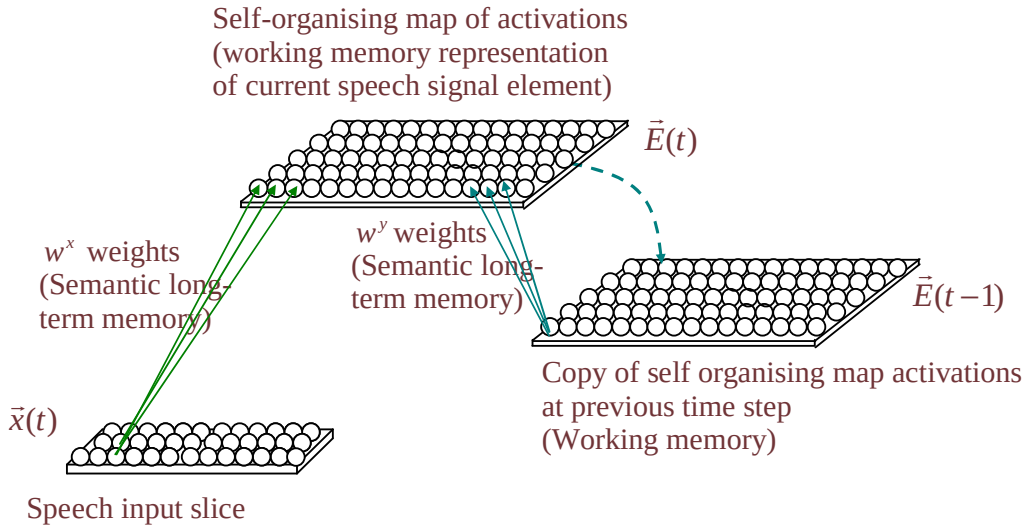


Figure 8 The representation of lower speech signal recurrent self-organising network structure.

The activations (working memory in ACORN memory architecture) in the lower speech signal recurrent self-organising network is determined using two different sets of Euclidean distance values and in the upper associator recurrent self-organising networks using three sets of Euclidean distance values. The two sets of Euclidean distance values $\vec{A}$, $\vec{B}$ for the lower speech signal recurrent network are based on the difference between the speech input slice $x(t)$ and its weights w^x and the activation for the previous time slice of the self-organisation network $E(t-1)$ and its associated weights w^y. The three Euclidean distance value sets for the upper associator recurrent network $\vec{F}$, $\vec{G}$, $\vec{H}$, are based on (i) the difference between the activation $\vec{E}(t)$ from the lower speech signal recurrent network and the weights w^E, (ii) the difference between the activations $\vec{S}(t)$ from the semantic factor self-organising network and the weights w^S; and (iii) the difference between the activations of the upper associator self-organising network at the previous time

step $\vec{J}(t-1)$ and the related weights w^J. The sets of Euclidean distance values are normalised based on the largest value in each of the sets. Two parameters α and β for the lower speech signal recurrent self-organising network and three parameters χ, δ and ε for the upper associator recurrent self-organising network are used to control the level of impact of the sets of Euclidean distance values when creating the activation for the recurrent self-organisation maps.

To determine the activation (short-term memory) for the units in the lower speech signal recurrent self-organising map $\vec{E}$ and the best matching unit, $\vec{A}$ and $\vec{B}$ are combined (Equation 12) and normalised based on largest value.

$$E_i = ((\alpha \cdot A_i) + (\beta \cdot B_i)) \quad (12)$$

To determine the activation for the units in the upper associator recurrent self-organising map $\vec{J}$ and the best matching unit with lowest activation, $\vec{F}$, $\vec{G}$ and $\vec{H}$ are combined (Equation 13) and normalised based on largest value.

$$J_i = ((\chi \cdot F_i) + (\delta \cdot G_i) + (\varepsilon \cdot H_i)) \quad (13)$$

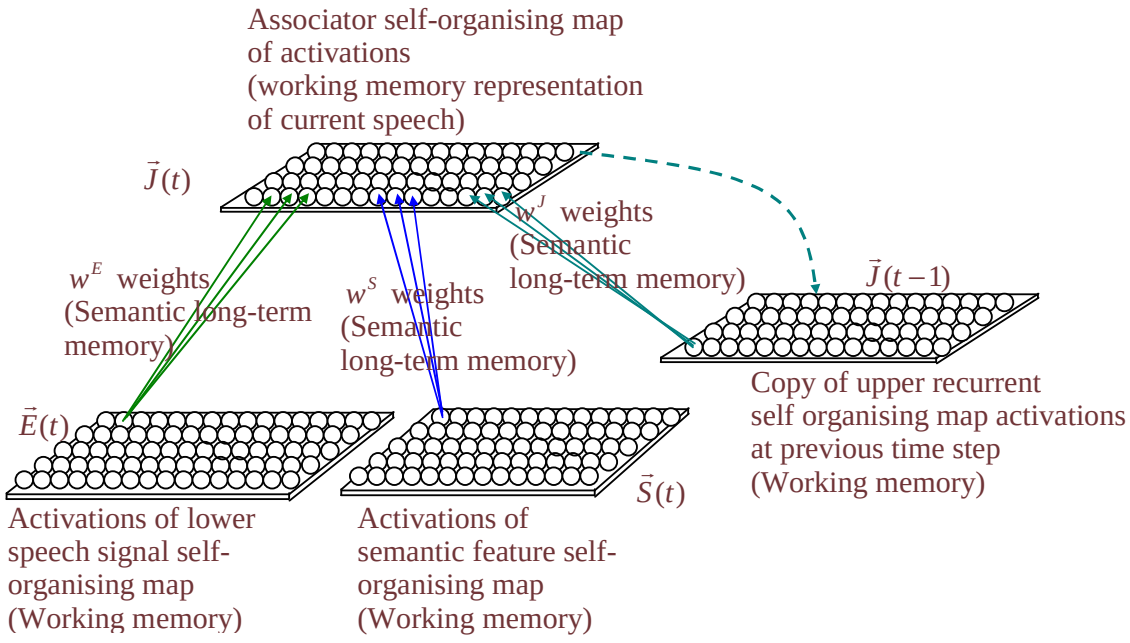


Figure 9 The representation of upper associator recurrent self-organising network structure for emergent speech representation.

The weights of the lower speech signal recurrent self-organising network (long-term memory) are updating according to Equation (14) and Equation (15). The weights of the upper associator recurrent self-organising network are also updated in a similar manner.

$$\Delta w_i^x = \gamma h_{ik}(x(t) - w_i^x) \quad (14)$$

$$\Delta w_i^y = \gamma h_{ik}(E(t-1) - w_i^y) \quad (15)$$

where the learning rate is γ (0.01) and the neighbourhood function is h_{ik}. k is the index of the best matching unit and i index of other units in self-organising network, h_{ik} reduces the greater the distance between i and k. The shape and units in the neighbourhood depends on the neighbourhood function used. The number of units in the neighbourhood usually drops gradually over time.

The training and test input for the recurrent self-organising memory model are words extracted as they appeared in 50 utterances taken from the ACORNS English database from a female speaker repeated 5 times and the test data is the same words extracted from 5 new recording by this female speaker of the 50 training utterances. 703 words are used for training and the same number for testing, with 42 distinct words, with the same words extracted when recorded in utterances containing different words. For instance, two diverse utterances made of different words from which the same word 'shoe' at different points is extracted are 'a shoe is a fashion item' and the second 'what matches this shoe'. The input includes words that could be learned by a young child including 'daddy', 'mummy', 'nappy', 'did', 'what' and 'shoe'. The semantic feature network is trained and tested using semantic (visual) feature inputs for 30 nouns and verbs used in the lower speech signal recurrent self-organising network. Semantic features were only produced for a selection of nouns and verbs as an examination for the approach and so words such as 'what', 'did' 'finally', 'today', 'matches' and 'the' were not consider. The upper associator recurrent self-organising network is trained and tested using the activations produced by the lower speech signal recurrent self-organising map for the logarithmic mel-spectrum value speech time slices and the semantic feature self-organising map for the subset of 30 nouns and verbs. The speech waveforms for the 30 distinct nouns and verbs are extracted as they appear in the same 50 recorded utterances as those used for the lower recurrent speech signal self-organising network which produced 417 words for training and the same number for testing the upper associator recurrent self-organising network.

The lower speech signal recurrent self-organising network and the semantic feature self-organising network are trained separately with the words in each training set epoch introduced in random order. For the semantic feature self-organising network the semantic features for the 30 nouns and verbs are introduced as a single representation per word. The semantic feature inputs are based on an approach similar to McClelland and Kawamoto (1986) and use various semantic features and their values. For example, for the verbs one such feature is the level of physical effort required when performing the action associated with the verb and the possible values are 'small', 'medium' or 'large'. The full set of features for the verbs and nouns and possible values are shown in Table 1 and Table 2, respectively. The values associated with semantic features that have a single option have an extent value between 0 and 1. A telephone can be seen as a piece of furniture to the extent 0.5 for example. For those features that have multiple possible options such as texture each of the three options should have a value that adds up to 1. A shoe for example has a smooth top and a rough sole and hence would have the following value for semantic feature 'texture': smooth: 0.5, rough: 0.5 and liquid: 0.0.

Table 1 Semantic features for verb meaning

Semantic Features	Responses	Semantic Features	Responses
Body Movement	Small/Medium/Large	Precise of activity	Extent (0-1)
Interaction with object	Small/Medium/Large	Communication	Extent (0-1)
Interaction with agent	Small/Medium/Large	Change to object	Small/Medium/Large
Task Complexity	Small/Medium/Large	Cognitive complexity	Small/Medium/Large
Emotion related	Extent (0-1)	Instigated activity	Extent (0-1)

Table 2 Semantic features for noun meaning

Semantic Features	Responses	Semantic Features	Responses
Worn	Extent (0-1)	Tool	Extent (0-1)
Food related	Extent (0-1)	read	Extent (0-1)
Furniture	Extent (0-1)	animate	Extent (0-1)
Inanimate	Extent (0-1)	man made	Extent (0-1)
Communication device	Extent (0-1)	Provides information	Extent (0-1)
Gender	Male/Female/Neuter	Texture	Smooth/rough/liquid
Used by	Child/Adult/Non	technology	Small/Medium/Large
Creates noise	Extent (0-1)	Size	Small/Medium/Large
Breakable	Fragile/Durable/Strong		

When considering the best matching units for the semantic feature input for the semantic self-organising network (Figure 10) in most cases each word is located in an individual unit on the network. Furthermore, similar words are also located in close regions of the self-organising map. For instance higher level cognitive function such as 'like' and 'see' are located in the top left of the network and words associated with humans such as 'daddy', 'mummy', 'Ewan' and 'baby' are found in lower left hand corner. Also the written communication media 'newspaper' and 'book' are located close together as are the action verbs 'comes' and 'join'.

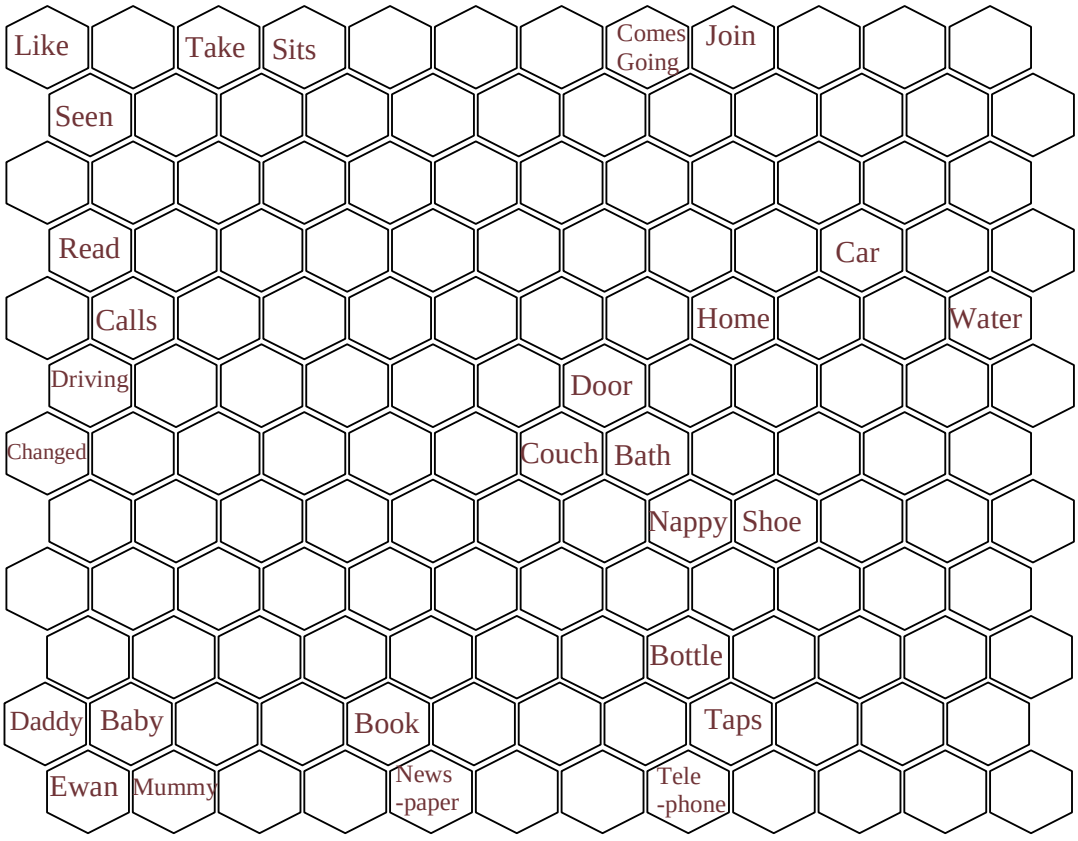


Figure 10 Best matching units for semantic feature self-organising map.

Table 3 Best matching unit locations of the upper associator recurrent self-organising for the speech time slices for two examples of the words 'change' and 'baby'.

Time slice	Change (1)	Change (2)	Baby (1)	Baby (2)
1	10 16	8 16	10 11	7 14
2	10 16	10 16	16 10	16 10
3	10 16	10 16	16 10	16 10
4	8 15	10 16	16 10	16 10
5	18 12	8 15	16 10	16 10
6	18 12	18 12	16 10	16 10
7	18 12	18 12	16 10	16 10
8	18 12	18 12	16 10	16 10
9	18 12	18 12	16 10	16 10
10	18 12	18 12	12 12	16 10
11	18 12	18 12	13 10	14 11
12	16 17	18 12	7 13	8 12
13	15 17	16 17	6 14	6 14
14	14 17	16 17	5 13	5 14
15	13 18	14 17	9 12	7 14
16	4 16	3 15		10 13
17	7 15	4 16		12 12
18	8 16	7 15		
19	8 16	8 16		
20	8 16	8 16		
21		8 16		
22		8 16		

When examining the best matching units for the speech slices for lower speech signal self-organising map and the speech slices and the semantic (visual) features upper associator recurrent self-organising network it is possible to identify that the word units have distinct representations. Such a representation would allow an agent to learn and recognise words without a real need for an understanding of the underlying speech sounds structures underlying the words. For instance, two representation from the upper associate recurrent self-organising map showing the best matching unit locations for each of the speech time slices for two examples for the words 'change' and 'baby' are shown in Table 3, with those for 'baby' being very different to those for 'change' and visa-versa. In the representation the x-axis location is the first digit in the pair and the y-axis location the second. This feature of the recurrent self-organising model in that speech in the first case is represented in terms of the individual word units is also felt to be found in children (Saffran et al. 2001).

In addition, when considering the sequences of best matching units produced by the lower speech signal recurrent self-organising map and upper associator recurrent self-organising map for the speech slices it is found that they are associated with specific speech sounds. The speech sounds include English phones from the DARPA phonetic alphabet such as N, IY, EY, AH, SH and CH. By developing such speech sound representation this would aid the learning of new words which contain similar sound sections. The ability to recognise that words are made up of distinct speech sounds such as phones also occurs in children once they learn the capacity to recognise individual word units (Dietrich et al. 2007). For the lower speech signal recurrent self-organising network, it is possible to identify various best matching unit regions for sequences of input slices that are found to be associated with different speech sounds. The top left hand area coloured light grey on Figure 11 represents the sound 'S' at the end or start of words such as 'matches', 'taps', 'news', 'seen', and 'comes', as well as the 'S' sound inside words such as 'newspaper', 'closer' and 'house'. The units in top right of the recurrent self-organising map in Figure 11 are those best matching units for the input speech slices that are associated with 'SH', 'CH', 'JH' and 'K'. These are sounds found in words such as 'fashion', 'shoe', 'shy', 'matches', 'couch', 'join' and back. The sound 'AH' is located in the lower left corner of the network and occurred in words such as 'telephone' (T EH L AH F OW N), 'Ewan' (Y UW AH N) and 'what' (W AH T). In the bottom right of the map there is a region of best matching units that is associated with the 'A' , 'I' and 'OW' sounds.

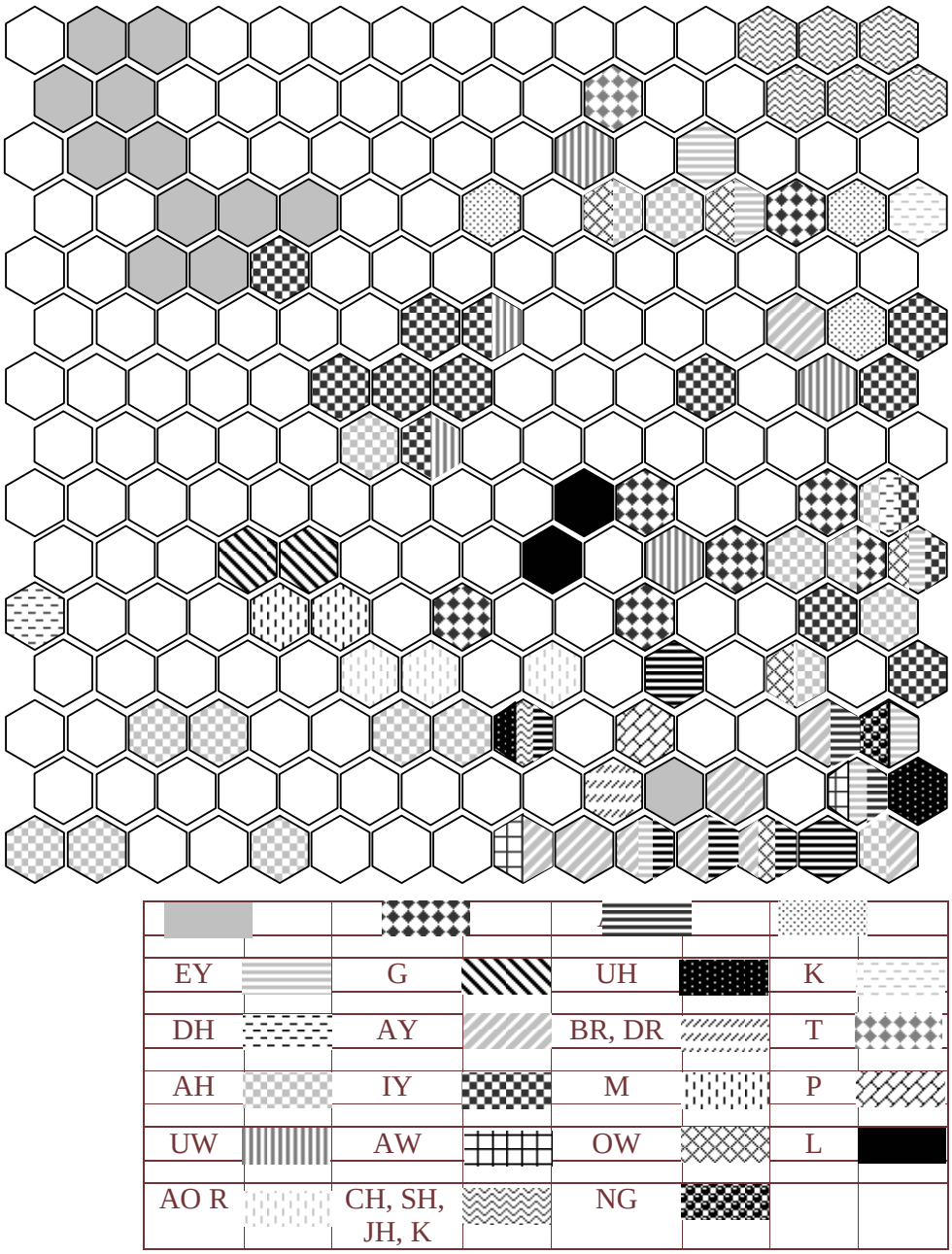
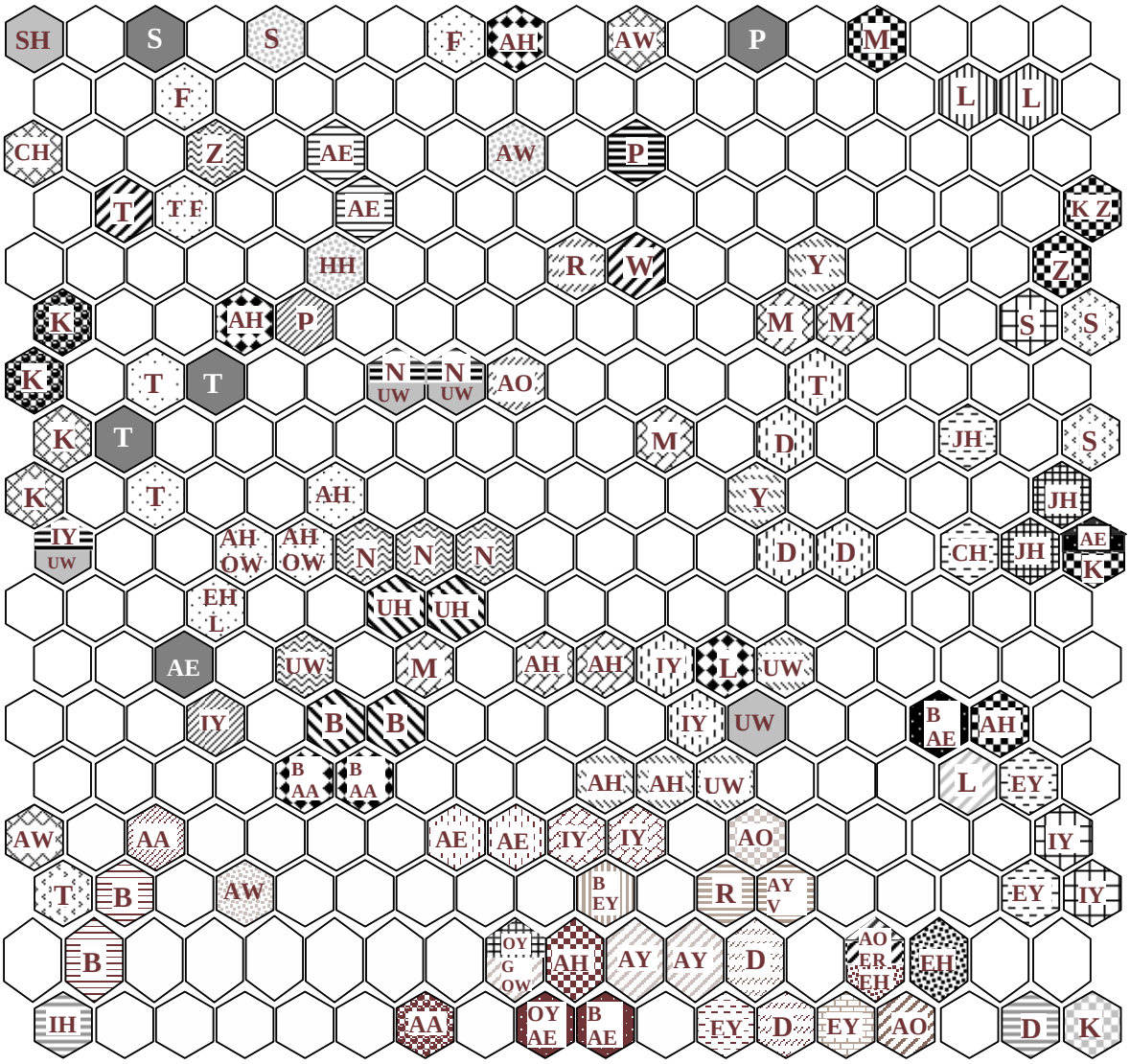


Figure 11 Units of recurrent self-organising map associate with specific speech sounds from word speech signal.



Shoe		Taps		House		Telephone		Bottle	
Couch		Comes		Like		Bath		Newspaper	
Nappy		Water		Door		Ewan		Car	
Mummy		Seen		Sits		Book		Daddy	
Changed		Join		Back		Like		Calls	
Baby		Driving		Read		Going		Take	

Figure 12 Units of upper associator recurrent SOM that have learned to associate specific speech from the speech signal.

The representation of the upper associator recurrent self-organisation network in Figure 12 shows the location of best matching unit sequences related to specific speech sounds for particular words. The labels represent the associated speech sounds (using the DARPA phonetic alphabet). The colour patterns indicate the associated words based on semantic (visual) features. It is possible to identify that various units of the network are associated with specific speech signal sounds and semantic (visual) features for words. For instance, the 'SH' speech sound for the word 'shoe' (SH UW) is associated with the top-left cell (x=1,y=1). This is the light grey unit labelled 'SH'. When considering the sequence of best matching units it is found that the 'S' speech sound for the words 'taps' (T AE P S) and 'house' (HH AW S) are located close together on the map in cell (x=1,y=3) and cell (x=1, y=5) respectively. It is also possible to identify further regions of the map that are associated with specific speech sounds such as the 'K' sound. Units x=1 and y= 6 and 7 are associated with the 'K' sound for 'car' (K AA R) and units x=1 and y= 8 and 9 are associated with the 'K' sound for 'couch' (K AW CH). The 'T' sound for the words 'telephone' (T EH L AH F OW N) and 'taps' (T AE P S) can also be seen on the upper associator self-organising map to be location in their own individual units but close together on the map. These units for the speech sound 'T' for 'taps' are represented on the map as units that are dark grey labelled 'T' and the units for the sound 'T' for 'telephone' are those that have black dots with a white background labelled 'T'.
It is also possible to see that related words are located in near units on the associated self-organising map. For instance, the sounds 'M' in 'mummy' (M AH M IY), 'D' (D AE D IY) in 'daddy' and 'Y' for 'Ewan' (Y UW AH N) and hence family-human related words are located around x=8 and y=13. This also the case for family-human related words for sounds such as 'AH' and 'IY' for the word 'mummy' (M AH M IY), 'IY' and 'AE' for the word 'daddy' (D AE D IY), 'AH' and 'UW' for 'Ewan' (Y UW AH N) and 'baby' (B EY B IY) 'B EY' speech sound. These units are located around x=11 and y=15. This is also the case for communication media such as 'telephone', 'newspaper' and 'book', with speech sounds associated with these words located close together on the self-organising map around x=10 and y=7. The sounds associated with these words at this location are 'AH' and 'OW' for 'telephone' (T EH L AH F OW N), 'UW' and 'N' for 'newspaper' (N UW Z P EY P ER) and 'UH' and 'B' for 'book' (B UH K). The speech sounds that are included in a specific word can be seen to be distributed in different units despite the semantic (visual) features input being the same for full length of word, which indicates that the speech signal and semantic feature representations are combined in such away that the two sets of activations have a joint impact on the associator recurrent self-organising network organisation. For the word 'like' (L AY K) the 'L' speech sound is located on x=14 on x-axis and y=16 and 'AY' speech sound is located at x=17 and y=11 and 12.

5 Episodic long-term memory in ACORNS memory architecture

There is growing interest in the use of episodic long-term memory for automatic speech recognition, hence in this section of the report we consider one model within the ACORNS memory architecture. The episodic long-term memory model in the ACORNS memory architecture incorporates a modification of the MINERVA2 approach to perform alpha character and keyword recognition. MINERVA2 is a computational multiple-trace episodic memory model that successfully predicts basic findings from the schema-abstraction literature. MINERVA2 simulates episodic long-term memory by first storing 'traces'. Inputs to the system - 'probes' - are compared to all of the traces in long-term memory. The retrieved 'echo' returns a vector containing additional knowledge that is unspecified in the input, e.g. its class. The activations are determined by the similarity between the input and each stored trace. Hintzman (1986) showed that such a model is able to create abstract representations of stored data, and that 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 implicit relationships between individual stored traces. In response to a probe, MINERVA constructs an echo by activating all samples in the training data – see Figure 13.

The main parameters of the model are (i) the feature representations, (ii) the similarity function, (iii) the activation function and (iv) the echo retrieval function. In the implementation the feature vector consisted of the standard representation used in ASR tasks – mel frequency cepstral coefficients (MFCCs). The class

labels (i.e. the identities of the keyword) are stored as blocks of features. The similarity between the input and stored traces has to be computed using an intermediate step that is different to Hintzman's original binary approach. In our implementations, the distance measure used is the Euclidean Distance:

$$ED_{It} = \sqrt{\sum_{i=1}^{n} (|I_i - t_i|)^2} \qquad (16)$$

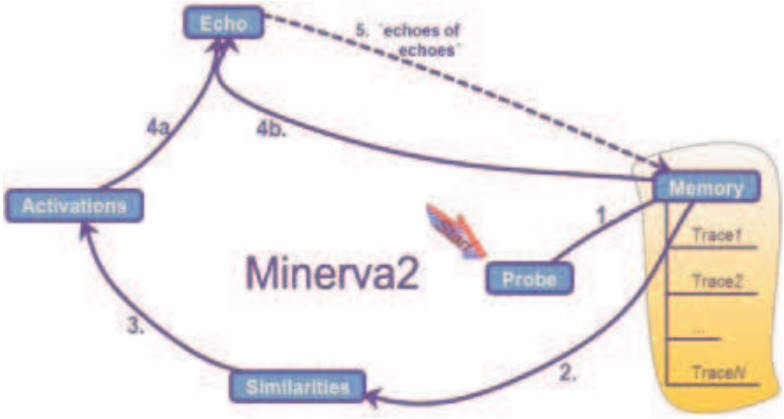


Figure 13 Schematic diagram of MINERVA2

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:

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

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 normalise *ED* in order to ensure that the range of $sim_{I,t}$ is between 0 and 1. To gain the final activation value *w* of the traces with respect to input *I*, the similarity measure is raised to the power of *p*. This in effect gives more importance to the most similar traces and less to those traces that are not similar.

$$w_{I,t} = sim_{I,t}{}^{p} \qquad (17)$$

Hintzman sets the value of the power factor *p* to 3, however *p* can have any value.

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 judge frequency and familiarity; it is defined as follows:

$$\text{int}_I = \sum^{T} w_{I,t} \qquad (18)$$

where *I* is the input, *T* is the total number of traces stored.

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

$$echo_I = \left(\sum_{t=1}^{T} w_{I,t} \cdot trace_t \right) / \text{int}_I \qquad (19)$$

where $w_{I,t}$ is the weight on trace *t* for input *I*, and *T* corresponds to the number of stored traces. Note that in our adapted approach, a normalisation of this value is necessary for numeric reasons.

MINERVA2 offers a powerful means for generalizing by accessing the fine detail retained in all the training data. However, it is severely hampered by its inability to model temporal sequence. MINERVA2 is essentially a single-frame classifier; hence moving to a corpus of utterance as provided by the ACORNS database requires the addition of a mechanism for handling variable length tokens. However, such a step constitutes a fundamental change in the underlying methodology. Prior to the development of a fully functional temporal episodic model, several intermediate solutions present themselves. In this study, a 'bag-of-frames' (BoF) approach was adopted as the configuration that involves the least number of assumptions about the temporal evolution of speech patterns. BoF simply means that a word is classified according to the accumulated response of all of its constituent frames regardless of the order in which they occurred.

$$bagsClass = \arg\max_{w \in c} \left(\sum_{n=1}^{N} echoClassVals_n \right) \qquad (20)$$

where *bagsClass* is the class that is attributed to the whole 'bag-of-frames' constituting an utterance, *W* is a class from the set of all classes *C*, n is the frame-index, and *echoClassesVals* are the values that the echo returns for all possible classes.

Before assessing model results, it is necessary to assess the relationship of single-Gaussians and Gaussian mixtures models (GMM) to the MINERVA2 approach. MINERVA2 is an instance-based approach which has an element of abstraction. Instead of performing the abstraction before the current problem (i.e. the current test input) is known, MINERVA2 waits for "training" of some of its parameters (such as setting the mean of the data to the current problem). At "training time" the similarity weighting in MINERVA2 substitutes all training data for a locally averaged, weighted single mean that best fits the current input. Hence, MINERVA2 models the various classes to be expressed using only one value per feature. This means that after training, its parameter complexity could be considered to be comparable to a single Gaussian.

On the other side, allowing for multiple Gaussian to represent the data could allow the storage of more and more detailed, "episodic" information, with the extreme being to allow one Gaussian per training frame. At the extreme end, the HMM can no longer approximate the variance as there is only one data point per sample Gaussian, and hence either the variance is set to zero or to an artificial value. If the variance were set to zero, the system would lose any power of abstraction. On the other hand, setting the variance artificially to any value above 0 means over generalising the dataset, and hence discrimination between classes should become very difficult.

Of course, it is extreme to model a whole sentence with one Gaussian. This has been done in those experiments that were performed on the ACORNS database, since the speech databases used only offers a general label for the sentences; it should be noted, though, that often in real ASR applications different classes will have some frames that are very similar. For example, sometimes the models of plosives such as /p/ and /t/ will include the generic silence and burst at the beginning of the phones. Each separation of parts of a speech signal spoken as a stream holds assumptions. As such, this can be seen as an extreme example of a problem that occurs in ASR.

In order to consider the suitability of the episodic long-term model some initial experiments have been performed on less complex data than what is found in the ACORNS databases. The database chosen for this investigation was the TI-ALPHA isolated word corpus. The data consists of 16 speakers (eight male and eight female) uttering the 26 letters of the US English orthographic alphabet ("A", "B", "C", etc.). The complete test set consists of 6628 utterances, and the complete training set consists of 4142 utterances. All experiments were conducted using standard MFCC features and their first and second derivatives, giving rise to a total of 39 features per frame. A 25ms frame was taken every 10ms. The classes corresponded to whole-word labels. Results were also obtained using a standard whole-word HMM baseline that employed left-to-right HMMs with three emitting states per model. A further HMM model was trained with only one emitting state in order to emulate the same 'temporally-invariant' model as in the BoF scheme. All HMM models were trained by incremental mixture splitting. The number of components per mixture was optimized for best performance.

In statistical pattern recognition, the process of generalization is achieved by combining information during training. For example, in state of- the-art classifiers such as HMMs or Gaussian mixture models *(*GMMs), training data is used to find the mean and variance of a single- or multi-component Gaussian mixture distribution of the data. In direct comparison, episodic long-term memory model does something very similar – it also computes the mean of similarity-weighted data; however, there is no overall mean as the similarity weighting attempts to substitute a general distribution for one that best fits the current input. Hence, the episodic long-term memory approach models the various classes to be expressed using only one value per feature. The consequence is that the use of such similarity weighted training data allows the constructed models to take into account the fine-phonetic similarity found within a frame. Therefore, a hypothesis to be tested is as follows: does the use of similarity-weighted training data enhance the model's recognition performance using the minimum number of model parameters? If so, then one would expect that episodic long term memory model would outperform a one-state single-Gaussian HMM, if it makes sense to take the similarity of such fine details into account. As can be seen from the results shown in Table 4, episodic long-term memory model clearly outperforms the single-state HMM.

Table 4 Comparison between a single-Gaussian and MINERVA2 model. Multi- Speaker (MS) and Speaker-Independent (SI) recognition results on TI-ALPHA data.

Classifier	Error Rate
MS: HMM S1 (single-Gaussian)	35.41 %
MS: Episodic Model	11.27 %
SI: HMM S1 (single-Gaussian)	39.97 %
SI: Episodic Model	27.53 %

HMMs typically use GMMs (rather than single Gaussians) in order to allow data belonging to one class to be modelled using different distributions. In effect, the training data is split up and clustered during training to a previously defined number of Gaussian distributions. This means that in the subsequent testing stage, partially clustered training data is compared to the unknown input. However, in direct contrast, MINERVA2 is based on the assumption that an online comparison of the input data to *all* of the training data leads to a more appropriate weighting of the information, and this may offer an advantage in recognition accuracy. It is interesting to find out just how well/badly MINERVA2, which uses episodic long-term memory would perform in comparison to HMMs using GMMs and/or multiple states. The first experiments were run on the complete test- and training data in multi-speaker mode, and the results are presented in Table 5. As expected, the best recognition performance was obtained using the three-state-HMM with 120 Gaussians per state.

In the second experiment a subset was developed using utterances from the English ACORNS database, which consists of 4 speakers (two male and two female) saying 6 different utterances. These 6 utterances contain 6 keywords and 2 different carrier sentences (i) 'daddy comes'; and (ii) 'where is the'. The keywords associated with the first carrier sentence are 'back' and 'closer' and the keywords associated with

the second carrier sentences are 'car', 'daddy', 'book' and 'nappy'. The training and test data consists of each speaker repeating the utterances 5 times, there are 120 training utterances and 120 test utterances. This approach allowed us to establish the performance of the episodic long-term memory model on speaker independent (SI) and speaker dependent (SD) data. All experiments were conducted using standard MFCC features for a 27ms frame taken every 13.5ms. The classes corresponded to keyword labels.

Table 5 Multi-speaker recognition results. S1 (S3): HMM with one (three) emitting states.

Classifier	Error rate
HMM S3 (30/60 GMM)	11.7 %
HMM S3 (1 GMM)	33.4%
HMM S1 (60 GMM)	11.9%
HMM S1 (3 GMM)	52.6%
Episodic Model (p=29)	27.5%

For the speaker dependent (SD) experiments the Minerva2 system shows superior recognition results to the GMM. The best GMM recognition results were found for 15 Gaussians per GMM. Experiments were performed with up to 120 GMM. This provides enough parameters to model every part of a sound in each of the test phrases. In the SI condition the 120 GMM model shows superior recognition performance over the MINERVA2 based episodic long-term memory system. It is noticeable that the number of Gaussians for optimal performance is rather high, (given that there are only about 15 utterances per model in the SI condition). This suggests that the individual Gaussians in the mixture generalise less than the echo response of MINERVA2 based episodic long-term memory model, and hence the decision may be based on even less information than the echo acquired by MINERVA2.

Table 6 Comparison between a single-Gaussian and MINERVA2 model. Speaker dependent (SD) and Speaker-Independent (SI) recognition results on ACORNS speech data.

Classifier	Error rate
SD: HMM S1 (single-Gaussian)	26.03 %
SD: HMM S1 (15 GMM)	10.94 %
SD: Episodic Model	5.0 %
SI: HMM S1 (single-Gaussian)	66.58 %
SI: HMM S1 (120 GMM)	44.80 %
SI: Episodic Model	58.33 %

In response to the Episodic long-term memory based MINERVA2 bag-of-frames model's ability to classify temporal speech data, the aim is to build on this by using a new model known as TEMM (Temporal Episodic Memory Model). TEMM not only overcomes the inability of MINERVA2 to use temporal sequences for recognition flexibly, but it also employs a prediction mechanism as an additional source of information. As the base operation, TEMM follows the principles of MINERVA2 and in doing so, 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. 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 (i) the goodness of fit of previous decisions (i.e. future decisions can influence past decisions), and (ii) 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.

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.

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.

6 Discussion and Conclusion

In this report we have reviewed how the ACORNS project as successful developed a memory architecture within which various long-term memory models have been developed towards the aim of an intelligent agent that is able to communicate. While these models in isolation can either focus on semantic or episodic long-term memory, within the overall memory architecture they complement each other to work towards the overall project aim. For the selective attention semantic long-term model (Section 4.2) it can be said that reinforced learning seems to compensate for keyword spotting. Focused attention does not directly lead to better recognition results in this type of a learning problem, but it may help word segmentation and therefore acquisition of word models. However, a reinforced learning algorithm can also detect keyword locations with a moderate accuracy. The attention-gating mechanism (Section 4.3) offers a semantic long-term memory based model that offers the opportunity to limit the data being introduced into the working memory and so prevents it from being swamped by the use of reinforcement dopamine-like feedback by learning to differentiate between speech and non-speech. The model when learning offers feedback in the form of the agent to itself as an immediate reward and from a caregiver in the form of a delayed award at the end of the auditory sample to achieve speech detection.

By expanding on the current NMF models it has been shown that within the overall ACORNS memory architecture it is possible to make use of the hierarchical nature of the architecture to perform phone and word recognition (Section 4.4). In line with the memory prediction model, we have strived to allow the representation of semantic (visual) features of words (Subsection 4.5) to work with minimal a priori knowledge, i.e. the processing architecture should not have any prior knowledge about what concepts it was expected to learn. Both self-organising maps and Biased Competitive Layers appear to be good theoretical and computational accounts for the emergence of conceptual units as well as being biologically inspired unsupervised learning algorithms. Both models solve the hypernym/hyponym problem on the conceptual level. The can transform a set of input features into a set of conceptual probabilities, i.e. a vector defining for every concept that has emerged during training, how well it applies to input. The only weakness of both approaches is that a maximum number of output units need to be determined a priori and this number has a strong effect in the resolution of conceptual distance. While the hyperonym problem is not very prominent in the speech corpus produced in first period with its limited words, it will additional records towards the corpus collected in period 2 when words like man and daddy, bear and toy, or food and apple are in the vocabulary of almost every child, indicating that the human cognitive system is able to deal with this problem.

The use of an adaption of the recurrent self-organisation map approach (Section 4.6) offers a new manner of temporal representing of speech emergence that combines working memory (activations) and long-term memory (weights). Similar to the neurocognitive model of Pulvermüller, the recurrent self-organising network model uses different regions of the associator recurrent self-organising network to represent different word sounds. This model of language acquisition as an emergent property offers interesting parallels to the memory-prediction theory of intelligent neural processing put forward by Hawkins and

Blakeslee (2004). Drawing on the hierarchical structure of the neo-cortex the recurrent self-organising memory model detects recurring patterns in speech. These patterns are stored in higher levels of the cortical hierarchy, where they are associated with visible and tangible objects, actions and concepts in the external environment through the use of semantic features. There are also parallels with the memory prediction theory as the self-organising networks have the same structure but perform different activities such as providing a contextual memory representation for lower and upper recurrent self-organising network, semantic feature representation and combine the speech signal representation with the semantic feature representation. The activations on the lower speech signal recurrent self-organisation network acts as the phonological loop in working memory while the semantic features network activations act as the visuospatial sketchpad in working memory. The final speech representation in a working memory episodic buffer model is a combination of the visual semantic feature activation representation and the speech signal representation. The weights in these models represent emergent speech patterns and are stored in semantic long-term memory.

Speech recognition may benefit from the use of some episodic long-term memory (Section 5). Episodic memory keeps the fine details of the underlying data, fine details that get abstracted away in semantic long-term memory. Episodic long-term memory for ASR has been defined as a system that has the raw data available. In statistical pattern recognition, the process of generalisation is achieved by combining information during training. For example, in state-of-the-art classifiers such as HMMs or GMMs, training data is used to find the mean and variance of a single- or multi-component Gaussian mixture distribution of the data. The episodic memory model on the other hand computes the mean of similarity-weighted data: there is no overall mean as the similarity weighting attempts to substitute a general distribution for one that best fits the current input. Hence, in the episodic long-term models the various classes are expressed using only one value per feature. The consequence is that the use of such similarity-weighted training data allows the constructed, very simple models to take into account the fine-phonetic similarity found within a frame. However, the semantic long-term memory models described here offer learning that produces weights and representations in a more emergent and less directed manner than currently available in the episodic long-term memory model. The semantic long-term memory models predominately offer learning and abstraction from examples in a manner typically found in the cerebral cortex rather than a comparison with a set of existing representations stored in the memory as found in the episodic long-term memory model.

The episodic long-term memory model offers a centre-of-attention-component, which can be seen as one of the main differences between instance-based and statistical models. In a statistical model such as HMM trained on example data, the model represents the mainstream speech patterns. The variance is influenced by all those speech patterns that are atypical. Such a model performs well with test speech that is very typical, but less so for those that are not mainstream. For an episodic long-term memory model instance-based system based on the same training data, the centre of attention can be said to be there where the test input lies. Just by comparing the bottom-up acoustical data, it will find the most relevant examples and shift focus there. This means that such a system pays always more attention to that data that should be the most relevant to the current test input. While statistical, pre-trained systems have no other choice but to predefine the centre of attention the way the training data says it is most likely that the class's centre is, instance based systems have the luxury of paying close attention to data that is the most relevant to the current input. Instead of predefining the centre of attention for all possible future data, instance-based minimum-distance systems can set the centre of attention to where it is the most relevant.

References

Baddeley, A.D. (2002) The psychology of memory. In: A. D. Baddeley, B. A. Wilson & M. Kopelman (Eds.) Handbook of Memory Disorders, 2nd Edition. Hove: Psychology Press, pp 3-15.

Baddeley, A. D. (1992) Working memory. Science, 255(5044), pp. 556-559.

Clark, E. V. (1973). What's in a word? On the child's acquisition of semantics in his first language. T. E. Moore (Ed.), Cognitive development and the acquisition of language (pp. 54-110). New York: Academic Press.

Dietrich, C., Swingley, D. and Werker, J. (2007) Native language governs interpretation of salient speech sound differences at 18 months. Proceedings of the National Academy of Science of United States of America, 104(41), 16027-16031.

Eysenck, M. W. & Keane, M. T. (2005). Cognitive psychology: A student's handbook. (5th ed.) Hove: Erlbaum.

Gleitman, H., Fridlund, A., Reisberg, D. (1999) Psychology, W.W. Norton & Company Ltd., Ltd.

Hawkins, J. & Blakeslee, S. (2004) On Intelligence. New York, NY, Times Books.

Hintzman, D. (1986) Hint Schema Abstraction in a Multiple-Trace Memory Model. Psychological Review, 1916(93), pp. 411 -428.

Honkela, T. (1997) Self-organising maps in natural language processing, PhD Thesis, Helsinki University of Technology, Espoo, Finland.

James, W. (1890) Principles of psychology. New York, Holt.

Kohonen, T. (1997) Self-organizing maps, Springer-Verlag, Heidelberg, Germany.

McClelland J.L &Kawamoto, A.H. (1986) Mechanisms of sentence processing: Assigning roles to constituents. J.L. McClelland and D.E. Rumelhart, eds, Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models, MIT Press, Cambridge, MA .

Mountcastle, V. (1978) An organizing principle for cerebral function: The unit model and the distributed system. In: G.M. Edelman & V.B. Mountcastle, Editors, The Mindful Brain, MIT Press, Cambridge, MA.

Neath, I. & Surprenant A. (2002) Human Memory: An Introduction to Research, Data, and Theory, Wadsworth Pub Co.

Pardo, J. V., Fox, P. T., & Raichle, M. E. 1991. Localization of a human system for sustained attention by positron emission tomography. Nature 349, pp. 61–64.

Posner, M. I. & Presti, D. (1987) Selective attention and cognitive control. Trends Neurosci. 10, pp. 12–17.

Pugh, K., Shaywitz, B., Shaywitz, S., Fulbright, R., Byrd, D., Skudlarski, P., Shankweiler, D., Katz, L. Constable, R., Fletcher, J., Lacadie, C., Marchione, K. & Gore, J. (1996) Auditory Selective Attention: An fMRI Investigation, Neuroimage, 4, pp. 159–173.

Pulvermüller, F. (1999) Words in the Brain's Language, Cognitive Neuroscience, 22(2), pp. 253-336.

Pulvermüller, F. (2002) A brain perspective on language mechanisms: From discrete neuronal ensembles to serial order, Progress in Neurobiology, 67 (2002), pp. 85-111.

Pulvermüller, F. (2003) The neuroscience of language: On brain circuits of words and language, Cambridge Press, Cambridge, UK.

Pulvermüller, F., Mohr, B. & Schleichert, H. (1999) Semantic or lexico-syntactic factors: What determines word class specific activity in the human brain? Neuroscience Letters, 275(81-84), pp. 81-84.

Saffran, J., Senghas, A., & Trueswell, J. (2001) The acquisition of language by children, Proceedings of the National Academy of Sciences, 98, 12874-12875.

Smith, L.B., & Yu, C. (2008). Infants Rapidly Learn Word-Referent Mappings via cross-situational Statistics. Cognition, 106, 333-338.

Spitzer, M. (1999) The mind within the net: Models of learning, thinking and acting, MIT Press, Cambridge, MA, USA.

Sutton, R. & Barto, A. (1988) Reinforcement Learning: An Introduction. A Bradford Book.

Tulving, E. (1972) Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of memory. New York: Academic Press.

Voegtlin, T. (2002) Recursive self-organizing maps, Neural Networks, 15(8-9), pp. 979-991.