Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D5.2 System capable of learning a 50 word vocabulary

Due date of deliverable: M24 (1 December 2008)
Final version Dec 23, 2008.
Submission date: December 23, 2008
Corrected version (title, number): Feb 2, 2009.
Second submission date: Feb 2, 2009

Start date of project: 1 December 2006 Duration: 36 months

Project coordinator name: Prof. Lou Boves
Project coordinator organisation name: Radboud University Nijmegen.
Version: 1.1

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	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

1 Introduction

Louis ten Bosch, Lou Boves

In this document we provide an overview of the experiments in ACORNS during the second year, emphasising the context within with these experiments were performed. In addition, this document discusses the ACORNS database that has been recorded during the second year of the project, and it provides a reasoned description of the feature approach that was used to improve the cognitive plausibility of the encoding of the visual channel in the multimodal experiments. The deliverable concludes with an overall discussion of the experiments and about how WP5 experiments give feedback to the experimental questions in other WPs.

1.1 Conclusions from the first year experiments

Experiments that were performed during the *first* ACORNS year were based on the 'Y1' database (Task 5.4, Deliverable D5.4.1). The Y1 database was specifically recorded within the ACORNS project. It contains utterances in 4 languages (Finnish, Swedish, Dutch, English), from 4 (2 male, 2 female) speakers per language. For Finnish, Dutch and Swedish each speaker uttered 2000 utterances in two 'speaking modes', viz. infant-directed speech and adult-directed speech (IDS and ADS). For English there were 1000 utterances per speaker (infant-directed speech). Per language the database contains in total 13 different target words. The syntax of all utterances in this database has been motivated by the simple syntactic structure that characterizes infant directed speech, while the choice of the target words was inspired by information available in Communicative Development Inventories (CDIs) and the language acquisition literature. Per utterance, there was only one target word (target concept). The experiments performed in the first year gave rise to four major conclusions, which are summarised in this chapter as a background for the research in Year 2.

The first conclusion, which is supported by almost all experiments, is that computational models are able to build and update internal representations of 'word-like' units in the speech signal such that these learned representations can be used to classify unseen new stimuli. Accuracies up to 95 percent and higher were contained using NMF, while alternative techniques appeared to be promising as well (cf. Deliverable D4.1, Stouten et al, 2007, 2008; ten Bosch et al., 2008).

Secondly, several experiments showed that these internal word-representations adapt to the most recent speaker. That is reflected in the shape of the learning curve over time, when stimuli are presented in a speaker-blocked fashion. Every time a new (previously unheard) speaker starts, the accuracy of the learner drops substantially – a result of the mismatch between the new speaker characteristics and the present internal representations. The internal representations must first accommodate to the acoustic properties of the new speaker, before the accuracy reaches its previous high level. This adaptation phenomenon is clearly shown in figure 1.1, in which the accuracy of the learner is shown as a function of the number of stimuli presented. The x-axis represents the stimuli; 8000 stimuli are presented in speaker-blocked mode. A new speaker starts at utterance 1, around 2000, 4000 and 6000. The dips in accuracy along the horizontal axis at around x=2000, 4000 and 6000 can mainly be attributed to the unseen acoustic characteristics of a new speaker compared to the previous speaker.

The third conclusion is that the modelling of hierarchical structures is a complex issue. The experiments specifically aiming at hierarchical (multi-layer) structures (see e.g. Del D.4.1, section 5; ten Bosch et al., 2008) were able to show glimpses of emerging hierarchy structure. One of these experiments used an elaborate tag set to force the learner to build internal representations of the *combination* of word and the speaker identity. As a result, over 40 instead of only 10 internal representations were built. These internal representations are genuinely speaker-dependent. For example, one of these internal representations corresponded with 'car' and 'Els', another with 'car' and 'Peter'. In ten Bosch et al (2008), it was investigated how these low-level representations could be organised such that speaker-dependent realisations of a word (in the example 'car') can be grouped

into speaker-independent representations. The ratio of between- and within-group (cluster) variance of the resulting groups is a measure for how each cluster 'stands' out among the other clusters. Figure 1.2 shows this ratio as a function of the number of presented stimuli. Internal representations start randomly, and so the ratio will be fairly large. After a few hundred utterances, the clusters can be identified more clearly.

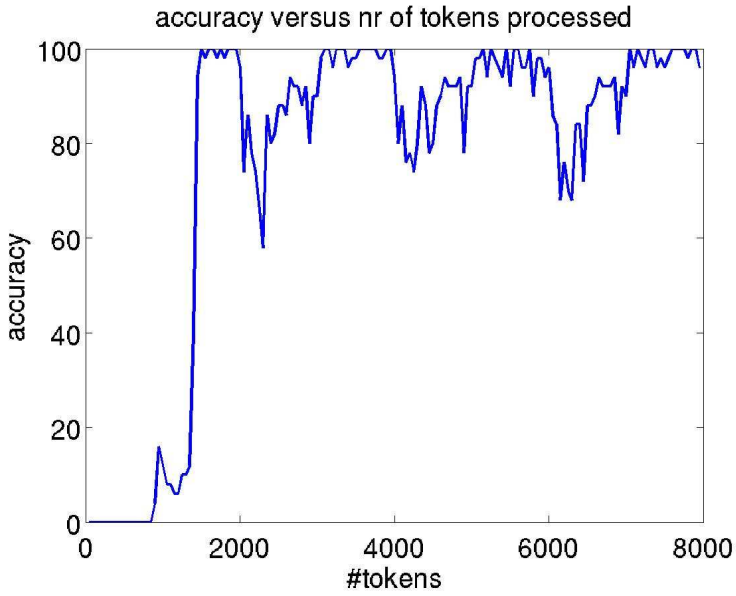

Figure 1.1. This plot shows the accuracy of the learner as a function of the number of processed utterances, taken from the Dutch Y1 database. The data are presented in speaker-blocked mode. There are four speakers. A new speaker starts at utterance 1, around 2000, 4000 and 6000.

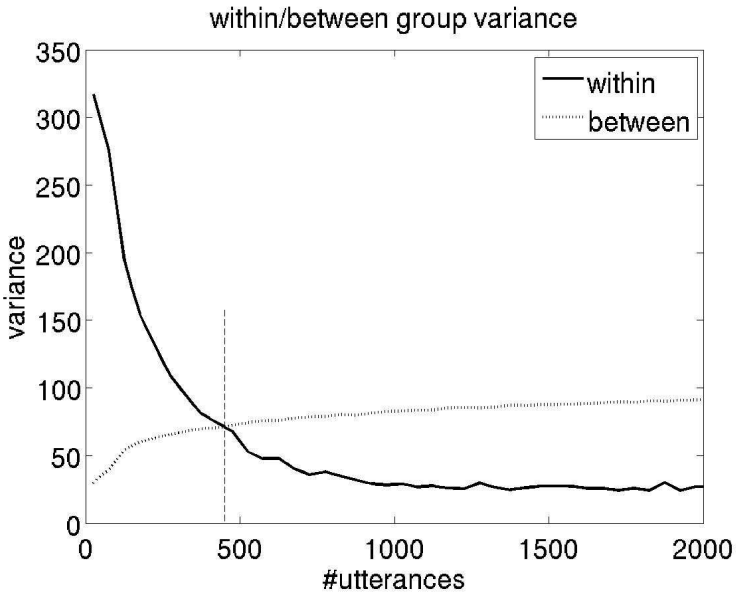

Figure 1.2. The more utterances are presented, the more each cluster stands out among the other clusters.

The fourth conclusion, in line with the results from most computational approaches, relates to the dependency of the learner's accuracy as a function of four parameters in the computational model. The four parameters are:

(a) how much data is required to *initialise* the internal models. This parameter has a clear interpretation in terms of cognitive plausibility of the resulting model: a successful training with a low value means that training of internal representations can recover from local minima. The amount of data that can be used to initialise an internal model will be related to the processing and storage capacity of the Short Term Memory (STM).

(b) how much data is required to *update* internal models (also this is related to the storage and processing capacity of the STM). The amount of data required to maintain or update representations may be less than the amount of data requited to robustly initialise representations. The experiments show that in general the amount of data required for *updating* the representations must be larger than for their initialisation.

(c) the frequency of updating the internal models. The learner might update models after each realisation (token), or update representations only when sufficiently many tokens have been perceived. In 'real life', this may be related to 'sleeping'.

(d) the way stimuli are used to update internal representations. Once a stimulus is perceived, it is well possible that it is used more than once for the update of the internal representations. It may be related to rehearsal.

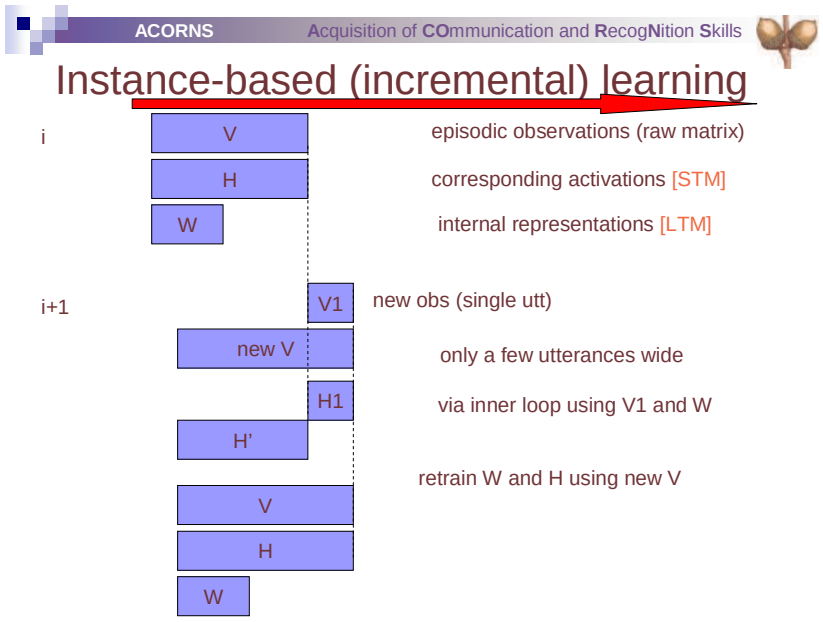

Figure 1.3. This figure shows a schematic picture of the instance-based incremental processing. The horizontal arrow represents the signal evolving over time. Vertically the evolution of the learning process is displayed.

In figure 1.3, all these parameters are shown in the context of an instance-based incremental learning procedure. As an *example*, we here show the essence of these parameters for an NMF-based type of learning. The horizontal axis in this figure represents the number of stimuli that is presented to the learner. At a certain learning stage (stage i) a raw episodic 'sensory data' matrix *V* is available (which is the input for NMF), in combination with associated activations *H* and the internal representations *W* (the output of NMF). The question is then, what exactly happens if the learning process is confronted with a new stimulus (which makes the learning process proceed from its current stage i to its new stage i+1). When new (unseen) stimuli are presented (denoted *V1*), first the episodic matrix *V* in STM is updated and it may happen that the oldest information in *V* is lost or partly lost (the amount of loss is

determined by the storage capacity of STM). By using the unseen stimuli *V1* as test-stimuli, a new activity matrix *H1* can be obtained as the result of the decoding of the new utterances. After that the internal representations *W* are updated on the basis of the old *W* and the updated *V* and *H* (the learning process is now in stage i+1).
The scenario described above relates to the case where the learner does not get any feedback about correctness (no corrective feedback). In another scenario, the learner may receive corrective feedback from the caregiver. In that case, the *H1* is directly based on the corrective feedback rather than the learner's own speculation about the stimuli in *V1*.
After that, learning process moves on to its next stage by observing and processing another new stimulus.

The parameters related to (a), (b), and (c) are directly reflected in this learning scheme. Parameter (a) determines when the very first initialisation step is taking place. This could be determined on the number of tokens perceived, but a cognitively more plausible way is to let the learning algorithm decide how and when it starts (both options have been investigated in experiments performed by KUL, Del D4.1, section 2.4). Parameter (b) determines how much data is used to update the internal representations. Parameter (c) governs the frequency of the update. Basically, we assume that leaning is an ongoing process, taking place after each stimulus, which implies that the update frequency is very small (say 1 utterance). (As said above, for humans, this parameter might be related to *sleeping* periods. The current computational model however does not have any sleeping mode.) Parameter (d) is more implicit, and is related to how often a once perceived stimulus, *while available in STM*, can be used to update the internal representations. This governs the amount of re-use of internally stored stimuli (while in STM).

The scheme (Figure 1.3) has a great deal of cognitive plausibility. It comes close to an algorithmic scheme at the second Marr level (Marr, 1982). For a learning system, it would not be plausible to allow several loops (epochs) over the entire database, as is usually done in conventional Automatic Speech Recognition (ASR). This would lead to internal representations that are constructed in a non-causal way. However, the *update procedure* that takes place after having observed new stimuli can re-use stored data available in STM more than once. The parameter in (d) exactly specifies this re-use.

2 Databases and experiments in Year 2

Louis ten Bosch, Lou Boves

The ACORNS experiments that we conducted in the second year ('Y2') are based on the experimental findings from Y1, and on discussions on how to exactly identify the strong and weak points of the various computational approaches. The strong point was that learning techniques (e.g. based on NMF and DP-ngrams) are indeed able to unravel the information in multi-modally presented stimuli and to build emergent internal representations of word-like units. Two major issues, however, were poorly addressed in the Y1 experiments. Firstly, the invariant symbolic tags are a poor means to 'code' the visual information presented in the multimodal stimulus, and secondly, the modelling of hierarchical representations was not addressed in detail.

To address both points, year-2 experiments have been defined with the major goal to cast light on the issue of hierarchy and the plausibility of the coding of the visual channel in the stimuli. These experiments are performed as part of the task T5.4, and will be discussed in detail in chapter 6 of this deliverable.

But before actually discussing these experiments, we will first describe the context in which they are performed. These experiments directly address the other tasks in WP5, which are Learning to Communicate (T5.1), Multimodal Integration (T5.2), and Model Architecture (T5.3). To make this clear, we will first discuss these tasks in more detail.

2.1 Learning to communicate (task 1)

2.1.1 Learner – environment

One of the basic assumptions in ACORNS is that acquisition of communication skills takes place within a communicative loop between a learner and its environment (specifically a caregiver). This acquisition, and more specifically the acquisition of language, is based on the learner's drive to understand its environment. This drive is ultimately rooted in the optimisation of the appreciation it receives from the caretaker, which is translated into an internal drive to optimally interpret each multimodal stimulus in terms of what it knows at that moment. Therefore all learning experiments make (explicitly or implicitly) use of this external loop.
The optimisation of the appreciation of the caretaker is a high-level goal, related to the *external* learning loop involving both caretaker and learner (see Figure 2.1). The 'interpretation' of the incoming stimuli in terms of internally stored representations is dealt with in the *internal* learning loop.

2.1.2 Learning drive and cost functions modelling the drive to learn

In the first year NMF-based experiments, we have used a minimisation criterion to obtain a good approximation of a given data matrix in terms of a product of two other smaller matrices. In the second ACONS year we elaborated on the interpretation of the mathematical target function in the context of high-level learning strategies. By *coupling* the internal and external learning loop, the learning drive to communicate is interpretable as a result of translating a high-level demand for communication into a low-level target function using the following cascade:

[a] *High level*: Optimisation of the appreciation received by the caregiver [via the external loop]
[b] Optimisation of the number of 'correctly understood' stimuli
[c] Optimisation of the interpretation of a stimulus GIVEN the so-far learned internal representations
[d] *Low level*: Optimisation of a target function [via the internal loop]

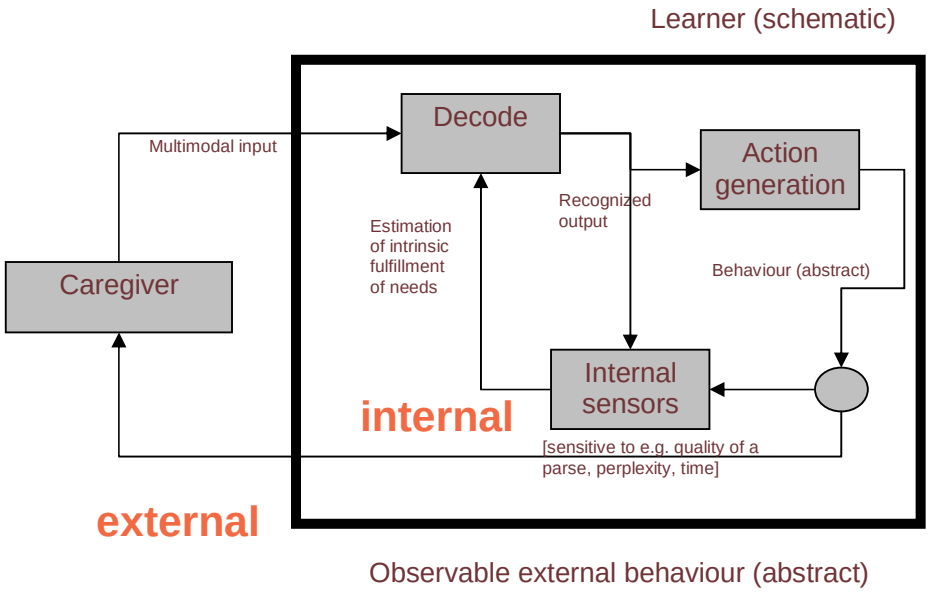

Figure 2.1. Global overview of the interaction between learner and its environment.

The explicit mathematical expression that is used in [d] depends on the learning algorithm. In the case of NMF and DP-ngrams, the expression can be formulated explicitly (Stouten et al., 2007, 2008; Deliverable D4.1). In the Concept Matrix approach (cf. , Deliverable D2.2) the target function is expressed in terms of the match between an incoming symbol sequence and any postulated word, by the evaluation of the corresponding word-dependent Concept Matrix.
For example, in NMF (that is, decomposition of $V = WH$ for structure discovery in the first layer) the target function is now chosen to be the Kullback-Leibler dissimilarity KL between the original episodic data matrix $V = v_{ij}$ and its NMF-reconstruction V-'*hat*' $= WH$

$$KL = \sum\sum v_{ij} * \log(v_{ij} / \hat{v}_{ij}) + \hat{v}_{ij} - v_{ij}$$

in which the sums are taken over all i and j (columns and rows) of V (see e.g. Stouten et al., 2007; Deliverable D4.1; O'Grady et al, 2008). In learning based on DP-ngrams, the target function is based on the distance between any new acoustic token and the set of stored prototypes (Deliverable D2.2).
The KL-distance expressed above can be expressed in terms of a Taylor series. Making use of the fact that the approximation is close to the original and by using the default Taylor series of *log*(x) around x=1, one obtains

$$KL = \sum\sum \frac{(\hat{v}_{ij} - v_{ij})^2}{v_{ij}} + \text{higher order terms}$$

which resembles a very close similarity to the X^2-distance used in statistics. The effect of applying this metric means that W (and H) converge in such a way that they statistically explain the vectors in V. If the distance exceeds a threshold, for example due to the unsuccessful attempt to explain a new unseen stimulus in terms of a too small set of internal representations, then this means that either one (or more) of the representations must be adapted, or that a completely new representation (i.e. W-column) must be initialised to explain the new stimulus in terms of W. This is interpretable as striving towards completion of a parse of the stimulus in terms of internal representations. This then, in turn, can be linked via the coupling of the internal and external learning loops to higher order learning goals, such as the more general strive to understand stimuli ('urge for learning') in the environment in terms of

what the learner knows. Eventually one may relate this to Maslow (1954) in the context of a hierarchy of 'needs' of any organism.

2.2 Multimodal Integration (task 2)

2.2.1 Combining information from different channels

In all experiments multimodal stimuli are presented in such a way that audio information along with information that codes the visual channel is available for the learner. The cross-modal combination of sensory information is very useful for learning: Roy and Pentland (2002) showed with a computational model that learning words is much easier in the presence of accompanying visual input, while Smith and Yu (2008) showed with real babies that the behaviour of babies can be modelled by assuming that they are able to associate cross-modal, cross situational information.
In the ACORNS learning algorithms, the information that is presented along the auditory and the visual channel can be combined at several levels. In almost all algorithms under development, the integration of sensory information takes place at an early stage, that is, at feature-level. This is done by creating one feature vector that encodes the information from all channels in a single stimulus representation.
The coding of the visual channel has been paid substantial attention in year 2. As already observed, an important basis of the Y2 experiments is the insight, obtained from the Y1 experiments, that the abstract symbolic invariant tag that associates each utterance in the Y1 database is a less than ideal coding of the information in the visual channel. The issues with respect to the visual channel are explained in detail in chapter 3 and will therefore only be summarized here.

1. The invariant symbolic tags do not allow between-token variance, as a realistic visual channel would do. Every time we see the *same* object, we might see it from different viewpoints. This variation in the visual presentation is natural and it is cognitively realistic to be able to deal with such variation.
2. The tags do not allow modelling any between-object variation within a given category or type. Although two plates both are referred to as 'plate' by their tag, they represent different objects, perhaps with different properties.
3. The tags represent a crisp categorical and nominal between-type difference. According to the symbolic tags, a cat is as dissimilar from a dog as it is from a human. There is *no meaningful distance measure defined on an unordered set of symbols*.

To address these problems we decided to investigate a better coding of the information that is available in the visual channel: a *visual/semantic feature coding*.
Although the conceptual difference between using invariant symbolic tags and real-valued feature vectors is substantial, in practice the transition between the two annotation systems and the exact processing based on these coding schemes can be made very smooth. In fact the tag-construction can be considered a special case of the more general feature approach. If each tag is 1-1 with a column of a diagonal identity matrix, the coding using tags and features is identical (apart from the evident implementation details). This means that the use of a feature matrix is a powerful tool to manipulate the complexity of the visual coding.

2.3 Architecture (task 3)

In year 2, the architecture has been discussed to obtain an elaborated architecture that could serve as a basis for ACORNS Y2 research and is complete enough to serve for Y3 experiments. Also a new integration computational platform has been discussed and consolidated (see Figure 2.3).
The memory architecture displayed in Figure 2.3 serves as conceptual guide line for the learning algorithms. The architecture is informed by general ideas about the functionality of human memory as described in the psychological literature. At the same time, it can be argued that it is compatible with the general concepts of the memory-prediction theory (cf. the summary drafted for the SAC members that is available on the ACORNS Public Website). Multimodal input is received from the environment (Figure, mid left). The stimulus is stored in the sensory store, which only can hold information for a

few seconds. From the Sensory Store, information (gated copies) is stored into the Working Memory. This Working Memory can store information for about a minute (if no rehearsal takes place). Information that is rehearsed is stored in Long Term Memory (which can hold information for a life time). In combination, the information in Working Memory and Long Term Memory yields the weighted activation of internal representations, which generates a response that is observable as a (virtual) action.

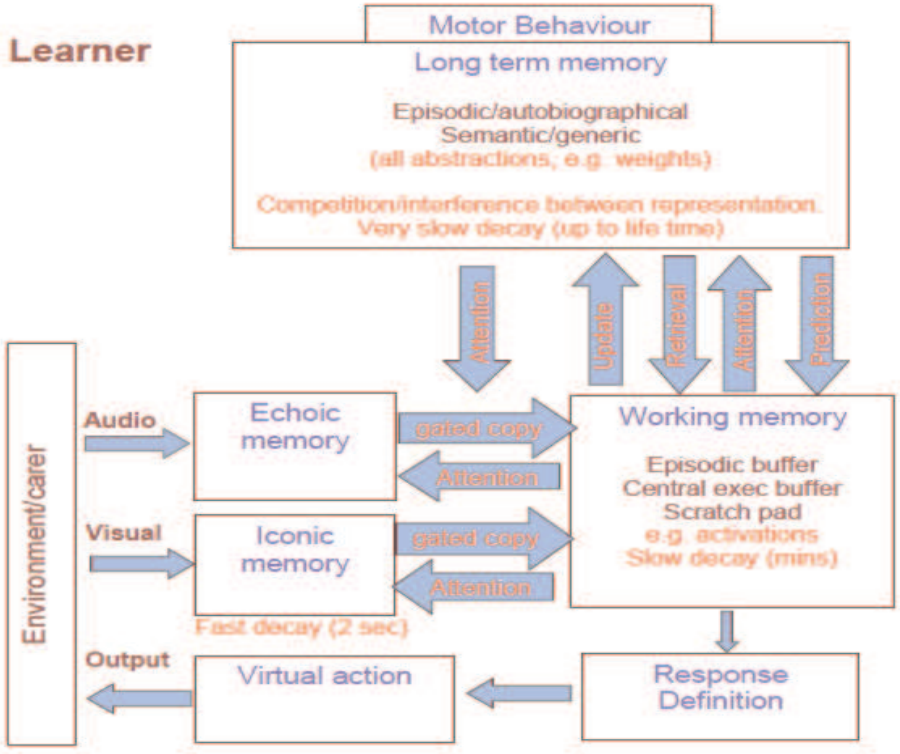

Figure 2.3. Memory architecture that underlies all end-to-end learning algorithms. The stimulus is stored in the sensory store. From the Sensory Store, information (gated copies) is stored into the Working Memory. In combination, the information in Working Memory and Long Term Memory yields the weighted activation of internal representations, which generates a response that is observable as a (virtual) action.

In Figure 2.3, the working memory unit receives a *gated* version of the current audio and visual input (from Echoic and Iconic memory), which produces at the 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. (Activations are in STM, weights in LTM). These weights are updated 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. An attention mechanism is used to control the updating of the learned weights. By combining the weights and the current activation patterns the model can perform activities such as retrieval and prediction.

Figure 2.4 focuses on the learning process, on the integration of visual/semantic features with audio information. It is a visualisation of the processes that takes place after a stimulus has been presented, and shows a conceptual representation of the hierarchical structure of the ACORNS learning system.

The audio and visual inputs are combined with the learned weights to produce representations at different levels of abstraction. The left 'region' *A* represents activations that are learned based only receiving audio input and as such are the representation of speech units such as phonemes. The second *A* region provides representations of the words by combining semantic (visual) information of words, for example provided by a Kohonen net with the phoneme representation previous produced by the first *A* region. The representation is based on learned weights on the upper layer of the model. The rightmost *A* region provides activation patterns that represent an utterance, based on learned weights.
The semantics (visual input) follows the same path as the audio. The input is the activation of semantic features in working memory (first layer, left box). This is used to train the weights (W) in long term memory. It also leads to a higher level (more abstract) representation of conceptual activations (A, middle region) in working memory.

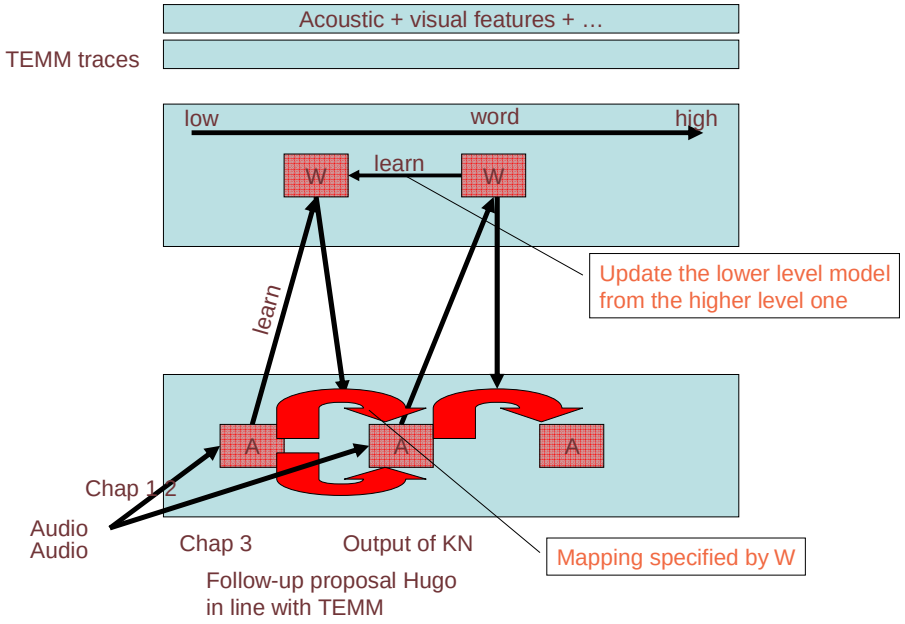

Fig 2.4. Processes that take place after receiving a stimulus. The picture includes 'traces' similar to those used in Minerva2/TEMM.

In the semantic part of the ACORNS model, the complete scene is presented to the iconic memory, while only the representations of the relevant object is maintained and processed in working memory (called episodic buffer in Figure 2.3). Episodic and semantic memories are made by changing the weights of model in long-term memory. These weights can be used to restore the activation in working memory if this is required. Conceptual activations in working memory can be used to find the auditory representations of word used to refer to these concepts.

3 A Feature Set for Simulating the Learning of First Words

Michael Klein, Louis ten Bosch, Viktoria Maier, Mark I. Elshaw, and Hugo Van hamme

3.1 Introduction and Motivation

The simulation of the acquisition of first word-like units involves not only the presentation of acoustic signals, but also information that makes it possible for the learning agent to associate acoustic signals with something visible or tangible in the environment (e.g. Smith & Yu, 2008; Gopnik et al., 2001; see also Hart et al., 1995; Holzapfel et al., 2008; Gold et al., 2009). In the Year 1 experiments the association between acoustics and environment was too crisp and unique, because the environment was coded in the form of unique tags. The use of tags caused three conceptual problems, all of which affect the cognitive plausibility of the experiments:

1. The invariant symbolic tags do not allow between-token variance, as a realistic visual channel would do. Each time we see the *same* object we might see it from different viewpoints. This variation in the visual presentation is natural and it is cognitively realistic to be able to deal with such variation.
2. The tags do not allow modelling any between-object variation within a given category or type. Different diapers are referred to as 'diaper' by their tag, despite the fact that they represent different objects, perhaps with different properties.
3. The tags represent a crisp categorical and nominal between-type difference. According to the symbolic tags, a cat is as dissimilar from a dog as it is from a human. There is *no meaningful distance measure defined on an unordered set of symbols*.

The uniqueness and crispness of the tags had a direct impact on all learning algorithms, since the tags determined whether or not a representation for a new 'word' should be created (or whether an existing representation had to be updated). Thus, it was evident that the crisp tags had to be replaced by a fuzzier and less crisp and unique representations of the environment to which the utterances refer to. Fuzzy representations would no longer force the learning agent to create new internal representations for new 'words', since it would now be possible that the simulated visual input might refer to an object or quality that had been encountered before.

While the conceptual problems with the crisp tags could have been tackled in many different ways, we decided in favour of a solution that will enable creating links to the development of adult semantic representations and processing. In doing so, we decided to encode the scene to which speech utterances refer with 'features' that mix visual and more abstract semantic properties of objects, qualities and actions. This allows us to avoid using tags, which are equivalent to the linguistic concept of 'word', and therefore amount to unrealistic pre-existing meta-level knowledge. In addition, using visual/semantic features will allow us to account for a number of behavioural findings, such as overgeneralizations during first word acquisition.

An approach that is relatively simple, while still being able to deal with all of the problems above is the use of a mix of visual and semantic features. Such features are closer to the perceptual reality of the language acquiring child and therefore cognitively more realistic. Features can be used to specify an individual which falls into several categories (e.g. a man, who is the daddy, who did not shave today, who is big, and sleeps) without giving away a priori which lexical items are used in a specific utterance. Further, a feature representation can reflect similarities between objects, and can be used to replicate many of the behavioural findings related to the semantics in language acquisition. It also allows different instances of a category to look different.

3.2 Environment, Ontology, Lexicon and Sentences

As a first step we designed a *virtual* environment resembling a possible *real* environment in which a child can learn first words. The environment involved the child itself, its care givers and possibly additional adults, standard household items, food, and a number of toys, in particular toy animals and vehicles. Based on the objects in this environment, we created the target vocabulary of 50 words. The nouns consisted of the words “baby”, “mummy”, “daddy”, “man”, “woman”, words referring to individual household items and toys, as well as more abstract words referring to categories such as ‘bird’ and ‘animal’. Adjectives were then selected depending on how well they could be applied to describe and distinguish the persons and objects in the scene. Verbs were chosen to denote suitable actions fir the objects. Words were cross-checked with CDIs (Communicative Development Inventories**)**. Words did not need to adhere exactly to the CDI-findings, but they do have the same phonetic/morphological complexity as the words figuring in the CDI.

3.3 Feature Coding

Given the many possibilities of coding meaning in features, a task force involving member of most ACORNS partners was appointed to develop a suitable coding scheme. Several possibilities were discussed. Simple binary features cannot distinguish the absence of a feature from ignorance about or irrelevance of a feature. For example, not knowing whether an item is edible would give the feature *edible* the value 0; the same value would apply if it is known that the item is not edible. In addition, binary feature cannot code probabilities and intensity values. While 3-valued features (e.g. -1 not present, 0 not know, 1 present) would solve the first problem, it cannot code probabilities and intensity values. More importantly, a 3-valued coding appeared to be problematic for some learning algorithms. For these reasons we decided to use features and anti-features with continuous intensity values and additional values for probabilities. Features and anti-features are as powerful as 3-valued features in coding the distinction between absence and ignorance. Not knowing whether an object has a feature would be coded by setting both the feature value and the anti-feature value to zero. However, in contrast to 3-valued features, they allow the coding of continues values. Therefore, they can code both, intensities and probabilities.

We came up with a number of (mathematical) constraints that have to hold about the feature set and the coding of objects with features. These constraints can be grouped into two types:

Logical Constraints:
The sum of the intensity values of features and anti-features should never be bigger than 1
Probability value of feature and anti feature should match

Semantic Constraints:
Unique decomposition: any combination of superimposed feature vector should be able to decompose into its parts in a unique way.

Since several other semantics constraints are essential in guiding the choice of features, we will discuss them in the next section

3.3.1 Choice of Features

Very little is not known about which features humans actually use to categorize objects or events. Recently, a lot of attention has been given in particular to the representation of actions and their arguments (agent, patients, etc.), i.e. how the brain represents the fact that certain individuals play a specific role during an action. However, the insights obtained with this method are limited to simple types of objects and actions humans are involved with. There are three main types of behavioural observations that can be used to define a set of features. First, by investigating the way human classify objects or actions, conclusions can be drawn about the features they use. Second, people can be asked what sort of properties they associate with specific objects and events. Third, semanticists have come up with distinctive feature sets to distinguish the meaning of words. We used all the above sources of

evidence as inspiration for the generation of our feature set. Since ACORNS aims to combine a stream of auditory information with a visual stream, we tried to restrict ourselves as far as possible to features belonging to the visual modality. Therefore, the creation of the feature set was mainly based on conventional *semantic constraints*: Different concepts have different features, similarity between concepts should result in a small distance in feature space, and the features should reflect hyper- and hyponymy.

To facilitate the creation of feature sets that adhere to all constraints we built a feature set generation tool (see Fig. 3.1). The final list of features is included in Appendix 1.

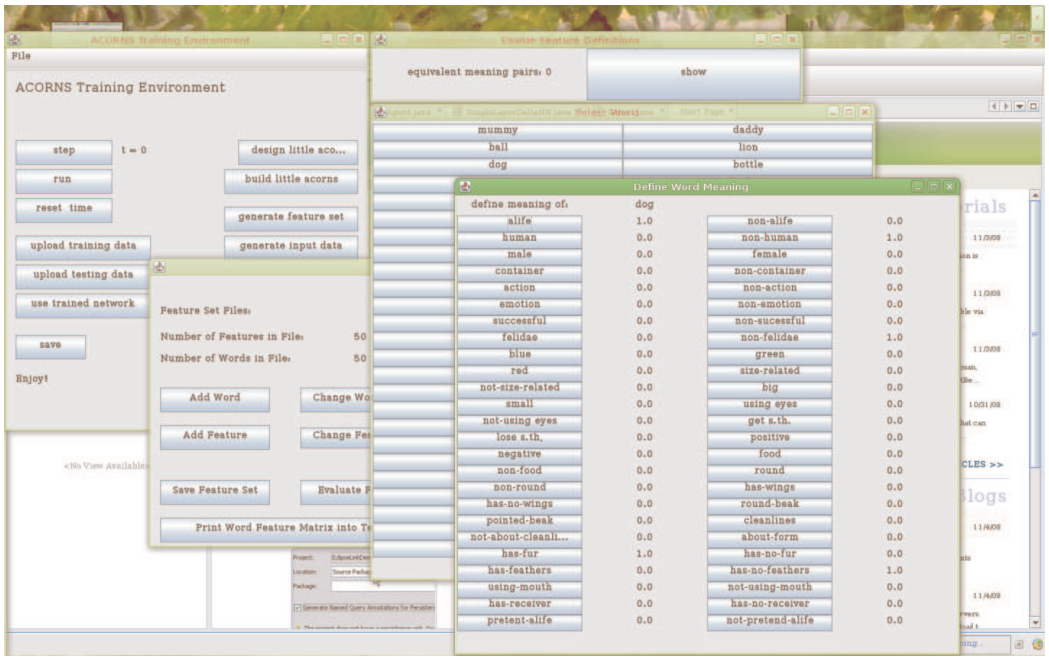

Fig. 3.1 Set-up of the feature set generation tool

3.3.2 Feature Representation

In ACORNS experiments, feature vectors will be presented at the same time as the wave file and the temporal link/pointer into a larger wave file (that points to the utterance that is to be presented to the learner). Given an utterance and a set of features, a feature presentation token (FPT) has to be generated. We have substantial freedom and flexibility in creating the visual/semantic environment that can ground the spoken utterance. The use of features allows us to create a gradual and graceful evolution from crisp tags to fuzzy representations of a scene in which fewer or more items may be present than mentioned in an utterance.
Moreover, the use of features allows us to code the utterance (by limiting the features exactly to a precise representation of the meaning of the utterance) as well as the scene, somewhat independent of the meaning or contents of an utterance.

4 On the ACORNS 'Y2' database

4.1 Introduction

The ACORNS Y2 database serves as an annotated speech repository for the Y2 experiments. These experiments aim at the detection of "words" by hypothesizing and strengthening internal representations of word-like units.

The database is recorded in three languages: (British) English, Dutch and Finnish. First a preliminary set of utterances was constructed for the Dutch Y2 database. A similar (not 1-1 translated, but identical in underlying statistical properties) version for British English was created and corrected by partners in Sheffield and Helsinki. These corrections were used to improve the Dutch version, and to create a Finnish version.

4.2 Construction of the sentences, including ecological validity

Design. Keywords. The 50 keywords in the Y2 database for Dutch were chosen on the basis of several factors. The English version of the keywords is included as Appendix 3.

(a) Ecological validity/plausibility. Not necessarily each single word, but the set of keywords should be ecologically plausible. CDIs show how the receptive (passive) lexicon grows with the number of months. The current list is inspired by the information from CDIs and the Dutch literature on language acquisition. Also ETLA (ecological theory of language acquisition, Lacerda et al., 2004) has been taken into account as one of the guidelines.

(b) Phonetic aspects. The set of 50 words has been made more interesting from a phonetic point of view without sacrificing the ecological plausibility by the selection of a few specific keywords. In the current word list, there are several word pairs that are or can be considered minimal pairs: examples *in Dutch* are heeft – geeft, zie – zei, ronde – rode (has – gives, said – sees, round – red). Furthermore, Dutch inflection of adjectives is a source of phonetic variation: rode auto – rood vliegtuig (red car – red airplane).

(c) Semantic factors. Semantics also played a role. We wanted to have hyper- en hyponyms figuring in the list, in order to be able to see the effect of different encodings in terms of semantic/visual features. As a result, the word 'car' appears next to 'porsche', 'animal' appears together with 'cat', 'dog', etc. It is useful to note that it depends on the word-feature coding matrix whether these words are actually hyper/hyponyms. For example, by using a 1-1 word feature relation, the set of words becomes 'flat' without any hierarchy.

Syntax. It is known that in English infant-directed speech the relevant noun is mostly located at the end of the utterance. The same is true for Dutch. This was taken into account in the construction of a grammar that was used to generate candidate sentences. The grammar, expressed in the BNF formalism, is included as Appendix 2.
The sentences in the Y2 database have been constructed in such a way that context and keyword are not necessarily correlated. A BNF was used to generate a large number of candidate sentences in which carrier phrase and keywords were combined. The BNF was not weighted, which means that special attention needs to be paid to obtain the desired (relative) frequency of the keywords. From the pool of candidate sentences, those with incompatible adjectives (such as in the utterance 'there I see a red green airplane') are ruled out.

Sentences. From the cleaned BNF-generated pool, 2000 sentences were selected randomly. Corrective utterances such as 'no, I mean FISH' with emphasis on *fish* have also been added.
A sentence contains minimum 1 target word and at most 4 target words. The median number of target words per sentence is 3. The number of target words that is actually applied in an experiment provides a way for making a scalable paradigm in learning complexity.

4.2 Speakers

The Y2 database consists of utterances read aloud by 10 different speakers. Per language 4 speakers (the same persons who also produces the Y1 corpus) utter 2000 'regular' sentences (see below) plus about 150 corrective sentences, whereas 6 additional speakers read aloud a (randomly chosen and fixed) subset of 600 utterances.

4.2.1 Speaking style

For the Y2 database, the speakers have been asked to imagine talking to a young infant of about 18 months old. The result is infant-directed speech (slower than average speaking rate, fairy-tale telling like), without the extreme characteristics of speech ('motherese' or 'parenthese') addressed to very young babies.

4.3 Recording

Recording was done in a low noise sound recording booth at the Radboud University. The recording conditions and equipment were identical to the recording and equipment used for the Y1 database. Sound files were digitally recorded in PCM wav format at 44.1 kHz sample frequency, and converted to 16 kHz, mono, little-endian, 16 bit/sample wav files.
In line with our experience with the Y1 database, 2000 utterances is about the maximum an average speaker is willing to do. It takes about 2 full hours, divided over two sessions.

4.3.1 Annotation / verification

After the recordings, the wave files and the prompt sheets were presented to SPEX. The task for SPEX was to provide the time stamps in the wave file and to annotate them in accordance with the prompt sheet. On the prompt sheets, each utterance was given a unique integer. This integer was used to mark the corresponding time stamp. This was all done in Praat, and the result is a collection of Praat text grids. These text grids have been converted to a cor-formatted file. This cor file has tab-separated columns: the name of the wave file, begin time of the utterance (in sec), begin time of the next utterance (in sec), and the annotation on word level. Usually the annotation is exactly the same as the one in the prompt sheets.
In case the speaker made a mistake, there are two possibilities. (a) it is corrected directly after the utterance. In that case, a time stamp is defined with annotation '<X>' for the reparandum and the correct annotation is aligned with the repair. (b) there is no correction by the speaker and the error was only detected by SPEX after the recording. In that case the annotation from the prompt sheet has been repaired such that the resulting annotation is in accordance with the actually uttered speech.

Metadata are available in different formats. For Dutch, cor-(corpus) files were created with the advantage of easy readability in MATLAB. The corresponding xml files were created by Leuven. The Finnish and British English corpora are accompanied by xml files. Cor files and xml files contain the same information and are 1-1 convertible.

KUL has processed the Dutch Y2 database in the version it has originally been posted by RU. By using HMM-based ASR technology, KUL was able to spot subtle and less subtle discrepancies between the speech and the annotation. A list of about 30 errors has been provided to RU, which were corrected in the Dutch Y2 database.

4.4 Contents of the ACORNS Y2 database

4.4.1 Scene

A simple but realistic scene has been taken as a starting point for the Y2-database. The scene is defined by and represented by a list of objects (including persons), properties (colours, shapes, sizes) and actions (a very limited number). Each of these items is associated with 1 literal target word. In total there are 50 of such target words. A scene does not make much sense if the learning agent cannot

act in that scene, at the very least in terms of gazing at one of the objects in the scene. To that end, the response by the learner, which is available as the state of activation of internal representations, will be translated into gazing time (Norris, 2005; Allopenna et al., 1998). This means that the output of the learner eventually can be interpreted as defining a probability distribution of gazing time on the set of objects and actions in the scene.

4.4.2 Communication

The objects, properties and actions defined in the scene lead in a natural way to a set of propositions about and questions about the scene. An example is given below.

Suppose {apple + cow + red + take + see} are keywords that relate to the scene. In that case, it makes sense to have propositions such as 'daddy sees a red cow', 'I take the apple' and questions such as 'where is the red apple ?', 'do you see the red cow ?'. The conversion from the list of keywords to a pool of possible sentences is based on the use of a finite state grammar (see below).
Apart from the propositions and questions described above, the database also contains additional utterances that make a dialogue between carer and learner possible. These additional utterances are utterances that correct the learners reply in the previous turn. Here an example is presented.

(suppose the scene contains a red and blue telephone)

carer: 'here we have a nice red telephone'
reply: 'blue' + 'sheep'
carer: 'no I mean the RED one'
(with prosodic marking of 'red')

Observe that we have to choose to correct either 'blue' or 'sheep', since it would be completely infeasible to pre-record sentences with all possible combinations of two or more keywords. That means that the number of different additional sentences is in theory equal to the number of keywords.

4.4.3 Contexts

To avoid a bias in the detection of a keyword based on insufficient variation of its context (such that the context cues the keyword), the set of carrier phrases had necessarily to be very limited. In the ACORNS corpus the selection of carrier phrases is such that each of the carrier phrases occurs randomly and sufficiently often not to cue the keyword(s).
Examples of useful contexts are:

```
there is a lion and a duck.
here is ...
he sits on...
what is ... ?
the mommy sleeps.
she gives a dirty tree.
do you like a [adjective] [object]?
```

4.5 Finnish and British English: specific issues

Toomas Altosaar, Guillaume Aimetti

This section describes the most significant facets of motivation and procedure for the production of the second ACORNS corpus for the English and Finnish languages.

4.5.1 Prompt ordering

The sentences in the database make up three different categories, depending on their function:

Dialog-normal:	Here is a small toy and a dog.
Dialog-query:	Cat?
Dialog-corrective:	No, I mean DOG.

As for the Dutch database, the 2000 dialog-normal elements were first ordered to avoid dwelling on a single subject across consecutive utterances. This was performed in order to avoid a loss of novelty in the utterances and the speaker changing the prosody due to becoming too familiar with the objects. Next, dialog-triplets were formed by adding query and corrective elements to some dialog-normal elements. The dialog-triplets were equally spaced apart in the material at a ratio of 13 dialog-normal elements to one dialog-triplet. Query-verbs were then added and also spaced equally apart. Finally, the sentences were split into two sets and a randomized set of 50 keywords in isolation pre-pended to each set to form a single 2397 element list of prompts.
The 2397 prompts were then split into 24 sets in a non-uniform manner so as to remove recording task monotony by providing recording set duration novelty to the speaker during the recording process. Target set sizes of 50, 100, and 150 utterances were employed resulting in measured recording times of approximately 4, 7 and 11 minutes per set, respectively.
The 24 prompt sets that included all of the 2397 prompts were designed to be spoken by each of the year 1 corpus speakers, speakers 1-4. For verification speakers 5 through 10, a 600 element subset of the 2397 set was used that consisted of 50 keywords spoken once in isolation followed by 550 other types of prompt elements and otherwise followed the statistics of the larger set. The 600 prompts were split into six sets that also followed a set size modulation using set sizes of 50, 100, and 150 elements.

4.5.2 Recording Software

For Finnish and British English, a software application designed for recording and generating speech corpora semi-automatically was employed. The application provided output to two computer displays; one for the speaker and another for the technician who actively controlled and monitored the recording process to ensure corpus production quality. The technician would request the software to present the next prompt to the speaker, the prompt would then be presented on both displays, the speaker would utter it, and the technician, when satisfied, would then move on to the next prompt. If any anomaly was detected by either the speaker or technician, e.g., a reading error or a flagged technical error such as signal level clipping, the prompt could be re-recorded immediately with only a minimal affect to the speaker's "rhythm" of the reading process.
This approach reduced the number of errors significantly, e.g., no missing prompts or out-of-vocabulary words are known to exist within the English and Finnish recordings. Besides real-time recording to the file system, start and stop times for each prompt were automatically recorded by the software to be used later in the utterance segmentation process.
To promote fixed-loudness recordings in an anechoic chamber, all speakers wore headphones where auditory-feedback was introduced to reduce the Lombard effect. To ensure similar recording characteristics between the Year 1 and Year 2 recordings, the same anechoic room, setup, and equipment, e.g., the exact same microphone, preamplifier, and A/D converter, was utilised.
To retain speaker attention throughout the recording task, speakers were forced to take a rest break of at least 30 seconds between each set. The number of sets recorded per day varied according to the capabilities of the speaker and fell in a range from a minimum of 200 utterances to a maximum of 1000. In all cases a prompt set was always recorded as one unit, i.e., at least 50, 100, or 150 utterances were recorded without any rest breaks in between prompts. Speakers presented with the 24 sets (2397 prompts) required 4 to 8 recording sessions spread out over different days to complete their task. Speakers of the smaller 600 element verification set were all able to complete their recordings within a single 75 minute period of time.

Iconic prompts
Speaker fatigue was reduced by the use of *iconic-prompts*. For all keywords existing in some utterance, a set of ordered images was displayed in unison to the speaker prior to the prompt text becoming visible. The purpose of these abstract representations of the keywords was to provide the

speaker with a sub-conscience challenge of the upcoming text thus in effect “priming” the speaker’s mind to attempt a synthesis of the upcoming text. The amount of time that a set of iconic prompts was visible for varied according to the number of keywords existing in the utterance (0.8 to 1.3 seconds). Iconic-prompts were designed to retain a large degree of abstractness and were restricted to simple black and white line drawings except for a few cases where a single colour was introduced. For diversity each keyword had two different iconic-prompts associated with it drawn by two different artists and were presented to the speaker in an alternating manner across utterances. Due to i) not having a one-to-one relationship between image and keyword, ii) the abstractness of the images, and iii) the limited amount of time the set of images was visible, it became very difficult for a speaker to “guess” all of the upcoming keywords correctly or their phrasing.

Iconic-prompts were also used to effectively regulate the implicit word rate of speakers by explicitly limiting their maximum possible utterance rate. The recording of each prompt, consisting of up to a single timed set of four iconic-prompts, followed by the text and read speech, took a minimum of 3-5 seconds of time. Therefore, the number of utterances spoken by the speaker ranged from 12 to 20 per minute. This rate is plausible for speech spoken to M12-M24 infants in an environment where objects, e.g., toys, are visually present and handled. Therefore, prosodic cues for this rate of relaxed speech are captured in the recordings. Without any imposed upper limit for utterance rate, speakers tend to increase their rate when fatigued due to their desire to complete the recording session as soon as possible. In the worst case this may produce a frenzied reading of text that includes an overly strong level of co-articulation or other errors, producing a non-uniform corpus.

5 The computational platform

Louis ten Bosch, Guillaume Aimetti, Kris Demuynck

5.1 Overview

The main computational framework used in the WP5 experiments is a MATLAB implementation of two interacting modules (protagonists), one module modelling the caregiver and the other module modelling the learner. The learner is the most complex module of the two, although the caregiver has become gradually more complex during the second year. The architecture of the learner is inspired by the scheme (memory architecture) presented in Figure 3.3. This architecture was discussed, updated and consolidated during a meeting in Leuven by a task force in which all involved ACORNS partners participated.
The communication between caregiver and learner is specified at a high level in figure 3.1 and 3.2. The details of the modules have been described in Deliverable 5.1.1 (M10) and an update will be described in Deliverable 5.1.2, 5.2.2, and 5.3.2 (due M30). For the sake of clarity, we here mention the most relevant conceptual issues that play a role in caregiver and leaner.

5.1.1 Caregiver

- The caregiver provides stimuli to the learner and reacts to the replies by the learner. The stimuli are chosen according to a predefined list. This list is created by making use of a pool of utterances and depends on the specific experimental design. Whatever the reply from the learner, the caregiver can in principle decide to present the next stimulus, to repeat the previous (or similar) stimulus, or to use a corrective utterance of the kind 'no I mean FISH'. Moreover, the caregiver can be relaxed or more aggressive in its corrective feedback (for example it can insist on correcting incorrect responses). In the Y2 experiments performed so far, the corrective sentences are not explicitly used as corrective feedback – they are just included as regular training and test utterance.
- To make this decision, the caregiver 'listens' to the learner and compares the learner's reply with the ground truth in the stimulus (which is specified in the database). In the case in which tags are used (all Y1 experiments, some Y2 experiments), this comparison is straightforward and very similar to the WER approach in conventional ASR. In the case of features, however, the comparison is much more complex, especially in the case where features are combined to represent a visual scene. In case a trivial feature matrix (e.g. a diagonal matrix) or a simple matrix (in which words and features are related 1-1) is used, the evaluation is straightforward. The exact way of how to compare presented feature vectors and reconstructed feature vectors in the complex case is still under investigation. The challenge is that the learner is presented with a combination with audio + a visual scene, and so in the test it reconstructs on the basis of audio a *visual* scene – and therefore not necessarily the constituent words themselves. In other words, the learner focuses on the compositional meaning of the speech part, rather than on the individual words.

5.1.2 Learner

- The learner receives input stimuli and updates the contents of the sensory store, STM and LTM. It also compares the stimulus with the internal models and then generates a virtual response that is provided to the caregiver. At a high level, the learning algorithm is specified by one of the 'routes' as depicted in Figure 6.1 and the way how stimuli are processed (amount of initialisation material, amount of update material, update frequency and the way perceived stimuli are reused in the update).
- The drive to learn is implemented via a coupling between the external and internal learning loops.

5.2 Integrating DP-ngram into the learner's architecture

The computational framework implements a memory architecture that attempts to achieve cognitive plausibility (cf. Figure 2.3). The details of the implementation of that 'abstract' architecture depend to a large extent on the learning algorithm that is used in a specific experiment. The DP-ngram algorithm has now been modified to work within this ACORNS framework. The memory structure that has been implemented can be seen in Figure 5.1.

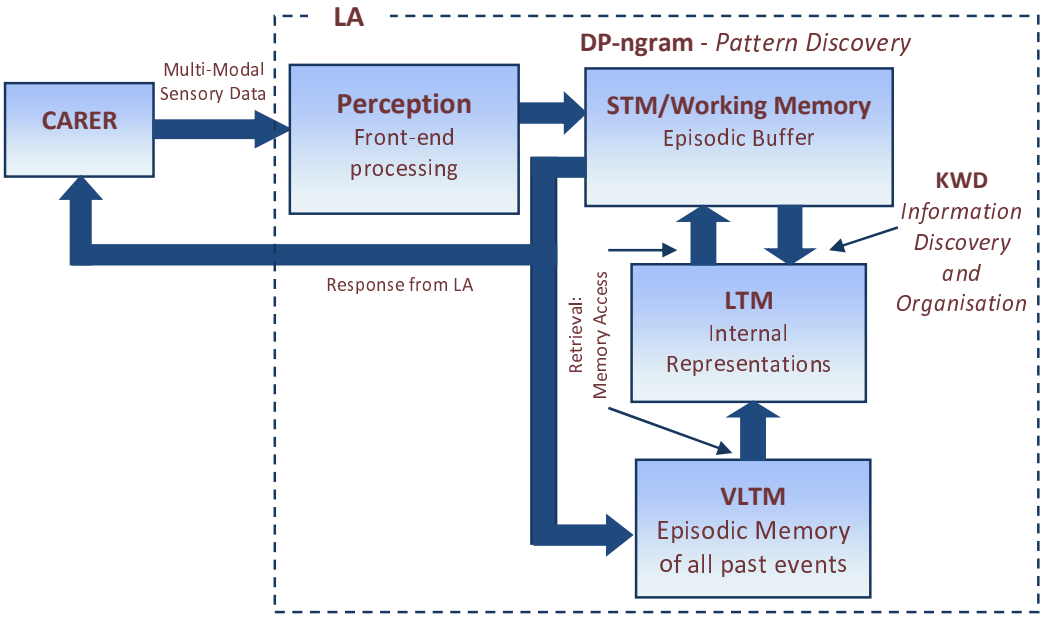

Figure 5.1 Integration of DP-ngram into the learner architecture

Carer – The carer feeds the learner with cross-modal input (acoustic & semantic).
Perception – The stimulus is processed by the 'perception' module which converts the acoustic signal into a representation similar to the human auditory system (mfcc's using ACORNS front-end).
Short Term Memory (STM) – The output of the 'perception' module is stored in a limited STM which acts as a circular buffer to store n past utterances. The n past utterances can then be compared with the current input to discover repeated patterns using DP-ngram.
Long Term Memory (LTM) – The ever increasing list of discovered units for each word representation are stored in the LTM. Clustering processes can then be applied to find the ideal representation. The representations stored within LTM are only pointers to where the segment lies within the very long term memory.
Very Long Term Memory – The very long term memory is used to store every observed utterance. It is important to note that unless there is a pointer for a segment of speech within LTM then the data cannot be retrieved. But, in the future additional 'sleeping' processes could be carried out on the data stored in VLTM to re-organise internal representations or carry out additional analysis.

6 Experiments

6.1 Aims, overview, performance measures

This chapter discuss the main experiments performed during the second year. All experiments described here aim at the 'end-to-end' processing of stimuli – that is, they deal with feature detection, learning (i.e. bootstrapping and updating) internal representations, and the decoding of new stimuli in terms of these internal representations. Experiments dealing with parts of the learning chain, such as the Self Organising Maps (SOM) and the Restricted Boltzmann Machine (RBM) are not discussed here. Work on SOM is described in WP report 3.2 'Report focussing on the results of the initial ASR experiments comparing episodic and semantic long term memory'.

6.1.1 Aim of the experiments

All experiments have the general aim of modelling the discovery of semantic units (word-like units) from multimodal stimuli. Table 6.I provides an overview of the *specific* aim per experiment.

Table 6.I. Overview of the WP5 experiments.

Partner	Aim of experiment
TKK	Based on DB Y1 and Y2: focus on hierarchical representations leading to clear picture to explain how the number of keywords is immaterial for accuracy
SHFD	DP-ngram comparison Y1-Y2 DB, using tags experiment Y1 DB, with visual features from WP5
KUL	Experiments showing that NMF approach works for Y2 (crisp tags, incremental, scoring ordering-independent) + use semantic features Experiments focusing on hierarchical representations
RU	Incremental processing; Semantic features; Construction of hierarchical representations

6.1.2 Overview of the learning algorithms

The computational models used in these experiments are based on different learning techniques and implementations. Figure 6.1 shows at a high-level view three main routes to deal with a speech stimulus.

The signal (left) can be processed via route 1. In this route, emphasis is on a a form of blind segmentation which produces a sequence of symbols that do not have any linguistic or phonetic reference. A subsequent structure discovery method operates on this symbol sequence by searching recurring symbolic sub-sequences. The step from sub-symbolic to symbolic representation is made in an early stage. Computational Mechanics Modelling and the multigram approach are examples of an algorithm along this route.

In route 2, the attempt to unravel the structure in the input signal is dealt with in a different way. Here, the signal is represented as a sequence of (deterministic or statistical) events that may have a phonetic or linguistic interpretation, after which a decomposition technique that operates on sub-symbolic data searches for common structure. NMF is a typical example along this route.

Finally, route 3 uses stimulus-to-stimulus comparison to hypothesize potentially useful building blocks ('chunks') that are common in both stimuli. After this step, hypothesized chunks might get demoted, created or sharpened based on the acoustical evidence in new stimuli. Only at a very late stage,

symbolic labels are attached to these 'chunks'. An approach along this route is the DP-ngram algorithm.

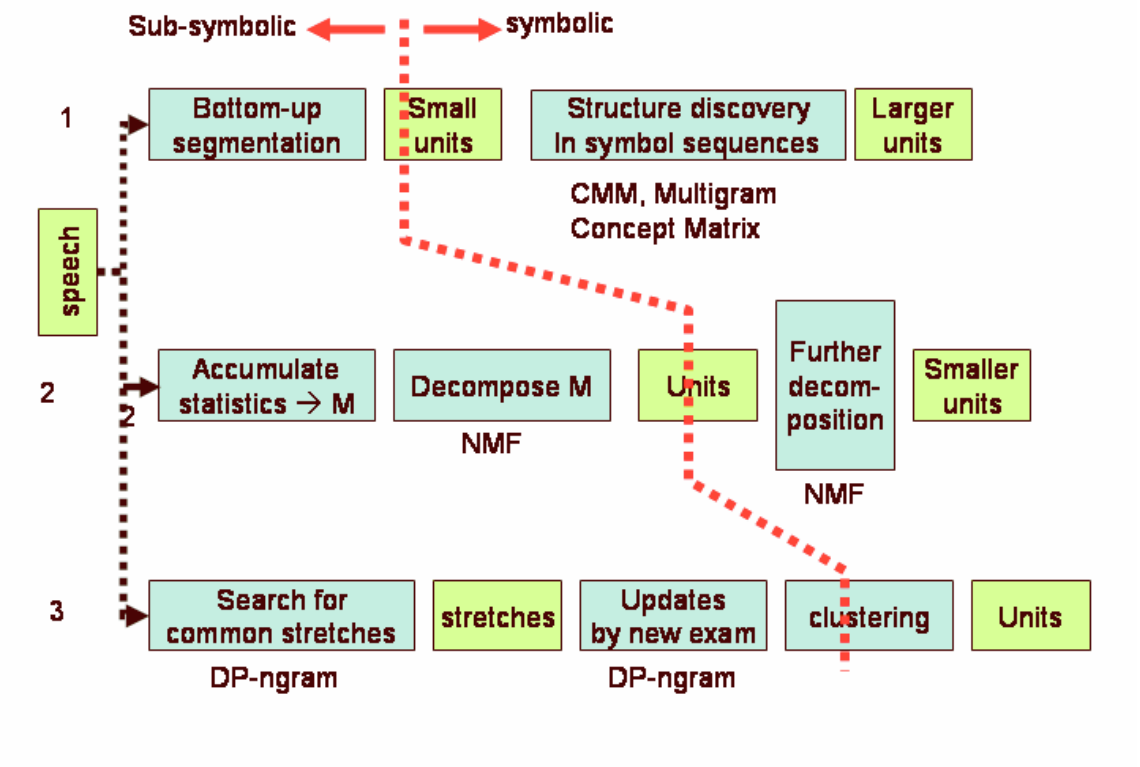

Figure 6.1. High-level overview of structure discovery methods.

6.1.3 Performance measures

Independent of the learning algorithm, several types of performance measures may be distinguished. We here briefly discuss five types.

(1) Comparison of tags (as invariant symbolic representations). This measure is directly comparable to the conventional automatic speech recognition (ASR)-like WER/accuracy measurement. This approach makes use of the fact that each utterance in the ACORNS databases is associated to one or more "concepts" (by the invariant symbolic tags, see chapter 5 of this report). Algorithms that are able to decode speech by providing *ordered* sequences (such as ASR-like graph-decoding techniques) can be judged based on the comparison of ordered sequences. (This includes #del, #ins, #subs etc.). Algorithms that do not provide ordering (such as the conventional NMF) can be judged by comparing the results without taking ordering into account.

(2) Comparison of the original and the reconstructed visual-semantic feature vector. This deals with the comparison between the feature vector as available in the input stimulus, and the one that is reconstructed on the basis of the audio information in the test. If the feature vectors code the words in a 1-1 way, e.g. by a diagonal matrix, (1) is very similar to (2). In more complex cases such as composite scenic feature vectors, one has to specify what kind of comparison is being performed.

(3) Empirical cross entropy. This amounts to $(1/N)\sum P(correct \mid stimulus)$, the sum taken over all N stimuli.

The above measure all are based on the input-output comparison. There are other measures that focus more on the *processing*. Examples are

(4) Comparison of *learning rate* in terms of the rate in which words are learned (i.e. shape of the learning curve). Since learning is a continuous process, internal representations will usually change throughout the entire learning process. We can therefore not define stability in naive terms such as 'invariant under learning processes, even if more learning stimuli would be presented'. However, we can define stability of internal representations in terms of the independence of the initial conditions that were used to bootstrap the training. To that end, we need means for monitoring the contents of the representations. In the experiments, this is possible by looking at the internal representations that are built and updated by the learning algorithm during the learning process.
(5) Comparison of *emergence of structures*. This measure refers to what happens during learning in terms of the organisation of the internal representations. For example, one may look at the minimum number of tokens per 'word' that are necessary to build a 'stable' representation.

Most of experiments provide a tag-comparison based error measure (1) combined with learning curves (5).

6.2 Emergence of word-like units using NMF

Louis ten Bosch, Joris Driesen

6.2.1 Introduction

Data

The training and test data for these experiments are chosen from the Dutch version of the Y2 ACORNS database. The data consists of all utterances (plus meta-information) from four speakers, viz. Els (f), Henk (m), Margot (f) and Peter (m), in this order. Within each speaker, the stimuli have been presented in the same order in which these have been recorded, so matching the ordering of the prompts. These prompts were mostly organised in a way that consecutive utterances have a high likelihood to share a target word. This implies a tendency for the within-speaker stimuli to be presented in word-blocked form. This word-blocked presentation is not strong, however, since in most utterances more than one target word occurs.
For all experiments mentioned in this section, the same sequence of stimuli was used in the same ordering. For reasons related to the character of the individual experiment, some experiments only used the first 2000 utterances, while other experiments use the entire series of 8300 utterances. This is indicated for each experiment.

Features

Each stimulus consists of one wave file in combination with a 'visual' feature vector. In all experiments, the feature vector has been constructed by making a 'scenic' overlay of *all* properties of the target words in the utterance – no matter whether these target words relate to one or more physical objects. For example we consider an utterance such as

'daddy looks at the small green duck'

with tags {daddy small green}. In the NMF experiments described below, an utterance is accompanied with a visual feature vector *w* defined by

$$w = code(\text{'daddy'}) + code(\text{'small'}) + code(\text{'green'})$$

where code('*w*') is the coding for word w as defined in the semantic feature matrix.
If this feature matrix is a trivial matrix, each target word is encoded by a 0-1 basis vector. This means that the between-type distance is constant for each pair of target words, and that the between-token variation is zero (i.e. no visual variation).The feature matrix can also be more complex, such that e.g. hypo- and hyponymy are reflected in the feature coding.

Unless stated otherwise, the experiments reported below are obtained by using a trivial 0-1 feature matrix.
More details about the features and feature encoding are described in chapter 4 of this deliverable.

Parameters
Each training/test is characterized by a number of parameters that define the exact settings of the algorithm. The parameters are presented in table 6.2

Table 6.2. Overview of parameters in the NMF-decoding algorithm.

Name	Meaning	Typical value
minUttNMF	The number of stimuli necessary to bootstrap the training	From 100 on. A value of 10 is certainly too low. 100 utterances are equivalent with approx. 5 acoustic realizations/word.
STMlength	The amount of stimuli than can be stored in short-term memory. It is used for the internal update of representations.	300-700 utterances. This amounts to about 25-30 acoustic realizations per word.
update_freq	The frequency of updating the internal representations.	This value has been set to 1 (without further experimentation)
nr_innerloops_init	The number of internal loops to create initial models.	varying
nr_innerloops_update	The number of internal loops to *update* internal models.	For NMF, 1 appears sufficient
semantics_factor	The weighting factor that weights the influence of the visual/semantic part of a stimulus compared to the visual part.	(must be in the order of a few 100 at least, to compensate for the small size of the visual feature vector compared to the size of the acoustic encoding)
nr_stim	Number of stimuli used for a training.	

The values of the parameters are presented as legend in the figures. The ordering is the same as presented in table 6.2:

```
minUttNMF-STMlength-update_freq-nr_innerloops_init-nr_innerloops_update-
                        semantics_factor-nr_stim
```

6.2.2 Results

A typical learning curve for the Y2 database is presented in figure 6.2.1. The horizontal axis presents the number of multimodal stimuli presented. The vertical axis presents the accuracy of the learner, linearly averaged over the most recent 50 utterances. The accuracy of the learner per utterance is the number of target words that has correctly been identified, compared to the number of target words as specified in the stimulus. If there is one target word, such as in the utterance ‘nee ik zei vis’ (no I said

fish), the accuracy is 0 (incorrect) or 1 (correct). If there are three target words, the accuracy might be 0, 1/3, 2/3, or 1. The vertical bars around x=2100, 4200 and 6300 denote a new speaker. The major dips of about 30 percent absolute every time a new speaker starts indicate that the internal representations have to adapt to the new speaker. In other words, internal representations are (at least to a certain extent) dependent on the acoustic characteristics of the 'most recently observed' speaker. Compared to the dips observed in the Y1 database (15 percent absolute), the Y2 dips are about twice as deep.

Not all dips are attributable to a speaker change. Some of the dips are due to the fact that within a speaker, stimuli are presented in work-blocked fashion (due to the ordering of the prompt sheets that were used in the Y2 database recording). Below, we will give an example of a 'speaker blocked words random' stimulus presentation.

The number of internal update loops can be reduced to 1 – without sacrificing overall learning rates. This is shown in figure 6.2.2 (1 loop), in comparison to figure 6.2.1 (4 loops).

Figure 6.2.3 shows that the minimal number of stimuli seen before initialisation can be reduced to 100 (that is, about 5 acoustic realisations per word). A too aggressive initialisation deteriorates the performance substantially (figure 6.2.4 and 6.2.5).

Comparison between figure 6.2.6 (word-blocked) and 6.2.7 (words randomised) shows that in the case of random word ordering the dips in the learning curve are attributable to speaker changes only. This explains the spiky character of most plots: there is an effect of the word-blocked presentation of stimuli on the representations during the learning process.

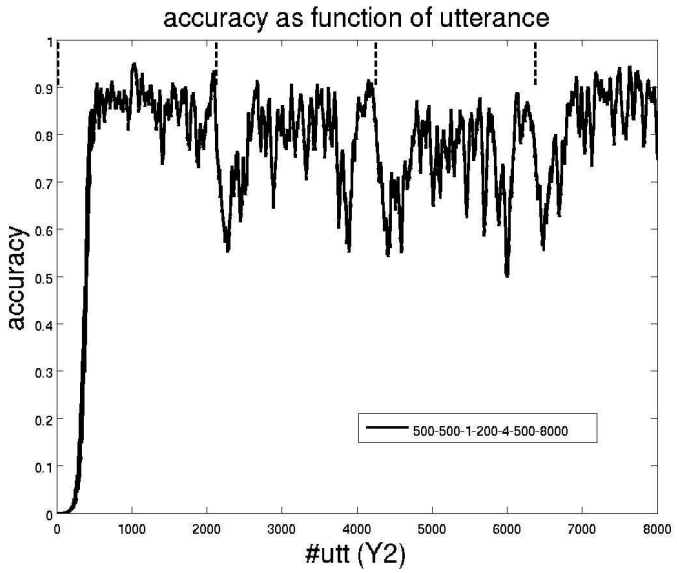

Figure 6.2.1. A learning curve, obtained on the Y2 database. Stimuli are presented speaker-blocked wise.

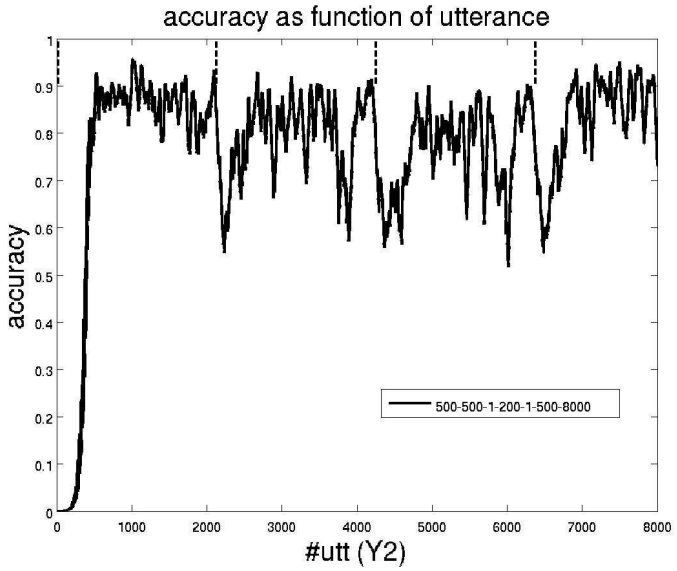

Figure 6.2.2. This figure presents the same experiment (fig. 6.2.1), the difference being the number of inner loops in the NMF update, which was 4 in the previous plot and equals 1 in this training. The tiny differences indicate that once the models are properly initialised, the internal update might be very shallow.

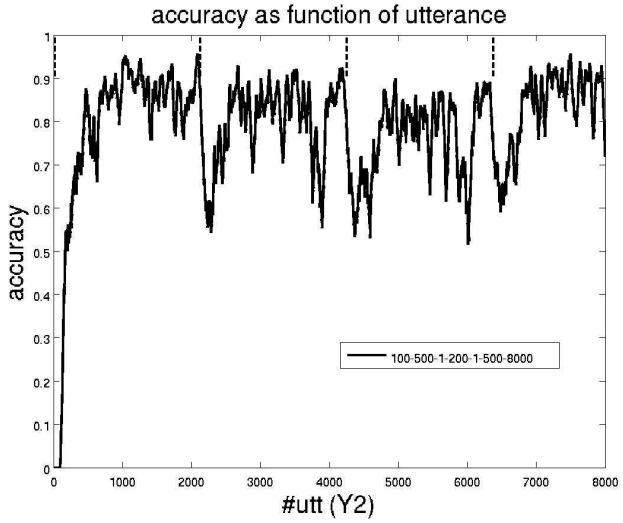

Figure 6.2.3. Same as previous figure, but now NMF builds representations much earlier in the training process, from stimulus number 100. This means that the bootstrap of the internal representations is based on only about 5 examples per word (average).

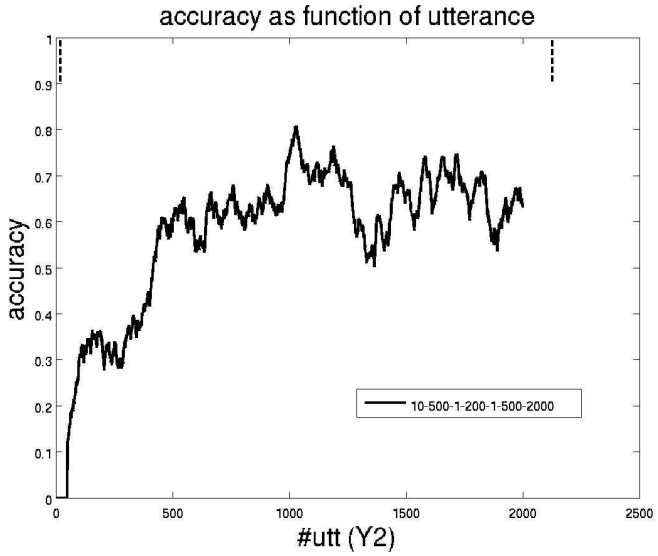

Figure 6.2.4. Results of a training in which very few utterances (10) are used for initialising the internal representations. In comparison with the previous figure, this result shows that once the learner attempts to bootstrap from only 10 utterances, the long-term performance deteriorates. This is due to the initialisation problem starting from too few stimuli (much less than the total number of keywords). For the sake of clarity, this plot shows the result for the first 2000 utterances.

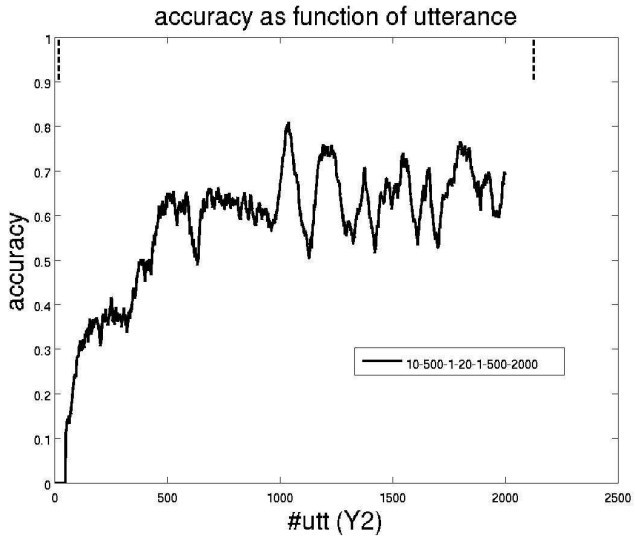

Figure 6.2.5. This plot differs from the previous plot in the lower number of initialisation loops (was 200, here 20). The differences are in the details; globally, however, there is no deterioration compared to the previous figure. It indicates that the bad initialisation is not due to over-training on a too small data set.

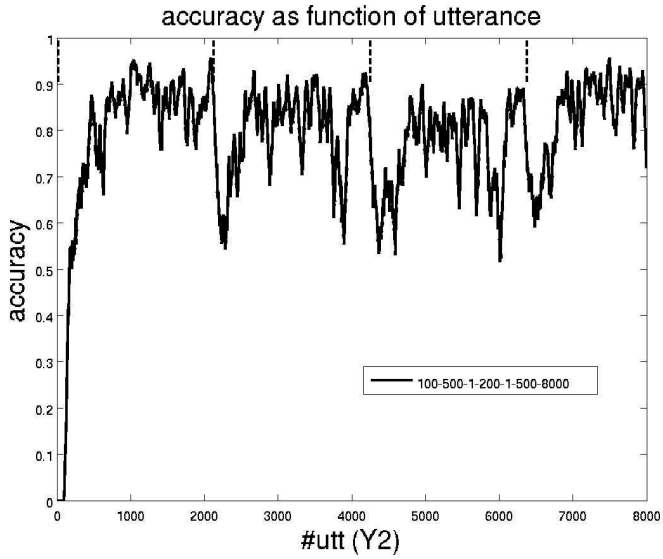

Fig 6.2.6. Same plot as figure 6.2.2. Copied here to allow easy comparison with the next plot.

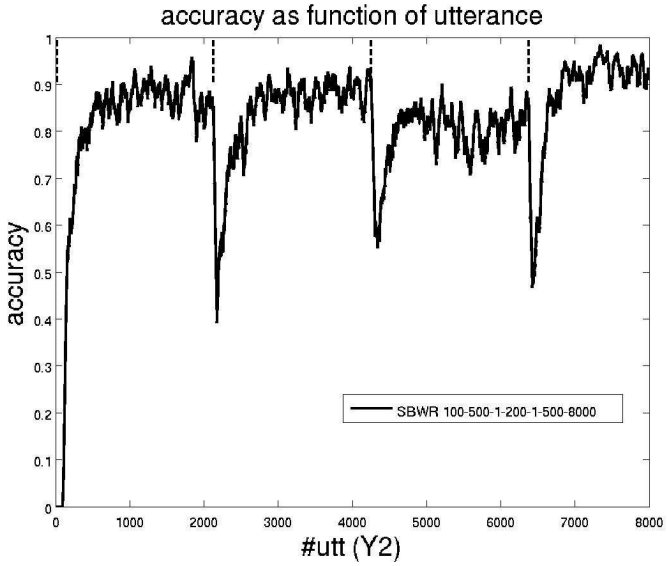

Figure 6.2.7. As figure 7.2.6, but with word order randomised. The dips are now attributable to speaker changes only. To be compared with the previous plot, in which the utterances were presented word-blocked.

6.2.3 Discussion

The experiments using NMF on the current Y2 Dutch database are encouraging. They show that the word discovery approach is able to build internal representations of meaningful units, also in the case of a database more complex than the Y1 database. The representations emerge during training. As the results show, the performance of NMF-based learning depends on various settings. Most important parameter is the size of the internal buffer, used to update internal representations. It must be large enough to have about 500 utterances in memory, which amounts to approximately 30 acoustic realisations per target word. At most about 5 acoustic realisations per word are necessary for robust initialisation.

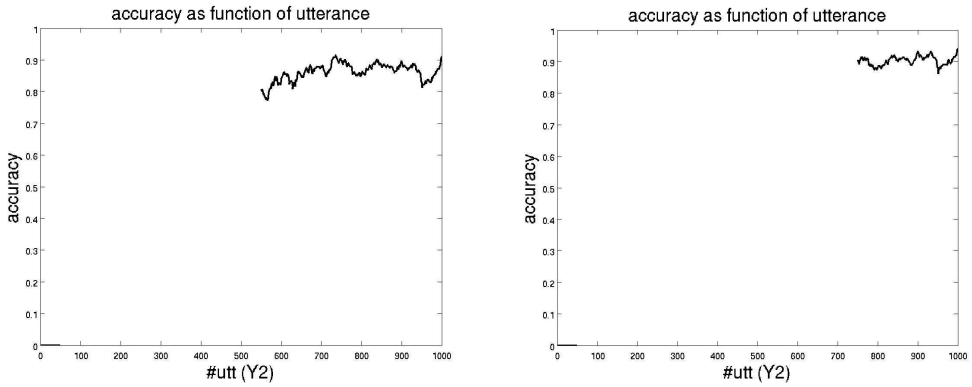

Figure 6.2.8. These plots indicate the performance of the learner if the learner waits with the initialisation until 500 (left) or 700 (right) stimuli have been observed. Comparison suggests that the performance can be improved by a better initialisation of the internal representations, or by a better use of the information that is conveyed in the first few hundreds of utterances.

6.2.4 Plans for NMF-based learning experiments for the third year

In the third year, at least the following issues will be focused on.

(1) The emergence of hierarchical representations. The first experiment in this direction will start with building speaker-dependent models as the first step in the modelling process. A subsequent clustering step will then generate hypotheses about speaker-independent word clusters (same technique as used in ten Bosch et al., 2008). Also the experiments with hierarchy based on cascaded two-layered NMF (deliverable D4.1, section 5) will focus on the hierarchical structure of internal representations.
(2) The flagging of the activation of internal representations. During the training in the NMF-based experiments described above, the learner makes use of the fact that the number of representations per stimulus is known. That is, for each stimulus, the learner knows how many target words (representations) must be looked for among the activated representations. This is a cognitively implausible situation. Recently, experiments have been performed that select the 'set of most active representations' by using an activation threshold (Del D4.1, section 4).
(3) The use of more realistic features. The experiments described above avoid the use of the invariant symbolic tags, but still use a 'trivial' feature matrix.

6.3 Concept Matrices

Okko Räsänen, Unto K. Laine, Toomas Altosaar

6.3.1 Introduction

The Y2 ACORNS speech material is richer than the Y1 database (50 instead of 10 keywords). Also more sentences with multiple keywords are present. This evokes the question how to select the methods so that we can cope with this more complicated speech material.

How are the keywords discovered if they always appear in a complex context of other keywords? One simple answer to this problem is cross-situational statistics (see, e.g., Smith & Yu, 2007). Co-occurring inputs and/or events form associations. Things that always occur together are processed as a one large associative chunk since there is no reason to split the representations (e.g., co-occurrence of

a word *dragonfly* and the corresponding insect). However, if a naive learner hears the pairs "*flying bat*", "*flying bird*", "*black bat*", etc., the situation changes. Let's assume that we do not know any words and our conceptual thinking is not (yet) dominated by linguistic structures. By first hearing the pair "*flying bat*" and seeing a dark winged animal moving in the air, we may form an association between an acoustic word form "*flyingbat*" and the perceived event, or, if we are able to separate doing from being, an association between "*flyingbat*" and the insect (similarly to a *dragonfly* example) is formed. However, as soon as we hear "*flying bird*" and there is nothing related to the bat-insect in our perceptual surroundings but a bird instead, we find a common link between both occurrences of the word *flying:* a subject moving in the air. The statistical binding between the word *flying* and perceiving something flying becomes stronger than the connection between the bat-insect and flying-doing due to the frequency of occurrences of these associations. Especially, if we then hear "*black bat*" it also becomes very clear what the word "*bat*" is representing.

In a nutshell, the idea is very simple: what systematically co-occurs forms an association. Associative agents (words, visual shapes, etc.) that are connected to each other can be split into two or more new agents if the sub-parts of the agents form strong associations to something else or seem to occur everywhere equally often in other contexts. From the language learning point of view this means that the richness of the language input is actually necessary for finding good compact models for words since changes in the word context will lead to splitting of models. However, not all word-to-word connections fade to a non-existent level unless the input is composed of entirely randomised non-syntactic and semantically incoherent speech. If the language input to the learner represents properties of a real language, the systematic properties (co-occurrences) that are left for the word context after massive amounts of exposure to the language lead to the formation of associative links that represent semantic and syntactic properties of a language (cf., e.g., Latent Semantic Analysis, Landauer & Dumais, 1997). To conclude, the number of words in an utterance is not an issue, as long as there is a sufficient amount of language input where the constituents occur in other contexts. What is sufficient, then, is dependent on the way the learner is processing the statistical information. As the language learner manages to learn meaning for a handful of words, the formation of accurate models for novel words becomes much more easier as the number of new words and their possible referents in each utterance becomes smaller. The correct question to ask is: "*How often does this event occur in this context in relation of this occurring in other contexts*?"

6.3.2 Recognition with multiple keywords per utterance

This section discusses the effects of the detection of multiple target patterns in the input when the models for patterns have been already learned (see deliverable D2.2, Association Response Table, section 2.5.3).

In order to do word recognition from a speech stream it is necessary to have some sort of recogniser, or model, in the memory of the system for the words that are being searched for. When a new signal is presented to the system, the system analyzes structures of the signal and matches them to the existing models. The models giving a good match form a set of word candidates (cf. "*cohort*", Marslen-Wilson & Tyler, 1980 or TRACE model of speech perception, McLelland & Elman, 1986). In the simplest case, the word hypothesis is chosen by a pure bottom-up lexical competition, that is, the winning word model is the one gaining the most activation from acoustics and contextual and syntactic information is either already embedded in the word model or is neglected.

The matching process is essentially a *temporal* process: activation of word models can be analysed quantitatively at any moment of time using information from the audio stream inside a limited temporal distance. However, looking at a single time instant is not sufficient for determining what word is occurring at that time since acoustic representations of spoken words are distributed temporally over several hundred milliseconds. Gathering and analysing this distributed information requires continuous integration of word-unit activations over temporal windows of several hundred milliseconds. Temporal integration leads to a mean activation level, or a cumulative activation level (depending on how the integration is performed) for each activated word candidate at each moment of

time. This type of temporal activation of each model can be then used to inhibit all other models based on their activation level if found necessary. The word model gaining the most activation is chosen as the detected word.

Now if the size of the lexicon increases, the acoustic distances of the word models become inevitably smaller. This makes competition harder as more and more word candidates gain activation from the bottom-up stream. Therefore, the number of keywords is an issue for the correct detection of the word *at a specific temporal location in the speech stream.*

However, in the word recognition task, multiple keywords in a single utterance exist in a serial form. The only way they affect each other is that they have a context effect on the neighbouring words with variable significance. However, the context is always present, no matter how many "tags" or target words there are in the utterance (unless there is no real sentence but just a single word, but even then the isolation of a single word affects the pronunciation of it). Having two known words adjacent to each other simply leads to a situation where the activity of another word ends and the activity of another word begins. Actually, the activation of the words is more accurately defined if adjacent words are keywords, that is, they are already familiar to the recogniser. This is because the activation of the subsequent word will override possible activations of the other recognisers that may arise from, e.g., syllable structure that is shared between words.

Consider, e.g., the Finnish utterance "*Hän antaa likaisen lehmän*", where the transition from /an/-/ta:/ to /li/-/kai/ shares a syllable structure with the keyword "*talitintti*" (/ta/ /li/, figure 6.3.1). The talitintti-recogniser becomes activated at the transition from *"antaa"* to *"likainen"* but as it has only one syllable that is shared with the first word and one syllable with the second word, its overall temporal activity falls to a lower level than the actual words spoken (figure 6.3.2) and is thus inhibited (figure 6.3.1, bottom).

A more ambiguous situation would occur with compound words and with words that have some other word as their sub-part. As there are currently no such words in the Y2 Finnish corpus, these situations cannot be tested empirically. However, we can hypothesize what would happen in this case: if the window of temporal integration is sufficiently long, the longest model will preside, as it will cumulate the most activation over time. In case of compound words the situation will also depend on whether they are modelled as two separate words and whether an association is made somewhere during semantic level analysis.

Figure 6.3.3 demonstrates the keyword recognition accuracy of the transition probability analysis as a function of number of keywords embedded in the test utterances. Bars on the left show the recognition rates when the same number of word hypotheses are permitted as the number of meta-tags in the utterance (chosen in order of total activity). As can be seen, recognition rates for utterances with only a single word turn out to be the most difficult. There is also a slight trend for decrease in accuracy as the number of keywords increases, but the recognition rates are still relatively high and since the possibility for ambiguities increases significantly as the utterance lengths increase, it would be hasty to say that this is because there are several keywords in the utterance.

The problem with single keyword utterances is probably due to the fact that other recognizers react to familiar structures in carrier sentences and therefore some other words may cumulate sufficient activity to surpass the actual keyword. For example, the utterance "*Ei, minä tarkoitin isiä!*" ("*No, I meant daddy!*"), where "*isi*" is the tagged keyword, there may exist several word hypotheses for the carrier sentence (that is much more longer than the keyword itself - these type of corrective sentences should be actually tested in a more natural way by implementing actual dialogue settings which they are intended for). It does not, however, mean that the word model wouldn't be activated at the time it is present in the stream. On the contrary, all utterances with two keywords in the Y2 corpus have the same carrier sentence ("*Where is the* <keyword1> <keyword2>") and therefore the additional words do not affect the detection of the tagged words, leading to very good recognition accuracy.

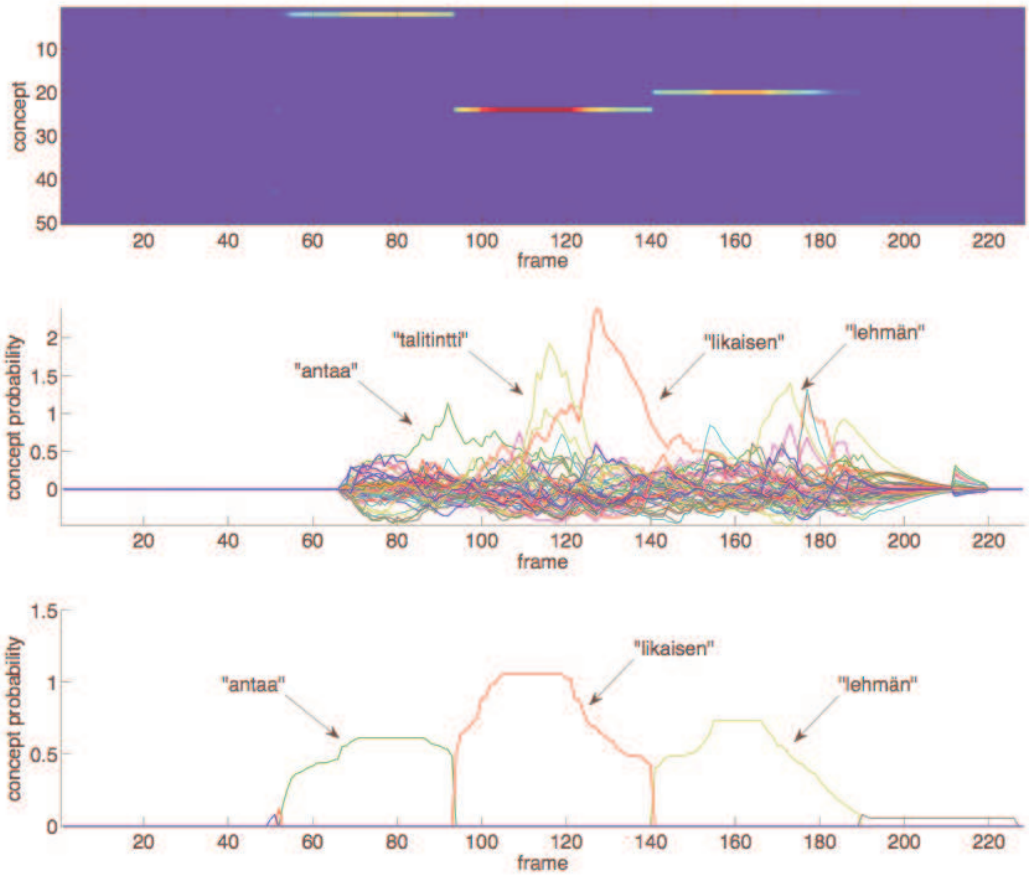

Figure 6.3.1: *"Hän antaa likaisen lehmän"*. The word "*talitintti*" becomes activated at the transition point between "*antaa*" and "*likainen*" but is inhibited by higher temporal overall activity of the neighboring known words.

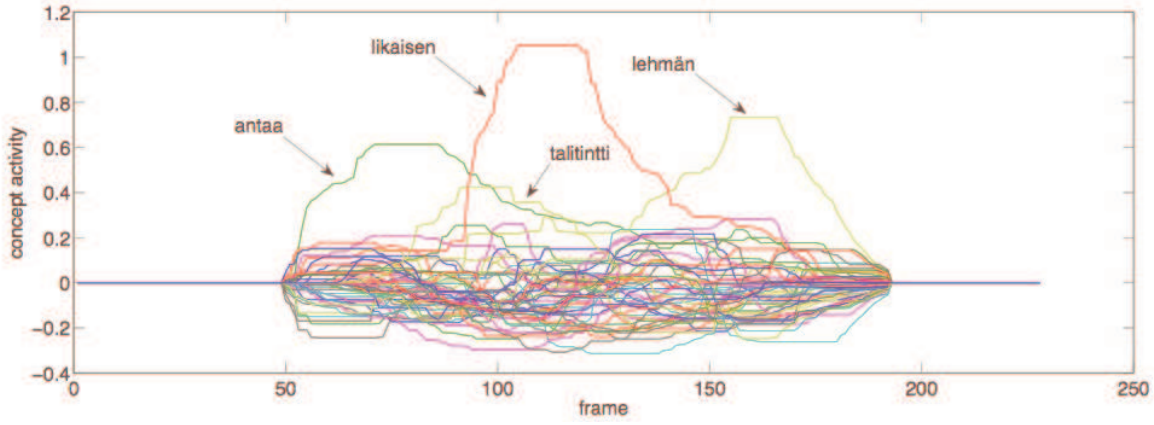

Figure 6.3.2: Recognizer activities for utterance *"Hän antaa likaisen lehmän"* after temporal integration (250 ms median filter window) and before inhibition. The talitintti-recognizer activity falls below other (correct) word candidates.

To overcome the problem with the evaluation method in the case of single keyword utterances, the recogniser was allowed to give N+2 word hypotheses for N keywords. Now the utterances with only one keyword are recognized with much more similar accuracy compared to the other utterance types, which at least partially proves the above hypothesis (figure 6.3.3, on the right).

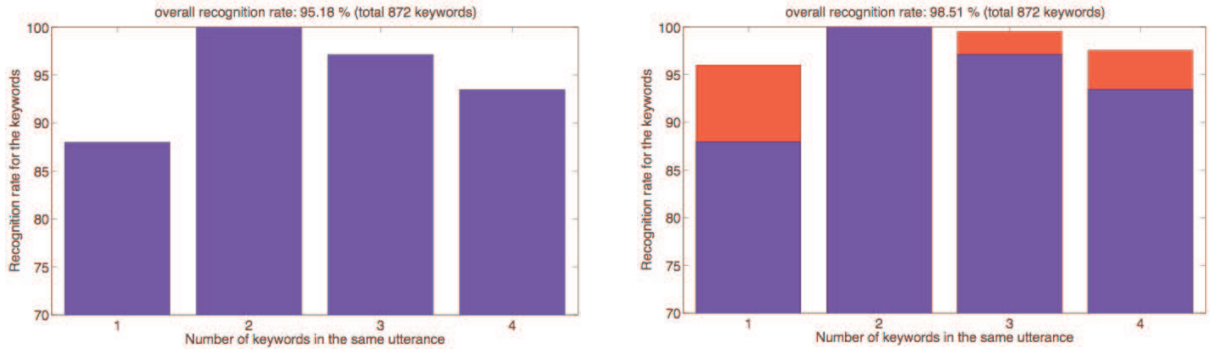

Figure 6.3.3: Keyword recognition rates for utterances containing 1, 2, 3, and 4 keywords when N (left) and N+2 (right) word hypotheses are allowed, where N is the number of utterance related keyword tags (increase from N to N+2 shown in red).

Figures 6.3.4 and 6.3.5 show some recognition examples for utterances with 4 and 3 keywords, respectively. The effect of multiple adjacent keywords can be easily seen as well-defined transitions from one word to another. On the contrary, figure 6.3.6 shows the situation for the utterance "*Tuolla on syötävä lintu ja auto.*" where the conjunction "*ja*" ("*and*") does not have a word model and causes an ambiguous situation between the last and the second to last word.

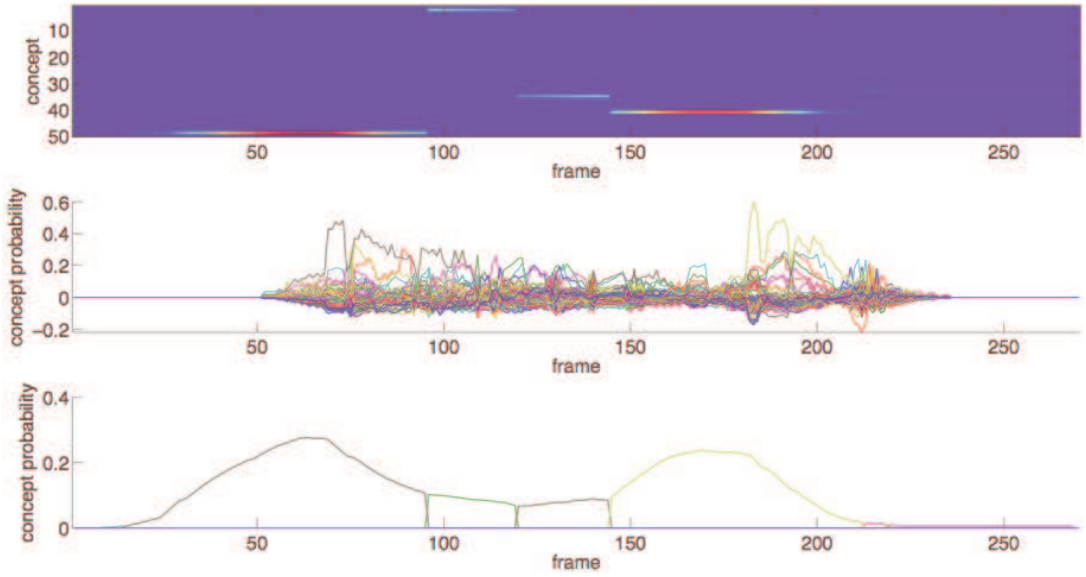

Figure 6.3.4: Recognition of the utterance "*Vauva antaa pienen puun.*". Recognizer activities for all 50 keywords are shown in the middle trace. Activities after temporal integration and inhibition are shown at the bottom and the association response table (ART) derived from the filtered representation is shown at the top.

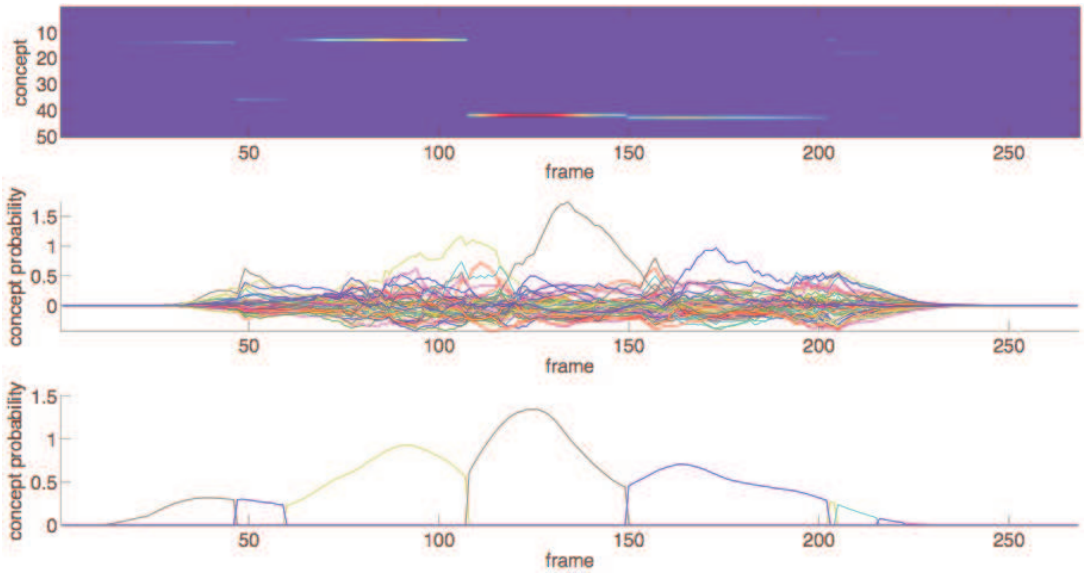

Figure 6.3.5: Recognition of the utterance "*Hän katsoo pyöreää rekkaa.*". Recognizer activities for all 50 keywords are shown in the middle trace. Activities after temporal integration and inhibition are shown at the bottom and the association response table (ART) derived from the filtered representation is shown at the top.

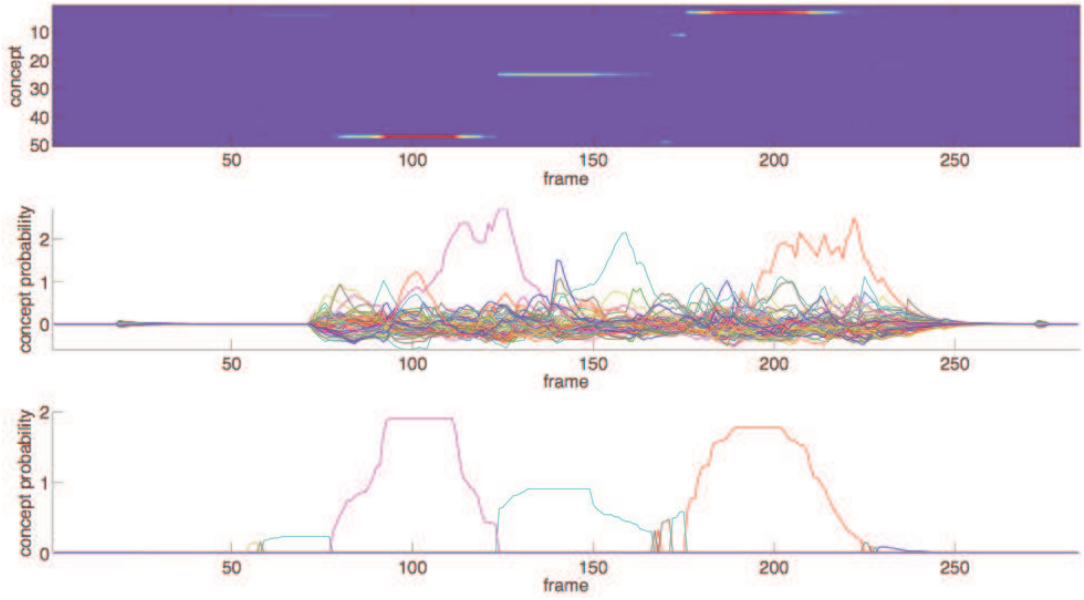

Figure 6.3.6: Recognition of the utterance "*Tuolla on syötävä lintu ja auto*". The conjunction word "*ja*" is not known to the system and this causes an ill-defined situation in the activity between the last and the second to last word.

6.4 Acoustic DP-ngrams - Word Discovery Experiments

Guillaume Aimetti and Roger Moore

6.4.1 Aim of the experiments

The experiments that are discussed in this chapter focus on the use of DP-ngram technique. In figure 7.1, this route is represented by the lowest branch. The experiments have been carried out to test the Acoustic DP-ngrams ability to automatically discover and build internal representations of key words from a sub-set of the entire ACORNS (Y1 and Y2) database. The algorithm is able to associate co-occurring events across multiple modalities (acoustic and semantic) to create internal representations of its surrounding environment. These experiments show how quickly the algorithm is able to create stable internal representations of the key words that LA is required to learn from the Y1 and Y2 database, and how the complexity of cross-modal input affects the word learning rate.

6.4.2 Acoustic DP-ngrams

There are two key processes to the language acquisition model described here; automatic segmentation and word discovery. The automatic segmentation stage allows the system to build a library of similar repeating speech fragments directly from the acoustic signal. The second stage groups these fragments into distinct key word classes.
The automatic segmentation process accommodates temporal distortion through dynamic time warping (DTW). The algorithm finds partial matches, portions that are similar but not necessarily identical, taking into account noise, speed and different pronunciations of the speech. Traditional template based speech recognition algorithms using DP would compare two sequences, the input speech vectors and a word template, penalising insertions, deletions and substitutions with negative scores. Instead, the acoustic DP-ngram model uses an accumulative quality score to reward matches and prevent anything else; resulting in longer, more meaningful sub-sequences.
Word discovery is carried out through a simple cross-modal association process. The algorithm learns by comparing two utterances; thus, the algorithm is able to create internal classes of co-occurring cross-modal (audio & semantic) events. Each internal class evolves throughout learning, building an ever increasing list of episodic acoustic units.

6.4.3 Experiment DP-ngrams 1 – Word Learning Rate

The first experiment is a comparison of the word learning rate between the Y1 and Y2 data set. The stability of LA's internal representations will be continuously measured during the learning process as the internal representations are constantly evolving.
There are 10 key words within the first year database and 50 key words within the second. LA is tested on a prototype exemplar of each key word and must reply with the correct semantic tag. It is important to note that LA may reply with more than one semantic tag as her internal class could be represented by multiple semantic features. The acoustic DP-ngram algorithm does not assume that each semantic feature is a single object, for example if LA always observes a green frog then its internal representation will be a single concept with the semantic features 'green' and 'frog' where the acoustic association would be 'greenfrog'. It's only if 'green' occurs separately that LA will create a 'green' concept. Therefore, if LA is tested on the key word 'frog' and replies with the semantic tags 'green' and 'frog' then she is penalized as having a distorted internal representation. A second, more lenient, measure will also be recorded, showing the total number of words learnt when allowing for internal semantic distortion.
LA was tested on a sub-set of 200 utterances from the Y1 and Y2 database of the same speaker.

Results

Figures 6.4.1 and 6.4.2 display the stability of LA's internal representations of the key words from the year 1 and year 2 database as a function of observed utterances. In both figures, the x-axis shows the number of utterances observed and the y-axis identifies the index of the key word LA was tested on. Areas of the plot in blue indicate that the key word has not yet been observed. Areas of the plot in green indicate that LA has observed the key word but does not have a correct internal representation of

it. The different shades of red indicate that LA has a representation of the key word, the darker the shade the more stable the representation. A distorted representation implies that LA has replied with an internal representation that contains multiple semantic tags.

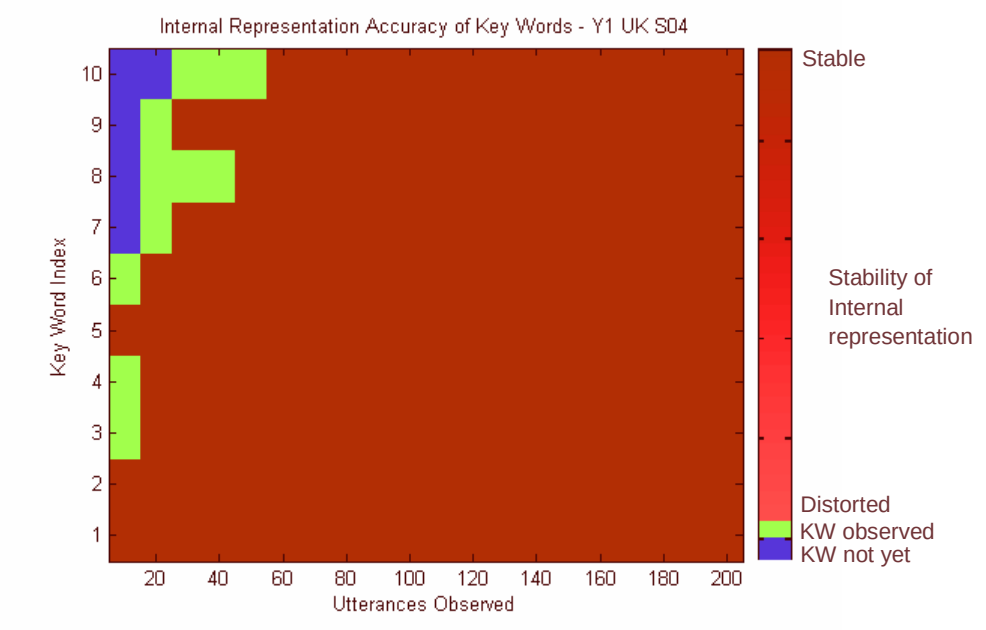

Figure 6.4.1. Internal representation stability of the 10 key words of the ACORNS year 1 UK database.

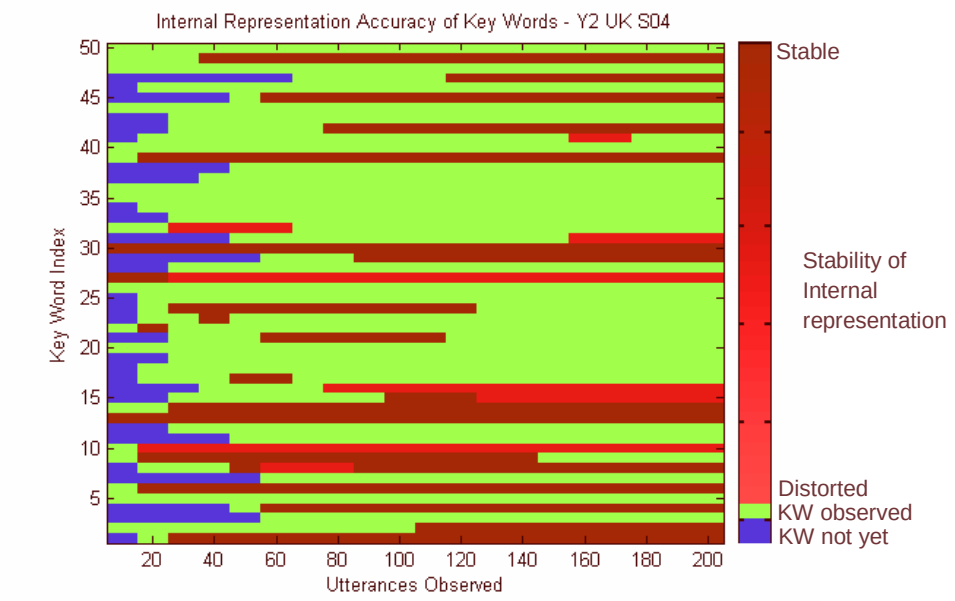

Figure 6.4.2. Internal representation stability of the 50 key words of the ACORNS year 2 UK database.

Figure 6.4.3 is a plot of the number of stable words LA has learnt during the training/test phase (utterances 1 - 300) on the Y2 database. The x-axis shows the number of utterances observed and the y-axis shows the number of LA's stable internal representations of the key words from both data-sets. The red and blue plots display LA observing year 1 utterances and the black plot year 1 utterances. It can be seen that LA achieves stable word representations for all key words within the year 1 data-set

after 70 utterances. LA seems to reach a plateau of 16-18 stable *clean* key word representations after 120 utterances and almost 30 if allowing for semantic distortion. This is still short of the total 50 key words required to be learnt from the year 2 database.

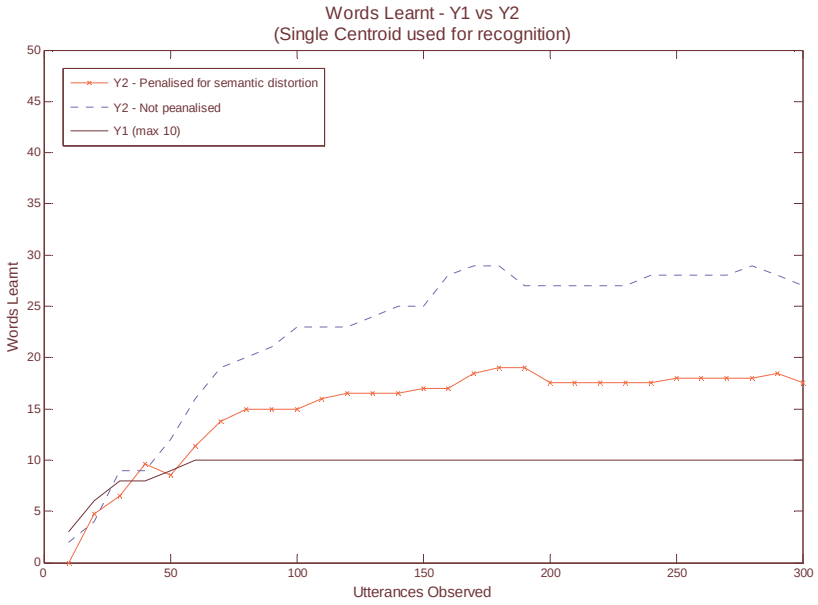

Figure 6.4.3. Comparison of word learning rate as a function of observed utterances for year 1 and year 2 data-sets. The two plots for the year 2 database show word learning rates with and without penalising LA for semantic distortion.

6.4.4 Experiment DP-ngrams 2 – Key Word Detection

The second experiment is a simple tag comparison task. During the learning process LA is tested on her ability to detect the keywords/concepts that are present within each utterance. The correct response is for LA to predict the semantic tags associated with the current incoming utterance while only observing the speech signal. LA re-uses the acoustic DP-ngram algorithm to solve this task in a similar manner to traditional DP template based speech recognition. The recognition process is carried out by comparing exemplars, of discovered key words, against the current incoming utterance and calculating an accumulative quality distance score.
Word detection of the first year data-set was a simple task as the algorithm would choose the exemplar unit giving the highest quality score. The second year data-set contains multiple key words. Tag detection is then carried out by setting a quality score threshold for each key word class (see figure 6.4.4).

Figure 6.4.4 displays the quality scores for each of LA's internal representations against the current utterance ('***Mummy looks*** *at the* ***big lion***'). The quality threshold is set to 100, therefore any internal classes that achieve a score greater than the threshold are predicted to lie within the utterance. The tags associated with each class are shown in the legend of the figure.
The model, in its current state, does not make any assumption of the correct attribution of the key words detected within the utterance. LA was tested on a sub-set of 200 utterances from the Y1 and Y2 database of the same speaker.

Utterance = '***Mummy looks*** *at the* ***big lion***'

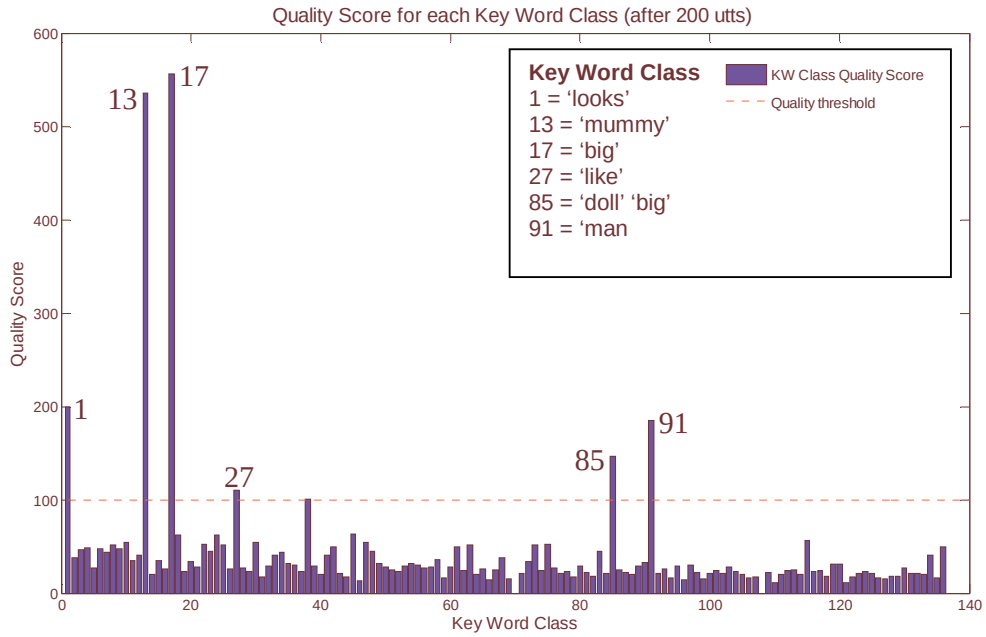

Figure 6.4.4. Quality scores for each of LA's internal representation class. Classes producing quality scores higher than the threshold are predicted to lie within the utterance.

Result

Figure 6.4.5 displays LA's word detection accuracy during the training/test phase. The x-axis shows the number of utterances observed and the y-axis shows the word detection accuracy as a function of utterances observed. The red, green and blue plots display LA observing year 2 utterances and the black plot year 1 utterances. It is evident that LA's word detection accuracy is considerably lower with the year 2 data-set. This is to be expected as the task is much harder. From the plot it can also be seen that using all the exemplars within internal classes is more accurate than using fewer cluster 'centroids' after an initial learning phase (70+ utterances). This shows us that this method of recognition is able to capture acoustic variation more reliably, but at the expense of computation increasing to infinity. A solution to this problem would be to move from template recognition to a statistical model for the internal class at the point where accuracy begins to flatten off or decline (~160 utterances).

When carrying out recognition, LA can use all discovered exemplars stored within each internal class (red and black plot), a single, most ideal, 'centroid' representation of each class (blue plot) or the 'centroid' of multiple sub-clusters within each class (green plot). The sub-clusters for each internal representation class were calculated using an agglomerative clustering method and setting a cut-off distance (see fig. 6.4.6).

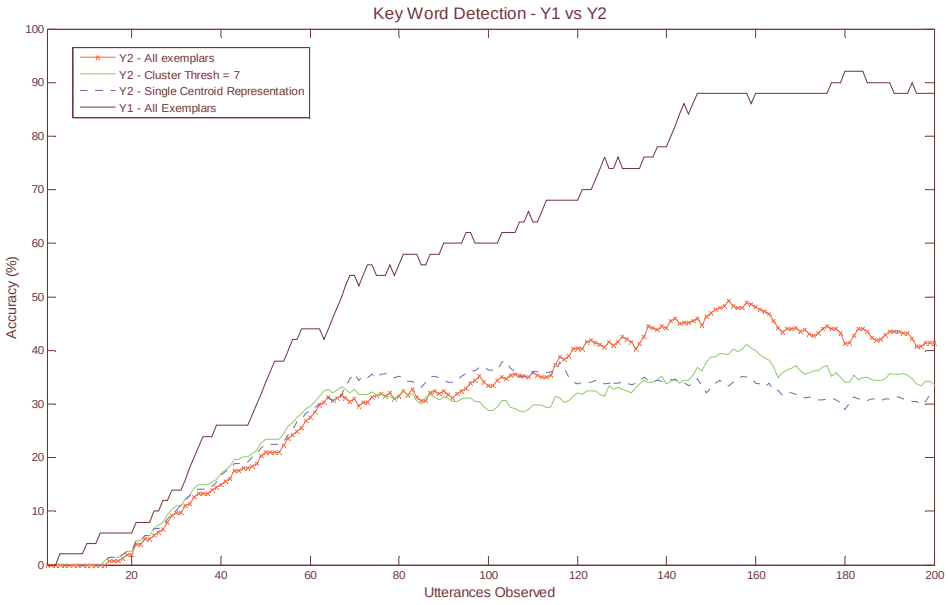

Figure 6.4.5. Comparison of word detection accuracy as a function of observed utterances for year 1 and year 2 data-sets.

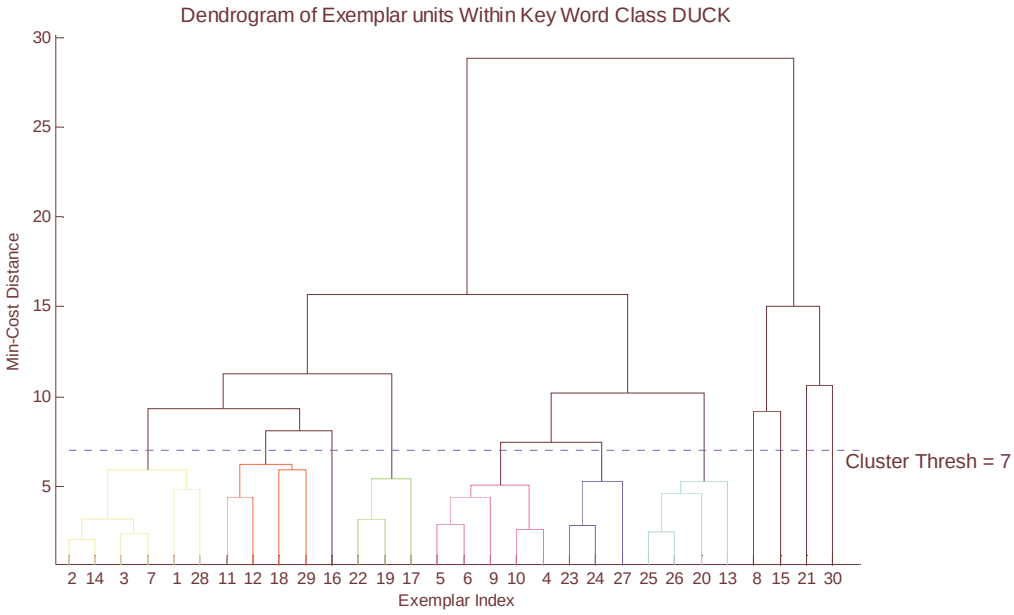

Figure 6.4.6. Dendrogram of the exemplar units within LA's internal representation of the semantic feature DUCK. Sub-clusters were achieved by applying a cut-off distance (example = 7).

6.4.5 Conclusions

Results from the first experiment using DP-ngrams show that, in its current state, the acoustic DP-ngram method successfully creates the required number stable key word representations for the year 1

data-set, but falls short with the year 2 data-set. However, it can be seen from figure 7.4.3 that the model has a faster word learning rate with the year 2 data-set. This may be an advantage of a more complex utterance structure containing multiple key words that occur more frequently.

From experiment DP-ngrams 2 we can conclude that the models word detection accuracy suffers from this additional complexity. The main reason for this low accuracy could be because the algorithm has not, as yet, been optimised for the year 2 data-set. The bar chart in figure 6.4.4 displays the quality scores produced by each key word class, therefore giving a probability of its occurrence within the utterance. With a quality threshold of 100 (as set for experiment 2) we can see that additional and unwanted classes are hypothesised, therefore penalising the learner's (LA's) word detection accuracy.

Factors affecting the results of both experiments could be due to the algorithm, in its current state, not cleaning up distorted internal classes that were created during the early stages of development and not being able to run on a larger data-set.

LA could only be tested on a very small sub-set of the year 1 and year 2 databases due to the ever increasing number of exemplar units discovered and stored within LA's internal memory architecture. In order solve this issue the algorithm is being modified to update and clean up unwanted exemplar units. The exemplars that are left can then be averaged to create a single abstract 'prototype' unit that statistically models the variance within each internal class. Allowing the model to carry out both 'exemplar' and 'prototype' recognition would concur with current developmental theories that young infants and adults use both depending on environmental conditions (Quinn and Eimas, 1996).

7 Relations between WP5 and the other work packages

The experiments done in WP5 by all partners all feed back to the design and approach followed in the experiments in the other work packages. This chapter briefly discusses this feedback.

7.1 WP1 Signal representations

In ACORNS WP1 takes a somewhat special position due to the fact that it provides the input for all learning algorithms and experiments. The performance of a hierarchical pattern recognition system such as developed in ACORNS is dependent on the information conveyed to the higher levels by the lowest level. The principle of ACORNS is to learn verbal communication skills using the human learning process as primary example. This implies that the set of acoustic features that are computed at the lowest level of the hierarchy should basically provide the same information as the output from the human auditory periphery. As a consequence, in contrast to the conventional approaches, the resulting set of features is not a priori dependent on specific design choices made for pattern recognition systems. Thus, at least in principle, no feedback from higher layers in the hierarchy is required for the design of the acoustic features.
However, chapter 4 of Deliverable 1.2 (Feature selection based on auditory models) shows how experiments that are based on a loop involving a back-end algorithm (in this case a specific auditory model) provide useful (and sometimes essential) information about the *experimental design* of the feature selection stage. Feature selection might be complex since the selection algorithms are not necessarily able to over-generate the feature set in a useful manner before subsequent feature selection takes place. Chapters 2 and 3 of Deliverable 1.2 describe the development of voicing-onset time and prosodic features. The use of prosody in experiments is shown to provide some gain during the first phase of the learning, but this positive effect diminishes during further training. This experimental result suggest that prosodic features are to be included in the entire feature vector, but that their weight must be re-estimated based on the activity levels of the internal representations as they get updated during training

7.2 WP2 Signal patterning

With respect to design choices and interpretation, WP2 (and WP3 and WP4) depend more than WP1 on the empirical observations in WP5. For WP2, it has been demonstrated (section 6.3) that the keyword recognition accuracy of the transition probability analysis is a function of number of keywords embedded in the test utterances. This shows that the signal patterning approach in WP2, which is actually an early step in the cascade of processes for learning, can only be optimised using the feedback from a loop that *involves the decoding step*. The dependency of performance of earlier stages in processing as a function of the processing in later stages of the processing shows that the modelling and optimisation of the learning process must be considered as the modelling of one integrated optimisation step in which all back-end results are incorporated. This directly implies the relevance of experimental empirical results on the detailed implementation of sub-stages in this chain.
An example is provided in section 6.3, where it is shown recognition rates for utterances with only a single word turn out to be the most difficult. This has consequences for the way how competition between representations must be modelled, as a function of the overall experimental results.

7.3 WP3 Memory organization and access

In May 2008, ACORNS participants from Sheffield, Nijmegen, Stockholm and Leuven convened to discuss an update and refinement of the ACORNS memory architecture that was originally proposed during the first year of ACORNS by SHFD and RUN.

This cross-WP work has implications for the interpretation of the modules that were defined within the learner in Task 5.1 and vice versa. For example, the exact functionality of STM and LTM has implications for the set-up of the data-processing within the learner (where the activations go, and where the representations are stored). At the same time, the design of caregiver and learner in WP5.1 imposed a specific interpretation of the memory architecture. A very explicit example of interaction between WPs (especially from WP5 and WP4 to WP3) is provided by figure 2.4.

7.4 WP4 information discovery and integration

Deliverable D4.1 describes a bottom-up, activation-based paradigm for continuous speech recognition. Speech is represented by co-occurrence statistics of acoustic events over an analysis window of variable length, leading to a vector representation of high but fixed dimension called “Histogram of Acoustic Co-occurrence” (HAC). During training, recurring acoustic patterns are discovered and associated to words through non-negative matrix factorisation (NMF). During testing, word activations are computed from the HAC-representation and their time of occurrence is estimated. Hence, words in a continuous utterance can be detected, ordered and located.

In the WP5 experiments, cognitive plausibility is a central issue. The way WP5 guides research in WP4 is exemplified in about how the *plausibility* of word activations is verified. In the deliverable D4.1 it is explained how this is done. First, the activations of internal representations must exceed a threshold. Second, the locations (in time) of the detected words need to be consistent over time. Third, it was verified if the order in which words are activated corresponds to the expected activation patterns as learned through previous exposure to the language.

8 Conclusion and discussion

8.1 Summary of the results in Year-2

Task 1: Platform:
Compared to the status after year 1, the caregiver has been updated. The caregiver can now respond to the learner with multiple reactions which may be especially relevant in case of errors made by the learner. These reactions are ignore, repeat the same or a similar utterance, and explicitly correct (by applying a corrective sentence). The learner is updated to deal with vector-based features as representation for the information in the visual channel.
Learning to communicate:
The internal and external loops within which learning takes place are updated with respect to the implementation of learning drive. The learning is now the result of the information provided by the caregiver via the external loop, and driven by the minimisation of a target function operating in the internal loop.

Task 2: Multimodal integration.
The integration of audio and visual information takes place at an early stage at the level of features. The learner combines the two streams of information. The simulated visual stream is upgraded beyond the use of the symbolic tags in the first year, by using a vector-based representations rather than symbolic representations of information presented along the visual channel.

Task 3: Architecture.
The architecture underlying the caregiver-learner interaction and the design of the learner has been discussed, refined, and consolidated. This has been prepared and carried out by a task force with the aim to make the functions of the various types of memory more precise. Moreover, the learner module has been updated such as to optimally reflect the enhanced memory architecture. The new software platform is operational and will be described in a deliverable due M30.

Task 4: Experiments
Experiments with NMF, DP-ngrams and Concept Matrices have been carried out. The NMF based learning experiments show that the use of the more complex Y2-part of the ACORNS database leads to an accuracy of about 85 percent correct identification of concepts in the case [a] trivial visual features [b] utterance-by-utterance incremental training [c] limited number of realisations per concept in active memory (STM).
Novelty detection:
NMF, DP-ngrams and CM all deal with a mechanism for novelty detection based on a metric in the representation space. Approaches with subsymbolic input deal with this in a fundamentally different way than approaches based on symbolic input.

The Y2 database is much more complex than the Y1 database in terms of its lexical contents and the number of concepts per utterance. The experiments in Year 2 have shown that as long as recognisers are used in parallel, the multiple keywords do not seem to be a problem for decoding, but such an approach needs well-trained word-specific recognisers. The NMF experiments on the Y2 database described in this paper show a performance of close to 90 percent, obtained under very specific interpretation of accuracy (accuracy based on tag-to-tag comparison, tag-ordering independent, and the number of items to be decoded known beforehand).

The results obtained with NMF (and DP-Ngrams) depend on parameters with a clear cognitive interpretation. The NMF results presented in this document show how that the learning results depend on basically four parameters: the amount of stimuli that is seen before the initialisation (this can be small if learning is done at all times, but can be larger if the learning only takes places when it is

efficient to build representations), the amount of stimuli used for *update*, the frequency of the update and the 'aggression' in the update. The cognitive plausibility relates to the fact that a conventional ASR-way of dealing with epochs over the entire database leads to non-causal modelling: testing of stimuli while future stimuli are already seen during training. This means that only stimuli that are perceived can be used (more than once) in STM to update internal representations.

8.2 Directions for experiments in Year-3

In this deliverable, several different activities in WP5 have been discussed. In the third year of ACORNS we will concentrate on combining the strong assets of these algorithms. In this section, we discuss a number of issues that play a role in the way how to move forward from the current state of affairs. These issues also make clear how experiments in WP5 influences the other work packages in terms of focussing on specific experimental questions.

Firstly, every learning algorithm must be able to deal with a distance measure (e.g. in terms of a statistical distribution), on the basis of which the algorithm can decide whether a stimulus refers to a new object, quality or event, or whether it refers to something that has previously been processed. Not all algorithms are equally powerful in this aspect. By considering the activations of internal representations, NMF (Del 4.1) can rate the novelty of a new input. The DP-ngrams approach (Del 2.1) handles variation by storing more prototypes after which a pruning must take place to increase the efficiency of the internal representations. The current implementation of Computational Mechanics Modelling (Del 2.1) handles variation in a manner that is basically different from NMF or DP-ngrams, making it very sensitive to *any* variation in the input.

Secondly, one of the questions is whether centroids and templates can be dealt with in a strictly mono-layer learning architecture. The literature on language learning suggests that learning is best modelled by a cascade of algorithms that interact, each on its own hierarchical level. When an input matches very well with one of the stored low-level representations, the learning algorithm does not need to elaborate upon a further interpretation of the input, while deviant inputs (for which a parse in terms of internal representations does not provide a satisfactory result) needs a more 'conscious' response which involves the activation of higher, more abstract levels of processing. The combination of architecture and attention mechanisms described in Del D3.2 in combination with the multi-layer decoding in D4.1 look very promising to address this issue.

Hierarchical models form an issue that must be addressed in more detail. As already mentioned, it plays a role in many of the experiments (D4.1, section 5; in Del 3.2; ten Bosch et al., 2008). It is also addressed, albeit in a different way, in the form of feedback from higher conceptual levels (knowledge) to the feature selection in Del 1.2, chapter 4 (Feature Selection Based on Knowledge of the Auditory System). In Del 1.2 a sensitivity matrix is used to select features according to a selection mechanism inspired by the way humans perceive. In this approach we establish a measure of impact of a given feature based on a perturbation analysis and distortion criteria derived from psycho-acoustic models. Based on this measure a compact set of relevant features is derived. It is assumed that both the features and the distortion criteria are continuous and differentiable functions of the speech signal. In Year 3 we will repeat a number of crucial experiments with the new features to investigate their impact on learning.

References

Baddeley, A.D. (1986) Working Memory Clarendon Press, Oxford.
ten Bosch, L., Van hamme, H., Boves, L. (2008). "A computational model of language acquisition: focus on word discovery", Proc. Interspeech 2008, pp. 2570-2573
ten Bosch, L., Boves L. (2008) "Language acquisition: the emergence of words from multimodal input", in Sojka, P., Horák, A., Kopecek, I & Pala, K. (Eds.) / Text, Speech and Dialogue, 11th Intern. Conference, TSD 2008/, Brno, pp. 261-268.
Boves, L., ten Bosch, L. and Moore R. (2007). ACORNS - towards computational modeling of communication and recognition skills. Proceedings IEEE-ICCI 2007.
Gold, K., Doniec, M., Crick, C., & Scassellati, B. (2009). Robotic vocabulary building using extension inference and implicit contrast. Artificial Intelligence, 173(1), 145-166.
Gopnik A, Meltzoff A N, and Kuhl P K. (2001). The Scientist in the Crib, New York: William Morrow Co.
Hart, B., and Risley, T. (1995). Meaningful differences in everyday experience of young American children. Baltimore: Paul Brookes Publishing Co.
Holzapfel, H., Neubig, D., & Waibel, A. (2008). A dialogue approach to learning object descriptions and semantic categories. Robotics and Autonomous Systems, 56(11), 1004-1013.
Kaplan, F., Oudeyer, P-Y., Bergen B. (2008) Computational Models in the Debate over Language Learnability, Infant and Child Development 17, p 55-80.
Lacerda, F., Klintfors, E., Gustavsson, L., Marklund, E. and Sundberg, U. (2004). Emerging Linguistic Functions in Early Infancy Berthouze, L., Kaplan, F., Kozima, H., Yano, H., Konczak, J., Metta, G., Nadel, J., Sandini, G., Stojanov, G. and Balkenius, C. (Eds.) Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Lund University Cognitive Studies, 123.
Landauer, T. K., & Dumais, S. T. (1997): A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, Vol. 104, pp. 211-240.
Marr, D. (1982) Vision. *A computational investigation into the human representation and processing of visual information.* New York: W.H. Freeman
Marslen-Wilson W. & Tyler L. K. (1980): The temporal structure of spoken language understanding. *Cognition*, Vol. 8, pp. 1-71.
Maslow, A. (1954) Motivation and Personality New York: Harper & Row.
McClelland J. L. & Elman J. L. (1986): The TRACE Model of Speech Perception. *Cognitive Psychology*, Vol. 18, pp. 1-86.
O'Grady, P. D., & Pearlmutter, B. A. (2008). Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. Neurocomputing, 72(1-3), 88-101.
Quinn, P. C. and Eimas, P. D. (1996) 'Perceptual organization and categorization in young infants', *Advances in Infancy Research*, vol. 10, pp. 1-36.
Roy, D.K. and Pentland, A.P. (2002). Learning words from sights and sounds: a computational model. Cognitive Science, 26: 113--146.
Smith, L., Yu C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. Cognition 106. Pp 1558–-1568.
Stouten, V., Demuynck, K., and Van hamme, H. (2007). Automatically Learning the Units of Speech by Non-negative Matrix Factorisation. Interspeech 2007, Antwerp, Belgium.
Stouten, V., Demuynck, K. and Van hamme, H. (2008). Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation. IEEE Signal Processing Letters, volume 15, 131--134.

Background Literature

Baddeley, A.D. (1986) Working Memory Clarendon Press, Oxford.
Bellegarda, J. R. (2000) Exploiting Latent Semantic Information for Statistical Language Modeling. Proc. IEEE, Vol. 88: 1279-1296.
ten Bosch, L., Van hamme, H., Boves, L. (2008). "A computational model of language acquisition: focus on word discovery", Proc. Interspeech 2008, pp. 2570-2573
ten Bosch, L., Boves L. (2008) "Language acquisition: the emergence of words from multimodal input", in Sojka, P., Horák, A., Kopecek, I & Pala, K. (Eds.) / Text, Speech and Dialogue, 11th Intern. Conference, TSD 2008/, Brno, pp. 261-268.

Boves, L., ten Bosch, L. and Moore R. (2007). ACORNS - towards computational modeling of communication and recognition skills. Proceedings IEEE-ICCI 2007.
Cerisara, C. (2009). Automatic discovery of topics and acoustic morphemes from speech. Computer Speech and Language, 23(2), 220-239.
Cooke, M. and Ellis, D. P.W. (2001). The auditory organization of speech and other sources in listeners and computational models, Speech Communication 35 (2001), pp. 141-177
Deerwester, S., Dumais, S. T., Furnas G. W., Landauer, T. K. and Harshman, R. (1990). 'Indexing by latent semantic analysis.' Journal of the American Society for Information Science 41, 391-407.
den Os, E.A., Boves, L., Rossignol, S., ten Bosch, L. and Vuurpijl, L. (2005) Conversational Agent or Direct Manipulation in Human-System Interaction. Speech Communication, 47: 194-207.
Driesen J. and Van hamme H. "Improving the multigram algorithm by using lattices as input", Proc. ICSLP, Brisbane, 2008.
Ernestus, M., Baayen, R.H. and Schreuder, R.. (2002). The recognition of reduced word forms. Brain and Language 81, 162-173.
Feldman, J. (2006) From Molecule to Metaphor: A Neural Theory of Language, Cambridge, Mass: MIT Press.
Gerken, L., and Aslin, R.N. (2005) Thirty years of research in infant speech perception: the legacy of Peter Jusczyk. Language Learning and Development, 1: 5-21.
Gold, K., Doniec, M., Crick, C., & Scassellati, B. (2009). Robotic vocabulary building using extension inference and implicit contrast. Artificial Intelligence, 173(1), 145-166.
Goldinger, S. D. (1996) Words and voices: episodic traces in spoken word identification and recognition memory, J Exp Psychol Learn Mem Cogn, 22(5): 1166-1183.
Goldinger, S.D. (1998). Echoes of echoes? An episodic theory of lexical access. Psychological Review 105: 251--279
Goldinger, S.D. (2000). The Role of Perceptual Episodes in Lexical Processing. In: SWAP-2000, 155--158.
Goldinger, S.D., Azuma, T. (2004). Episodic memory reflected in printed word naming. Psychon Bull Rev. 11 (4):716-22
Gopnik A, Meltzoff A N, and Kuhl P K. (2001). The Scientist in the Crib, New York: William Morrow Co.
Graf Estes, K., Evans, J.L., Alibali, M.W., and Saffran, J.R. (2007). Can infants map meaning to newly segmented words? Statistical segmentation and word learning. Psychological Science. 18(3): 254--260
Hart, B., and Risley, T. (1995). Meaningful differences in everyday experience of young American children. Baltimore: Paul Brookes Publishing Co.
Hermansky, H. (1996) Auditory modeling in automatic recognition of speech. ESCA Workshop on the Auditory basis of speech perception, Keele University (UK), 15-19 July, 1996.
Hintzman, D. L.(1986) Schema-abstraction in a multiple-trace memory model, Psychological Review, 93: 411-427.
Holzapfel, H., Neubig, D., & Waibel, A. (2008). A dialogue approach to learning object descriptions and semantic categories. Robotics and Autonomous Systems, 56(11), 1004-1013.
Hoyer, P.O. (2004) Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research, 5, 1457--1469.
Johnson, S. (2002) Emergence. New York: Scribner.
Jones, D.M., Hughes, R.W. and Macken, W.J. (2006) Perceptual organization masquerading as phonological storage: Further support for a perceptual-gestural view of short-term memory, J. Memory and Language 54, 265 281.
Jusczyk, P.W. (1999) How infants begin to extract words from speech. TRENDS in Cognitive Science, 3: 323--328.
Kaplan, F., Oudeyer, P-Y., Bergen B. (2008) Computational Models in the Debate over Language Learnability, Infant and Child Development 17, p 55-80.
Kuhl, P.K. et al. (2003) Foreign-language experience in infancy: effects of short-term exposure and social interaction on phonetic learning. Proc. National Academy of Science U.S.A., 100: 9096-9101.
Kuhl, P. K. (2004). Early language acquisition: Cracking the speech code. Nature Reviews Neuroscience, 5, 831--843.
Kuhl, P. K., Conboy, B. T., Padden, D., Nelson, T. and Pruitt, J. (2005). Early speech perception and later language development: Implications for the critical period. Language Learning and Development, 1, 237--264.
Kuhl, P. K., Stevens, E., Hayashi, A., Deguchi, T., Kiritani, S., and Iverson, P. (2006). Infants show facilitation for native language phonetic perception between 6 and 12 months. Developmental Science, 9, 13-21.
Lacerda, F., Klintfors, E., Gustavsson, L., Marklund, E. and Sundberg, U. (2004). Emerging Linguistic Functions in Early Infancy Berthouze, L., Kaplan, F., Kozima, H., Yano, H., Konczak, J., Metta, G., Nadel, J., Sandini, G., Stojanov, G. and Balkenius, C. (Eds.) Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Lund University Cognitive Studies, 123.

Landauer, T. K., & Dumais, S. T. (1997): A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, Vol. 104, pp. 211-240.
Lee, D.D., and Seung, H.S. (2001). Algorithms for non-negative matrix factorization. In Advances in Neural Information Processing Systems 13, 2001.
Lee, C.-H. (2004) From Knowledge-Ignorant to Knowledge-Rich Modeling: A New Speech Re-search Paradigm for Next Generation Automatic Speech Recognition. Proc. ICSLP.
Lippmann, R. (1997) Speech Recognition by Human and Machines. Speech Communication, 22: 1--14.
Maloof, M.A., Michalski, R.S. (2004). Incremental learning with partial instance memory. Artificial intelligence 154, 95--126.
Marslen-Wilson W. & Tyler L. K. (1980): The temporal structure of spoken language understanding. *Cognition*, Vol. 8, pp. 1-71.
Marr, D. (1982) Vision. *A computational investigation into the human representation and processing of visual information.* New York: W.H. Freeman
Maslow, A. (1954) Motivation and Personality New York: Harper & Row.
McCarthy, J. (2008). The well-designed child. Artificial Intelligence, 172(18), 2003-2014.
McClelland J. L. & Elman J. L. (1986): The TRACE Model of Speech Perception. *Cognitive Psychology*, Vol. 18, pp. 1-86.
McMurray, B. (2007). Defusing the childhood vocabulary explosion. Science 317 (Aug. 3):631. Abstract available at http://www.sciencemag.org/cgi/content/abstract/317/5838/631.
McQueen, J.M., Cutler, A., Norris, D. (2006). Phonological Abstraction in the Mental Lexicon. Cognitive Science 30 (2006), 1113--1126.
Moore, E. and Clements, M. (2004), Algorithm for automatic glottal waveform estimation without the reliance on precise glottal closure information, Proc. ICASSP 2004, May 14-17, Montreal, Canada.
Moore R K. (2003) A comparison of the data requirements of automatic speech recognition systems and human listeners, Proc. EUROSPEECH'03, Geneva, pp. 2582-2584, 1-4.
Moore R K and Cunningham S P. (2005) Plasticity in systems for automatic speech recognition: a review, Proc. ISCA Workshop on 'Plasticity in Speech Perception, pp. 109-112, London, 15-17 June (2005).
Moore, R.K. (2007) Spoken Language Processing: Piecing Together the Puzzle. Speech Communication.
Moore, R. K., Russel,l M. J., Nowell, P., Downey, S. N. and Browning, S. R. (1994). A comparison of phoneme decision tree (PDT) and context adaptive phone (CAP) based approaches to vocabulary-independent speech recognition, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Adelaide.
Newport, Newport, E., Aslin, R. (2004). Learning at a distance: I. Statistical learning of non-adjacent dependencies. Cognitive Psychology, 48, 127-162.
Norris, D. (1994). Shortlist: A connectionist model of continuous speech recognition. Cognition, Vol. 52, 1994, pp. 189--234.
O'Grady, P. D., & Pearlmutter, B. A. (2008). Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. Neurocomputing, 72(1-3), 88-101.
Pfeifer, R. and Scheier, C. (1999) Understanding Intelligence. Cambridge, Mass.: MIT Press.
Ostendorf, M (1999) Moving beyond the 'beads-on-a-string' model of speech, in Proc. IEEE ASRU-99, Keystone, Colorado, USA. Dec 12-15.
Quinn, P. C. and Eimas, P. D. (1996) 'Perceptual organization and categorization in young infants', *Advances in Infancy Research*, vol. 10, pp. 1-36.
Rizzolatti, G. and Arbib, M. A. (1998) Language within our grasp, Trends in Neuroscience 21, 188-194.
Roy, D.K. and Pentland, A.P. (2002). Learning words from sights and sounds: a computational model. Cognitive Science, 26: 113--146.
Saffran J.R., Newport E.L., Aslin R.N. (1996). Word segmentation: the role of distributional cues. J Mem Lang 35:606–-621.
Saffran, J.R., Werker, J.F. & Werner, L.A. (2006) The Infant's Auditory World: Hearing, Speech and the Beginnings of Language. In: Damon, W., Lerner, R. M., Kuhn, D. & Siegler, R. S. (Eds.*) Handbook of Child Psychology, Volume 2: Cognition, Perception, and Language*, New York: Wiley, pp. 55-108.
Sarma, A. and van der Hoek, A. (2004) A Needs Hierarchy for Teams. ISR Technical Report: UCI-ISR-04-9.
Scharenborg, O., Norris, D., ten Bosch, L. and McQueen, J.M. (2005) How should a speech recognizer work? Cognitive Science: A Multidisciplinary Journal, 29(6), 867-918.
Shannon, C.E. and Weaver, W. (1949) The mathematical theory of communication. Urbana, Il.: University of Illinois Press.
Shalizi, C. R. and Crutchfield, J. P. (2000) Pattern Discovery and Computational Mechanics, Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2K).
Smith, L., Yu C. (2008). Infants rapidly learn word-referent mappings via cross-situational statistics. Cognition 106. Pp 1558–-1568.

Snow, C. and Ferguson, C. (1977). Talking to children: language input and acquisition. Cambridge, New York: Cambridge University Press.

Sroka, J. J. and Braida, L. D. (2005). Human and machine consonant recognition, Speech Communication: 44, 401--423.

Stouten, V., Demuynck, K., and Van hamme, H. (2007). Automatically Learning the Units of Speech by Non-negative Matrix Factorisation. Interspeech 2007, Antwerp, Belgium.

Stouten, V., Demuynck, K. and Van hamme, H. (2008). Discovering Phone Patterns in Spoken Utterances by Non-negative Matrix Factorisation. IEEE Signal Processing Letters, volume 15, 131--134.

Appendix 1 – List of visual/semantic features

Male	Female
Parent	Non-parent
Grown-up	Non-grown-up
Round	Non-round
Alife_or_pretend_alife	Non_alife_or_pretend_alife
Plant	Non-plant
Humanoid	Non-humanoid
Artifact	Non-artifact
Four_legged	Non-four_legged
Edible	Non-edible
Vehicle	Non_vehicle
Felidae	Non-felidae
Furry	Non-furry
Ride-able	Non-ride-able
Has-horns	Has-no-horns
Has-wings	Has-no-wings
Swims	Doesn’t-swim
Has-red-throat	No-red-throat
Aerodynamic	Non-aerodynamic
Positive	Negative
Possession-related	Non-possession-related
Expression	Non-expression
Emotion	Non-emotion
Object-related	Non-object-related
Red	Non-red
Yellow	Non-yellow
Blue	Non-blue
Stable_possession	Non-stable_possession
Gain_possession	Loose_possession
Successful	Non-successful
Big	Small
Has-mane	Has-no-mane

Appendix 2 – Example of BNF Grammar

```
$OBJECTS =
toy |
ball |
bottle |
dog |
doll |
cookie |
telephone |
banana |
bird |
duck |
frog |
cat |
apple |
airplane |
truck |
horse |
tree |
cow |
fish |
lion |
eagle |
robin |
car |
porsche |
animal
;


$PERSON_ADJ =
sad |
happy |
smiling |
crying ;


$VERBS2 =
look_at |
take |
give |
see |
like |
have ;


$PERSONS =
man |
woman |
daddy |
mommy |
baby
;

$PROPERTIES =
edible |
clean |
dirty |
```

```
happy |
sad |
big |
small |
round |
square |
red |
yellow |
blue
;

$DET = ( a | the ) ;

$PROP1 = ( here is | there is ) a [ $PROPERTIES ] $PROPERTIES $OBJECTS and
a $OBJECTS ;

$PROP2 = ( she | he ) $VERBS2 _S $DET $PROPERTIES $OBJECTS ;


$PROP3 = $DET $PERSONS $VERBS2 _S $DET $PROPERTIES $OBJECTS ;
$PROP4 = $DET $PERSON_ADJ $PERSONS $VERBS2 _S $DET $OBJECTS ;

$QUESTIONS1 = ( where is the | do you like a | do you have a | do you see
the ) $PROPERTIES $OBJECTS ? ;

$CORR = CORR [ no ] ( I mean ) ( $VERBS2 | $PERSON_ADJ | $PERSONS | the
$PROPERTIES one | $DET $OBJECTS ) ;


( $PROP1 | $PROP2 | $PROP3 | $PROP4 | $QUESTIONS1 | $CORR )
```

The BNF starts with a list of non-terminals with their value. The eventual graph is defined by the last line in the BNF file.
The BNF as such generates millions of different sentences. In practice, the Y2 database for British has been created by first over-generating about 2 million sentences, followed by a post-processing step that skipped the semantically nonsensical ones (no twice the same keyword in one sentence), no clashing properties (‘big small apple’).

Appendix 3 – Keywords English

Food, food-related

Apple	Banana	Cookie
Bottle		

Animals, toys, environment

Animal	Bird	Robin
Eagle	Fish	Dog
Cat	Lion	Horse
Cow	Frog	Duck
Toy	Airplane	Bottle
Telephone	Car	Porsche
Ball	Doll	Truck
Tree		

People

Woman	Man	Baby
Daddy	Mummy	

Properties

Red	Blue	Yellow
Clean	Dirty	Edible
Big	Small	Happy
Crying	Smiling	Sad
Square	Round	

Actions

Give	Look	See
Like	Take	