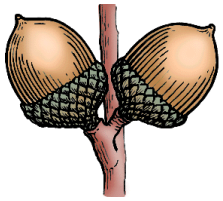


Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D4.3 Report on exemplar-based & activation based matching

Due date of deliverable: 2009-11-30
Actual submission date: 2009-11-27

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

Organisation name of lead contractor for this deliverable: Center for Processing Speech and Images, Department of Electrical Engineering, Katholieke Universiteit Leuven

Revision: 0.4

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

† in order not to interfere with publication of this work in a journal, the content of this report should not be made public before June first 2010

VERSION DETAILS	
Version:	0.4
Date:	27 November 2009
Status:	Draft

CONTRIBUTOR(S) to DELIVERABLE	
Partner	***Name***
K.U.Leuven - ESAT	Kris Demuynck

DOCUMENT HISTORY			
Version	***Date***	***Responsible***	***Description***
0.1	30 oct 2009	Kris Demuynck	first draft
0.2	19 nov 2009	Kris Demuynck	version addressing the reviewer comments
0.3	26 nov 2009	Kris Demuynck	version addressing the reviewer comments
0.4	27 nov 2009	Kris Demuynck	minor corrections

DELIVERABLE REVIEW			
Version	***Date***	***Reviewed by***	***Conclusion****
0.1	04 nov 2009	Lou Boves	see paper version
0.1	04 nov 2009	Vicky Maier	see paper version
0.2	25 nov 2009	Lou Boves	see e-mail
0.2	26 nov 2009	Vicky Maier	see e-mail

Contents

1 Situating this work

1.1 Rationale for an exemplar/episodic based approach

It is a well known fact that the accuracy of current generation state-of-the-art automatic speech recognition (ASR) systems is at least an order of magnitude below that of humans, even for rather simple tasks in noise-free environments. While there is general agreement that novel approaches are needed to overcome the deficiencies of current generation ASR systems, there is little agreement on what the most promising directions are. In the ACORNS-project, we take inspiration from the growing body of knowledge about human cognitive processing and intelligence.
Based on an extensive literature study, the following theories, concepts and observations on human language acquisition and communication were shortlisted as being highly promising:

- The **memory-prediction** framework [38] proposed by Hawkins and Blakeslee states that bottom-up activation in a hierarchically structured memory of frequently observed temporal sequences with associated labels, combined with top-down predictions may explain much of the brain potential observed in humans. As an overall framework, this theory is very appealing since it is based on long standing neurophysiological evidence [61].

- **Mirror neurons** [73] are neurons that act as if the observer itself were acting when merely observing the actions of another being. **Perception-action loops** [36] couple (motor) actions to perceptual patterns at different levels in the hierarchy, allowing fast (latency-free) responses in critical situations. Both mechanisms are thought to be a necessity for learning by imitation, a key mechanism for language acquisition [83, 73, 4].

- The available behavioural data on memory recollection have led many scientist to believe that memory for human speech and language processing is organised in an **associative** manner [62, 3]. The long term memory involved in human spoken language processing is also theorised to be partly of episodic nature [43, 37]. **Episodic memory** stores not only the events but some aspects of the entire context surrounding them as well [87, 88]. HMM-based ASR systems on the other hand work with abstract representations of concepts and knowledge of how these concepts interact with or relate to other concepts. This corresponds to **semantic memory** in humans, i.e. memory that refers to concept-based knowledge unrelated to specific experiences [87]. The relationships between episodic and semantic memory are not exactly known. Some researchers postulate that semantic memory is created by synthesising multiple exposures (possibly stored in episodic memory), updating the semantic representation on each exposure [9, 59]. In the extreme, semantic memory can even be theorised to be mainly a perceived effect of episodic memory [80]. In any case, the fact that creating semantic representations involves some form of "learning from experience" seems acceptable [67]. All these viewpoints suggest a more prominent role to episodic information (stored experiences) compared to the HMM-based modelling (purely semantic representations) used in contemporary ASR systems.

- For humans, patterns in the sensory inputs (e.g. speech) are **emergent properties** [44] that are learnt because of the innate need to associate sensory inputs to

meaningful objects and behaviour in the environment. This stands in stark contrast to current ASR systems which recast the recognition problem into a classification problem based on a fixed and a-priori defined set of patterns such as phones, words, syntactic classes, ...

When compared to human language processing, exemplar based recognition systems such as the one proposed by De Wachter and Demuynck [17, 15] show some intriguing properties. The initial reason in [17] for switching from HMM-based recognition to exemplar based recognition was to avoid the information loss resulting from abstracting large amounts of data into compact models. Similar to the trend observed in speech synthesis, the data is no longer condensed into a model beforehand but is stored unmodified, delaying the processing of the data till the data is actually needed. Hence, instead of first abstracting the data into concept-based models –similar to creating or updating information in semantic memory– the data is stored as recorded after being enriched with some annotations – similar to storing information in episodic memory. When handling the task at hand, the relevant data is extracted from memory and processed, typically using algorithms that involve fewer abstractions than when using an approach based on pre-constructed models.
Based on the above stated links with the human memory system, on could state that exemplar-based recognition systems rely mainly on episodic memory for the (low-level) acoustic decoding. Concept-based knowledge (semantic memory) is needed for both labelling the audio prior to storage in the episodic memory and when integrating the higher level linguistic information during the final decoding. However, in both stages, the higher-level concept-based knowledge used is directly linked with the underlying labelled and stored episodic reference patterns. Further links between the work described in [17, 15] and human language processing can be found in the use of associative memories, the use of exemplars for predictions (template transition costs, natural successors) and the hierarchically based processing, i.e. the acoustic decoding (first layer) comes before the remainder of the decoding, both physically and conceptually.
Another way of characterising the exemplar based decoding approach in [17, 15] is to compare it with the current de facto standard for speech recognition: HMM based systems. The exemplar based approach is both data-driven and hierarchical, i.e. the first layers of the decoder have no or limited access to the higher-level knowledge sources (lexicon, language model); they just find those exemplars (chunks of audio + meta-info) in memory that resemble the acoustic properties of the input data best. This approach differs from classical HMM systems in two major aspects. First, instead of building models for some a priori defined classes (phones), exemplar based decoders simply store all input data and then infer information about the classes under investigation given these examples. One could say that the exemplar based system is making problem specific models when needed and centred around the given input. Second, exemplar based systems organise the search in a hierarchical fashion, incorporating high level information (lexicon, language model) only in the later decoding stages. HMM decoders on the other hand use an all-in-one approach, incorporating all available information at once in order to make the search process maximally efficient (this is for example reflected in the fact that only those acoustic units are evaluated that are needed by the decoder which is directed by both lexicon and language model).
The comparison between exemplar and HMM based ASR at the one side and human language processing at the other side shows that exemplar based ASR is not only the better match but even has potential as a computational model of human language acquisition [57]. Theories and concepts such as memory prediction, the human memory structure and even

learning by imitation can be mapped easily to the exemplar based system. The other aspects of human language acquisition listed at the beginning of this section may form the basis for making the exemplar based ASR self learning. In the remainder of this report we work towards an exemplar-based ASR system that matches the theories and concepts on human language acquisition closely, with some focus on the self-learning aspect.

1.2 Idealised meta-information & high-level knowledge

In order to be able to evaluate the potential of the exemplar-based approach, we idealised the meta-data, lexical and grammatical knowledge sources.
Our exemplar-based decoder requires a labelled database to start from. This means that at least some level of linguistic abstraction (e.g. reusable acoustic units and a lexicon expressing the words in terms of those units) must exist for the current system to work. The first two attempts to automatically derive and use reusable acoustic units in the ACORNS-project, i.e. purely bottom-up [50, 92] and purely top-down [32, 91, 92] showed a noticeable performance drop when compared to the baseline systems that did not use the intermediate "reusable acoustic units" layer. However, based on these two attempts, and based on the work from a master thesis done at the Katholieke Universiteit Leuven [14] (text in Dutch), combined with the results provided in section 3.2 of this report, we believe that a combined bottom-up and top-down-approach will be feasible. A proposal for such a combined approach can be found in section 1.4.4 of this report.
For the time being and in anticipation of a competitive method to automatically derive reusable acoustic units, we resort to using the acoustic units proposed by the human experts, i.e. phones. The same reasoning holds for the other high-level knowledge sources (lexicon and grammar): since we have no generally applicable and systematic approach ready for learning these automatically, we start from human knowledge.
This approach allows us to derive important properties concerning episodic modelling without having to solve the automatic learning problem first. The insight gained this way may also prove useful in creating better algorithms for finding reusable acoustic units and for deriving other high-level knowledge (see section sec:npd2cluster and 1.4.4).

1.3 Scope and aim of this work

In the ACORNS-project, several learning techniques are being explored, such as (incremental) clustering [76], (incremental) matrix factorisation [84, 90, 89, 25], Hebbian learning (neural networks) [30, 31, 29], and computational mechanics models [50, 49, 51]. Most learning techniques imply some form of memory organisation [29]. This work on episodic modelling is different in the sense that here the starting point is the memory model (complemented with methods to infer knowledge). In order to create a self learning agent, one or more learning algorithms have to be added. In section 1.4.4 we will give some propositions for suitable learning algorithms based on the experience gained in the ACORNS-project. However, given that episodic modelling poses few constraints on the learning algorithm, alternative learning algorithms could be easily devised. Furthermore, as is explained in section 1.2, the potential of episodic based modelling can even be investigated independently of the learning algorithm by idealising the meta-information & high-level knowledge.
The fact that this work on episodic modelling is more about memory architecture than about learning, will be reflected in the way the potential of the method is evaluated. The evaluation will be mainly based on the following criteria:

- the cognitive plausibility (similarities with human speech processing), with emphasis on the scalability of the approach: humans can cope with extensive vocabularies and have stored (or condensed) large amounts of speech data during their life and hence the proposed episodic model should be able to match these capabilities;
- performance with respect to state-of-the-art hidden Markov models (HMM's);
- overall feasibility as a computational model, i.e. requiring a realistic and realisable amount of memory and computation power.

This also sets the aim of this work: the main focus is on the building blocks of the episodic model. We will give a description of all basic building blocks, a low level analysis of these building blocks, and the evaluation of a number of prototype implementations on the ACORNS year 2 Dutch database and on the Timit database (see section 1.5 for a description of the databases). A complete self-learning system based on these building blocks augmented with some learning algorithms will also be proposed. However, for an evaluation of the learning algorithms, we refer to the other ACORNS deliverables.

1.4 Situating this work in the ACORNS-project

1.4.1 Outcome of this work

In this deliverable, we will work towards a computational model based on episodic memory capable of doing speech recognition. Key aspects of the architecture are:

- a powerful content addressable memory (CAM) which still works well when looking for near matches (speech data is never identical)
- an efficient inference engine
- (loopy) belief propagation as the preferred method for propagating information and constraints, both horizontally (intra) and vertically (between hierarchical layers)

The operation of the first layer in the model (the low-level acoustic decoding) is summarised hereunder:

- For each input frame, the feature vector corresponding to that frame and some surrounding context is looked-up in the CAM; the CAM finds the k most similar traces (a central frame, some surrounding context, and a set of labels characterising the central frame).
- Based on the k most similar traces, the inference engine builds up an initial (probabilistic) belief on the main problem, i.e. identity of the acoustic unit corresponding to the input frame, and at the same time stores belief on secondary issues such as the gender of the speaker, the left and right context, or the word currently being uttered.
- By means of (loopy) belief propagation, information from surrounding frames (horizontal information flow) and at later iterations even from the higher levels such as lexicon and language model (vertical information flow) is collected.
- The CAM is addressed again for each input frame and the list of k most similar traces is updated taking into account the extra knowledge collected by means of belief propagation - see figure 13 on page 29 for an illustration of this process.
- The process is iterated until the system is satisfied with the explanation (verification of the result) or until a maximum number of iterations is exceeded.

Note: All experiments presented in this work focus on the low-level acoustic decoding. The higher hierarchical layers (lexical and language model) were abstracted using finite state

transducers – a technique borrowed from the HMM-world. Whether these higher layers are best implemented using pre-abstracted models such as finite state transducers or would be better of being implemented as episodic memory (cf. the low-level acoustic decoding) is open to debate and is not essential for the work presented here.

1.4.2 Other episodic modelling approaches in the ACORNS-project

In ACORNS, episodic memory based approaches are also investigated in the work pertaining to *DP-ngrams* [1, 2] and *Temporal Episodic Memory Models* [57, 58] (TEMM). Both studies are complementary to the work presented in this deliverable. The DP-ngram work is mainly focused on learning, i.e. finding recurring patterns and correlating these with the "grounding" information (any additional information, typically from the visual stream, that helps in identifying the object).
TEMM is a new model for automatic speech recognition (ASR). TEMM is derived from a simulation of human episodic memory called MINERVA2 [40], and it not only overcomes the inability of MINERVA2 to use temporal sequence for recognition flexibly, but it also employs a prediction mechanism as an additional source of information [57]. The ASR implementation of MINERVA2 has been shown to have a strong relationship with kNN [56]. The newer work on TEMM in the ACORNS project focused particularly on addressing limitations of current ASR approaches, including HMM, TEMM and k-NN, which is its overly simplistic handling of additional knowledge sources. "Two related fields provide inspiration for this new perspective: (a) cognitive architectures indicate how experience with related problems can give rise to more (expert) knowledge, and (b) case-based reasoning provides an extended framework which is relevant to any similarity-based recognition systems" [58].

1.4.3 Other (learning) techniques investigated in the ACORNS-project

Co-occurrence based techniques and matrix factorisation
Previous research has shown that matrix factorisation techniques are well suited to learn because a factorisation that is both compact and accurate almost automatically implies that a large part of the underlying structure of the problem has been exposed. For problems that can be modelled as a "sum of parts", non negative matrix factorisation (NMF) is the most appropriate technique. The HAC-representations (histogram of acoustic co-occurrences) used in the ACORNS NMF-approach [84, 90, 89, 25] transform the observed speech (a sentence) into a representation for which the "sum of parts" assumption holds, i.e. the HAC-representation of a sentence can be approximated by the sum of the HAC-representations of the underlying words. The HAC-feature vector may contain any countable occurrence or (time-shifted) co-occurrence of acoustic events, for example counts on vector quantised MEL cepstral coefficient labels. Additional acoustic streams and "grounding" information that meets the "sum of parts" assumption (typically counts) can be directly concatenated to the HAC-matrix. Such a concatenation setup impels NMF to decompose the problem into basic units which are consistent across all streams. Given that words (corresponding to objects or object features in ACORNS' world) are, by construction, the natural base for both the HAC-representations and the grounding information, NMF is capable of discovering this.
The concept-matrix approach [74] uses HAC-features as well but forgoes matrix factorisation as learning strategy, relying on other methods to derive simple class labels from the

"grounding" information instead.

Clustering techniques
Clustering algorithms group a set of similar observations (points) together and assign a single label to that group. One such example is the the self-learning vector quantisation developed within the ACORNS-project [76]. Adopting the viewpoint that each cluster c is represented by its mean observation vector w_c, the cluster operation can be readily recasted into a matrix factorisation problem:

$$V \approx W \times H^{(i)}$$

with $V = [v_1 \dots v_n]$ the set of n observations, $W = [w_1 \dots w_k]$ the k mean vectors (one for each of the k clusters), and $H^{(i)} = [h_1 \dots h_n]$ the indicator matrix. The elements of indicator vector h_i are all zero except for element c_i which equals to 1, c_i being the cluster to which observation v_i is assigned.

Spectral clustering
Spectral clustering [63, 24] takes the matrix factorisation point of view to clustering one step further and starts with an inter-point distance, similarity or probabilistic transition matrix.
In ACORNS, spectral clustering was investigated in [8, 14] (texts in Dutch) for finding reusable acoustic units in a bottom-up fashion. Starting point for the spectral clustering was a frame similarity matrix S which contained elements $S_{ij} = 0$ if frames i and j are unrelated and $S_{ij} > 0$ if frames i and j are to some extent related (likely to belong to the same class). For efficiency reasons (the matrix S is huge), S must be sparse and/or structured. Next, the matrix is made symmetric by means of an additive ($S' = \frac{1}{2}(S + S^T)$) or a multiplicative ($S' = \sqrt{S \times S^T}$) transformation. The symmetric matrix is then decomposed in orthogonal (preferably positive) components by either eigenvalue decomposition [63] or by symmetric (tri) NMF [24]. Eigenvalue decomposition only reduces the noise and lowers the dimensionality of the problem. The outcome must be further processed using established techniques such as K-means clustering. Symmetric (tri) NMF on the other hand will return values that can be interpreted as conditional class probabilities when starting from a probabilistic transition matrix S.
Evaluation on an artificial problem showed that spectral clustering can detect the underlying classes even in the presence of noise and showed that the method was not sensitive to the metric used for calculating the inter-frame similarities. Applying spectral clustering on a similarity matrix based on Euclidean distance between single frames (no traces) on a random subset (500k frames) of the Timit database (see section 1.5.2) showed a fair correspondence between clusters and phone labels. As can be seen in figure 1, this also leads to automatic temporal clustering (segmentation) of the data.
An appealing aspect of spectral clustering based on similarities or transition probabilities is that multiple streams of knowledge can be readily integrated. All that is needed is combining the multiple (probabilistic) similarities into a single measure.

Causal State Splitting Reconstruction (CSSR)
Another technique investigated within the ACORNS-project is CSSR [50, 49, 51]. CSSR tries to find the minimal probabilistic finite state transducer that models the observations. The work on CSSR in ACORNS focuses mainly on making the technique robust in the

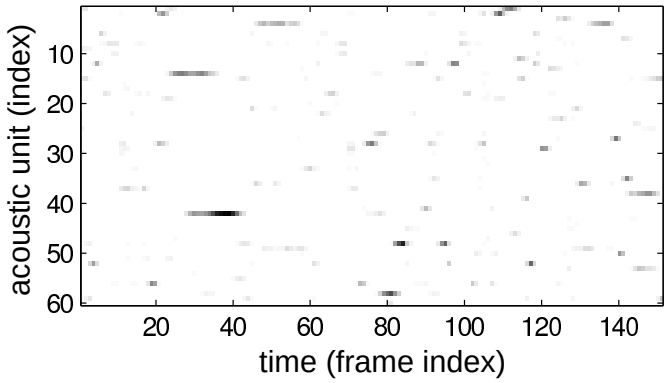


Figure 1: Applying spectral clustering (60 clusters) to the acoustic vectors on the Timit database automatically results in a segmentation of the data which shows a fair correspondence with the manual phone segmentation.

presence of noise. CSSR or similar techniques could be used to automatically derive the higher level representations (lexicon and language model) on top of the episodic memory based low-level acoustic decoding layer presented in this work.

Connectionist expert systems
A last set of modelling and learning techniques investigated in ACORNS are connectionist expert systems (neural networks) such as (temporal) restricted Botlzman machines [82, 29, 85] and hierarchical recurrent self-organising maps [46, 30, 31, 29]. Connectionist systems are based on the observation that most if not all processes can be modelled by interconnected networks of simple and often uniform units. Learning always involves modifying the connection weights, for example using Hebbian reinforcement learning [39].
Despite the large differences at first between our episodic model and connectionist models, it must be noted that most operations needed by our system (e.g. content addressable memory and distance calculations) can be readily mapped to a connectionist architecture. In fact, replacing some of the rigid operations present in our model such as the Euclidean based distance metric by a trainable neural network may add extra flexibility and robustness to the system.

1.4.4 Towards a complete self-learning system

The fact that stored audio samples enriched with some labels (meta-data) form the main knowledge in our approach opens the way to self-learning or adaptive systems. There is no need to retrain complex pre-computed models when new data becomes available, and even large changes such as changing the phone inventory or adding an extra layer of information can be easily handled.
The system can be bootstrapped with any method that can correlate a group of frames (a segment) with some "grounding" information and characterise this correlation with one or more symbolic labels. All learning techniques investigated in the ACORNS-project (see section 1.4.2 and 1.4.3) have shown to be capable of doing this. However, introducing reusable acoustic units in the setup remains a challenge. Hereunder, we propose a scheme that based on our findings is likely to succeed.
In the initial phase, the primary set of labels attached to the acoustics will be directly related

part	sentences	concept words	words	speech (h:mm:ss)	silence (h:mm:ss)
train	9085	29031	53641	2h37:03	4h11:49
test	3024	9682	17847	1h23:44	2h16:34

Table 1: Basic statistics about the train and test fraction of the year 2 Dutch database

to words, i.e. objects or adjectives in ACORNS' world. This automatic labelling can be done with existing techniques (see previous paragraph). The efficacy of the CAM (which is based on traces, i.e. short acoustic fragments) in mapping the acoustics to words is monitored. When the direct mapping from acoustics to word labels proves to be inadequate, some restructuring will be initiated. The restructuring is done by means of clustering. The clustering creates an extra layer of labels (the reusable acoustic units) and a higher-level finite state transducer (cf. CSSR) for linking the acoustic units to the words. Due to its built-in ability for integrating multiple knowledge sources into the decision process, spectral clustering is a prime candidate for this task. By integrating multiple knowledge sources we hope to avoid the performance drop observed when using methods working in a purely bottom-up [50, 92] or purely top-down [32, 91, 92] fashion. Some of the aspects that are though to be important for finding reusable acoustic units (phones) are:

- acoustic similarity (spatial distance) – here the sensitivity analysis from WP1 could be helpful
- temporal distance, cf. segmentation
- information from a previous clustering
- lexical properties, such as (1) this frame typically triggers for this set of words, (2) the frame is located approximately at xx% in word 'X' or (3) the frame has this or that left/right context (based on a previous clustering)

The spectral clustering is a dynamic process: when new words, languages or even dialects are encountered, the system can update its word and possibly phone inventory. Hence the reference to "previous clustering" in the above list.
Once belief propagation is used for both bottom-up and top-down information transfer, simple learning techniques such as Hebbian reinforcement learning can be integrated into the system, allowing small, continuous and incremental updates instead of the large re-structuring operations caused by spectral clustering.

1.5 Databases: content, labelling and baseline systems

We used two different databases for the speech related experiments in this report:

- the year 2 Dutch database, and
- the Timit database.

1.5.1 The year 2 Dutch database

The database year 2 Dutch database was split into two parts: $\frac{3}{4}$ of the data was used as training material while the remaining $\frac{1}{4}$ was kept for testing purposes. The division between train and test material is random (no balancing of speakers, keywords, or sentences) and is kept constant over all experiments (and identical to the division used in [45]). The main properties of the train and test parts are summarised in table 1.

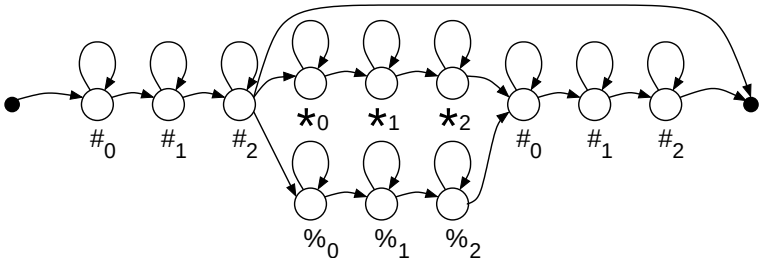


Figure 2: The 'complex' silence model: $\#_{0-2}$ are silence states, $\%_{0-2}$ are speaker noise states, and $\star_{0-2}$ are garbage states

The database contains 81 different word forms. Of these 81 word forms, 65 correspond to one of the 51 concepts to be learnt while the other 16 are considered to have merely a syntactical role. The many to one mapping between word forms and concepts stems from the fact that words (lemma's) get inflected when used in a syntactically correct sentence. There are no synonyms. Sentences contain between 1 and 3 concepts.
Conform to the other experiments conducted on this database in the ACORNS-project, all recognition experiments are scored on the concept level, i.e. the word forms are mapped to the corresponding concepts (or empty for the 16 word forms that do not correspond to a concept) before calculating the error rate (sum of substitutions, insertions and deletions) on the resulting concept strings.
In order to create the phone labelling (see section 1.2) and to provide a reference framework, a task specific HMM was created using the audio material from the train part of the year 2 Dutch database .
To train the HMM, the following resources were used:

- The orthographic transcription (word forms) of all sentences in the train database;
- A manually verified pronouncing dictionary (phonetic lexicon) containing the likely pronunciation variants of all word forms – this lexicon ended up having 32(+1) distinct phones (+silence);
- A set of phonetic properties of the phones used in the lexicon, used by the decision tree procedure to create the context-dependent tied states;
- An initial state-based segmentation created by means of an existing Dutch (Flemish) acoustic model.

The HMM was designed using our in-house state-of-the-art speech recognition system [20] using parameter settings that are expected, based on experiences with other tasks, to yield the best results. The 32 phones are modelled by standard 3 state left-to-right models. The silence/noise model contains 9 states in the configuration depicted in figure 2.
Our default tied gaussian approach (one large pool of gaussians, mixtures select those gaussians they need to model the observations best) was used when training the HMM. The density function for each of the 539 cross-word context-dependent tied states is modelled as a mixture over an arbitrary subset of gaussians drawn from a global pool of 5650 gaussians. The mixtures use on average 76.1 gaussians to model the 36 dimensional observation vector. The 36 dimensions were obtained by means of a mutual information based discriminant linear transformation (MIDA) [21, 20] on 24 MEL spectrum coefficients and their first and second order time derivatives.
The obtained HMM was used to create the phone and word segmentations needed by the

Acoustic model	WER	Ins	Del	Sub
Southern Dutch (Flemish)	4.09%	0.15%	0.64%	3.31%
Northern Dutch	1.48%	0.07%	0.18%	1.22%
ACORNS Dutch, Year 2	0.23%	0.03%	0.07%	0.13%

Table 2: HMM-based reference results (results in percent error)

exemplar-based decoder (see section 1.2 for the rationale) and to create some reference results. To create the reference results, one extra resource was used: a context free grammar (CFG). The grammar matches the sentences exactly except for the fact that the limit of having no more than three keywords was lifted. The grammar is listed in Wirth syntax notation [97] in appendix A.
Table 2 gives the word error rates (scoring on the concept level as described above) for the task specific HMM and for two generic models (one for Flemish and one for Northern-Dutch). These numbers indicate what kind of results one could obtain with state-of-the-art HMMs systems created in the old fashioned way, i.e. using human induced knowledge. This sets a target for the ACORNS systems as well: if a system can learn everything and at least matches the modelling capacity of HMM systems, it should be able to attain or surpass these results.
A quick error analysis showed that of the 24 errors,

- 8 errors are due to confusable pairs: 4× `(heeft, geeft)` confusion, and 4× `(groot, rood, rond)` confusion;
- 6 errors are due to highly prosodic speech in some of the corrective sentences (note that speech with exaggerated prosody likewise led to errors in the phone labelling of the train material);
- 5 errors are due to syntax mismatches between what was said (a hesitation, broken-off words, or an incorrect sentence) and the strict grammar used for recognition.

The remaining 3 errors could not be attributed to an observable property of the speech.

1.5.2 The Timit database

By design (a limited set of phonetically rich sentences repeated multiple times, availability of comprehensive phonetic transcriptions, no definition of a vocabulary or a language model), the Timit database [52, 34] is well suited for research on the low-level acoustic decoding aspect of a recogniser. This is also how the Timit database was used in this work. The Timit database was used for other task in the ACORNS-project as well:

- evaluation of the new features from WP1 [79]
- evaluation of the blind segmentation algorithm developed in WP2 [75]
- development of the first preliminary versions of the Self-learning vector quantisation (SLVQ) in WP2 [76]
- testing and evaluation of the permutation studies in WP3 [51]

For our evaluation, we followed the framework set forth by Lee [53]. The main characteristics of the train and test data are listed in table 3. The train and test speakers and sentences are mutually exclusive. For the baseline HMM system, we mapped the 61 phone (+silence) labels present in the Timit database to 51 labels used internally by the recogniser and the standard 39 labels for the final scoring. We trained both context-independent (ci)

part	speakers	sentences	phones	speech (hh:mm:ss)	silence (hh:mm:ss)
train	462	3696	142910	2:42:46	0:15:38
test	168	1344	51681	0:59:28	0:09:22

Table 3: Basic statistics about the train and test data in the Timit database

	phone N-gram				
Acoustic model	**/**	**1-gram**	**2-gram**	**3-gram**	**4-gram**
	our baseline systems				
context independent	30.3%	29.0%	26.7%	25.6%	25.3%
context dependent	27.9%	27.3%	24.8%	23.8%	23.6%
context dependent + VTLN	26.6%	26.2%	23.7%	22.7%	22.5%
	results from literature				
HMM, context independent			35.9%		
HMM, context dependent			30.5%		
HMM, discriminative training			25.6%		
neural network			24.2%		

Table 4: HMM-based reference results (results in percent error)

and context-dependent (cd) acoustic models. For the context-dependent model, we created a variant that uses vocal tract length normalisation (VTLN) [27] as well. VTLN is used to compensate for the difference between male and female voices. For the evaluation, the acoustic models were combined with phone transition models (N-grams) of different lengths estimated on the phone label sequences (remapped to 51 labels) for the training part of the Timit database. Table 4 gives the results obtained with our baseline HMM systems augmented with some reference results found in the literature [77].
The fact that our HMM baseline outperforms the context-dependent HMM systems from the literature in table 4 can be attributed to the following factors:

- The discriminative preprocessing optimisations [21, 20].
- The fact that our system shares resources to a larger extent [26, 20] than most other systems, giving it an advantage on relatively small databases such as Timit.
- A more sensible choice of the number of state used per phone: most systems use 3-state models for all phones. However, given that in the Timit database the the shortest phones, i.e. the plosives were subdivided in two parts (the closure and the release), we opted to deviate from the standard 3 state models and instead use in between 2 and 4 states for the phones based on their average length: plosives (closure and release) are modelled with 2 states; syllabic phones and diphthongs are modelled with 4 states, all other phones (and silence) use the default 3 states. Note that this choice was made based on the content of the Timit database only. It was never verified whether this choice actually improves the accuracy.
- A more sensible choice of the number of phones to model: we mapped the 61 Timit labels to our standard set of phones which have proven to work well on a variety of recognition tasks, resulting in 51 classes. Most other systems either model the 61

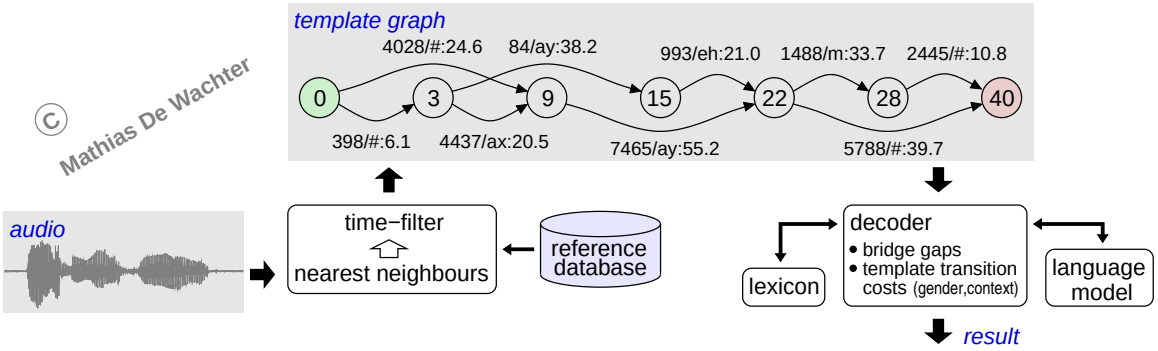


Figure 3: The exemplar-based ASR system created by Mathias De Wachter.

Timit classes or the 39 classes used in [53] directly, which may lead to under or over generalisation in both the acoustic and the language (phone transition) model.

- The use of vocal tract normalisation.

2 Time synchronous exemplar-based matching

Figure 3 gives a high-level overview of the exemplar-based ASR created by Mathias De Wachter [19, 15]. The system decodes a test sentence by means of the following steps:

Nearest neighbour search: for each frame in the test sentence, find the set of k frames from the train (reference) database that are most similar (nearest neighbours);

Time-filter: search for diagonal activation patterns in the activation plot created by the nearest neighbour search – this is illustrated in figure 4;

Template-graph: perform DTW-alignment for the regions found by the time-filter and store the result in a template graph (the templates represent phones or larger units);

Word-decoder: detect word sequences in the template graph using the template to phone (sequence) mapping, a lexicon and a grammar.

A comprehensive description of the system can be found in [15].
In this system, in order for the last step (the word-decoder) to have any chance of finding the correct string of words, there must be at least one path through the template graph that matches the phone sequence put forth by the lexicon given the word string exactly. This constraint frequently collides with the bottom-up activation approach used in the first layer. The bottom-up approach ensures that the information stored in the template graph matches well with the observed acoustics. The observed acoustics tend to differ substantially from the (canonical) lexical forms: in spontaneous speech, phones will be omitted or extra phones will be inserted, silence may contain speaker noises (breathing, coughing) or filler words, there are broken-off words, ...
By design, the time-filter approach may leave unexplained gaps (chunks of signal for which no template was activated) in the signal. These gaps occur if a chunk of signal doesn't match any of the templates well or if it matches too many templates. In both cases, the nearest neighbours for the input frames will come from a large set of different templates,

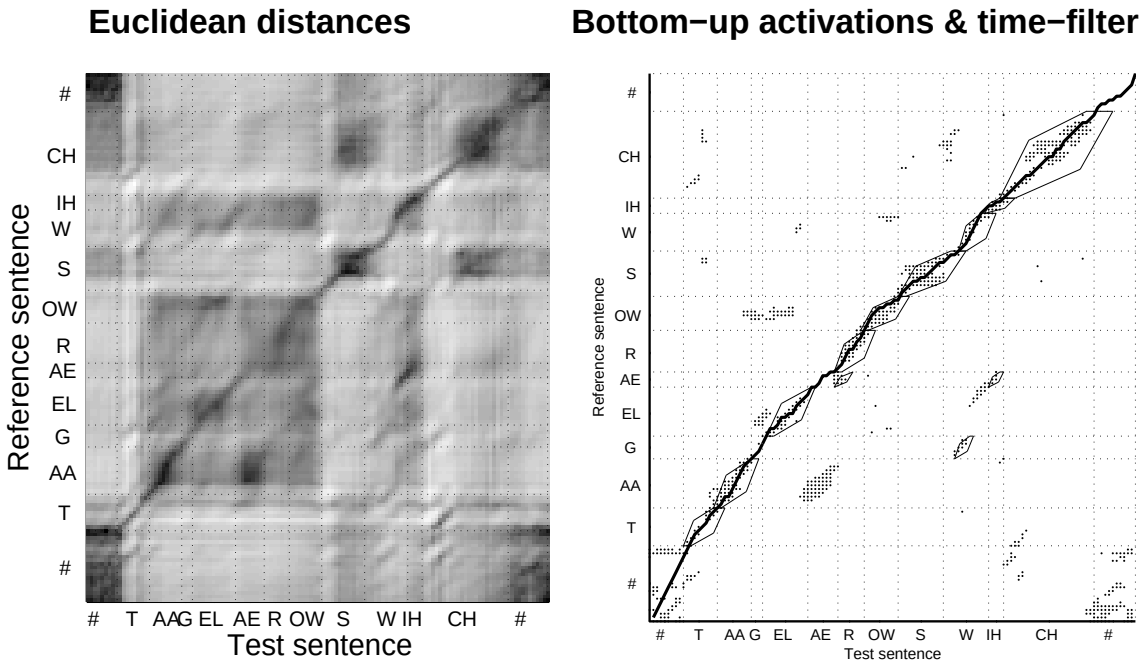


Figure 4: Operation of the time-filter. Based on the bottom-up activation pattern, the time-filter finds likely regions for doing the actual DTW-alignment (activation verification).

resulting in a low chance of having several matching frames from a single template in a near diagonal pattern (as needed by the time-filter to activate the template) among the k selected nearest neighbours. In the case of "no good match", this outcome is defensible and the decoder has provisions to work around these gaps: the "natural successor" scheme automatically promotes the successor phone template for each detected template as a candidate to bridge gaps, i.e. the stored templates are used to predict likely candidate templates when the acoustically driven bottom-up decoding leaves gaps. In the case of many good matches (e.g. silence), this outcome is sub-optimal since the work-around provisions are less reliable than having a good input lattice.

These two effects (gaps in the lattice and mismatches between canonical forms and observed acoustics) explain the fairly high error rate of 6.54% (0.62% insertions, 3.55% deletions, and 2.38% substitutions) observed when using De Wachter's system on the ACORNS year 2 Dutch database. The high error rate illustrates that the "beads-on-a-string" modelling technique [68] used in HMM systems cannot be transposed directly to a bottom-up (activation driven) template based system: left-to-right decoding with a beads-on-a-string modelling of words and sentences is likely to clash with an activation driven input.

The above described approach has some interesting similarities to and differences with the DP-ngram technique explored in the ACORNS-project (see also section 1.4.2). The first step in decoding with the DP-ngram technique is, similar to what is being achieved with the nearest neighbour search and time-filter in De Wachter's implementation, looking for matching templates (memory look-up) given the incoming signal. The outcome of this step (a set of activations, representable as a graph) is then forwarded to the next (hierarchically higher) layer which does the word (concept) decoding. Apart from small implementational differences, there are two key differences: (1) the DP-ngrams in ACORNS are self-learning and (2) the DP-ngrams work in a key-word spotting mode, i.e. only those parts of the signal

for which matching templates are found, are being decoded. In other words, the DP-ngram does not assume a beads-on-a-string model for speech: the relation between acoustics and words are self-learned instead of relying on a left-to-right phone string model while the keyword spotting behaviour allows for unexplained gaps in the signal, gracefully bypassing the overly strict left-to-right word sequence view of speech.
Note: De Wachter implemented techniques, similar to those presented is [23, 22] to cope with the differences between the phone sequences expected by the lexicon and the real observed acoustics. Since this requires even more knowledge to be brought into the system and since these techniques mainly work around instead of solving the two basic problems, we did not further investigate them.

3 Activation based matching

In this section, we will gradually build up an alternative to the time synchronous exemplar-based matching approach described in the previous section. Whereas the time synchronous approach tries to explain the incoming signal as a sequence of pre-defined templates, the activation based matching approach will adopt a more versatile frame-by-frame approach. For each frame (optionally extended with some surrounding context) we will look for similar frames in the episodic memory (the reference audio data, enriched with labels). In other words, at each time (frame), the new system will correlate the input with the episodic memory, and hence build some rudimentary idea about what is being said (or realise that it is unimportant, e.g. silence or noise). This initial belief will then be refined (verified) using belief propagation from nearby frames (horizontal information flow) or using back-propagated information from processing layers higher up the hierarchy (vertical information flow).

3.1 Frame classification using k-NN as non-parametric density estimator

In this section, we investigate the very first level of the activation based decoder: the frame by frame k-nearest-neighbours (k-NN) module.
As was explained in section 1.2, we use phone or word labels and segmentations to evaluate the episodaic memory model instead of relying on (yet to be proven) automatically learned units. The phone and word segmentations for both the train and test data of the ACORNS year 2 Dutch database and the Timit database were created using the database specific HMM's (see section 1.5). These phone and word labels, in combination with the speaker ID and gender form the meta-data of the frames. Hence, every word and phone in the database is characterised by a set of acoustic vectors provided by the preprocessing and some meta-information.
The aim of the experiments in this section is to see how good the meta-information could be predicted (by means of majority voting) given a single acoustic vector from a test sentence and the k nearest-neighbour frames (k ranging from 1 to 100) found in the labelled train sentences when using the Euclidean distance metric. So basically, the parametric probabilistic models used by e.g. HMM's are replaced by non-parametric probability estimates based on k-NN. This is depicted in figure 5.
The acoustic feature vectors used for k-NN based classification are identical to the ones used in the HMM-systems: MIDA transformed spectral coefficients and their first and

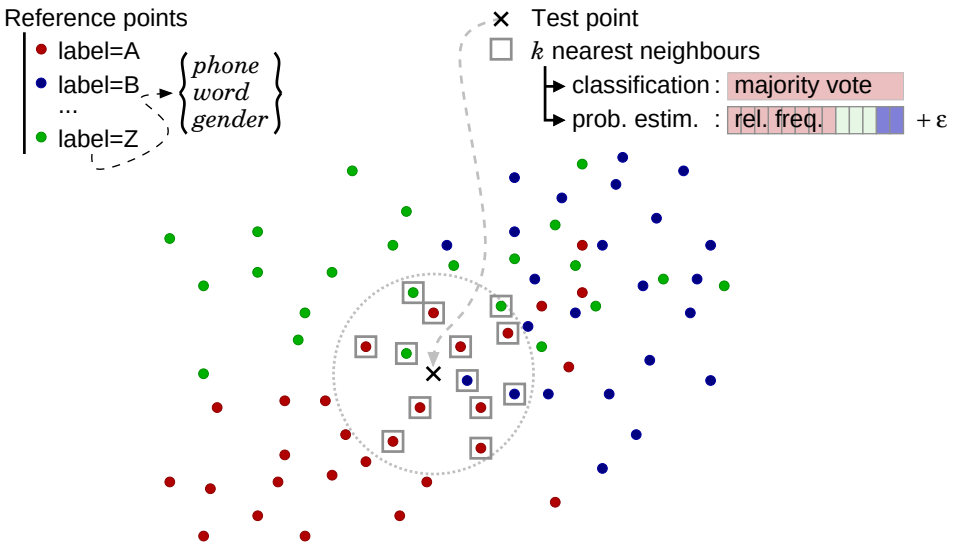


Figure 5: Using k-NN as a classifier or a (non-parametric) probability estimator.

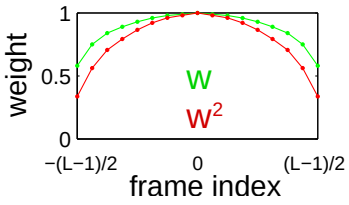


Figure 6: Typical weighting curve for the frames in a trace of length L (concatenation of L frames). Since ordering is based on Euclidean distances, the importance of a basic feature component is proportional to the square of the weight, i.e. w^2.

second order time derivatives. The Timit features also incorporate VTLN.
The k-NN based systems were also evaluated when using extended feature vectors. These extended feature vectors –called (short) traces[1] in the remainder of the text– consist of the concatenation of the feature vectors of L ($L = 3, 5, \ldots 19$) frames. The trace inherits its meta information (phone, word and speaker labels) from the central frame. In other words, the trace extends symmetrically around the original frame. The feature vectors for each frame in the trace get different weights depending on the frame's distance from the central position as depicted in figure 6. Since we use Euclidean distances (sum of squared differences), the importance of a basic feature component is proportional to the square of the weight (i.e. w^2, see figure 6).
Figure 7 summarises the results for the year 2 Dutch database for $k = 15$. The figure also shows the phone classification error rate based on the posterior phone probabilities of the reference HMM system. The posterior phone probability for a given observation x was derived from the state likelihoods $f(\text{state}|x)$ as follows:

$$P(\text{phone}|x) = \sum_{\text{state}} f(\text{state}|x)\, P(\text{state}|\text{phone})$$

[1] the meaning of the term "trace" in this text differs from the meaning given to that term in the literature about TEMM, see also section 1.4.2

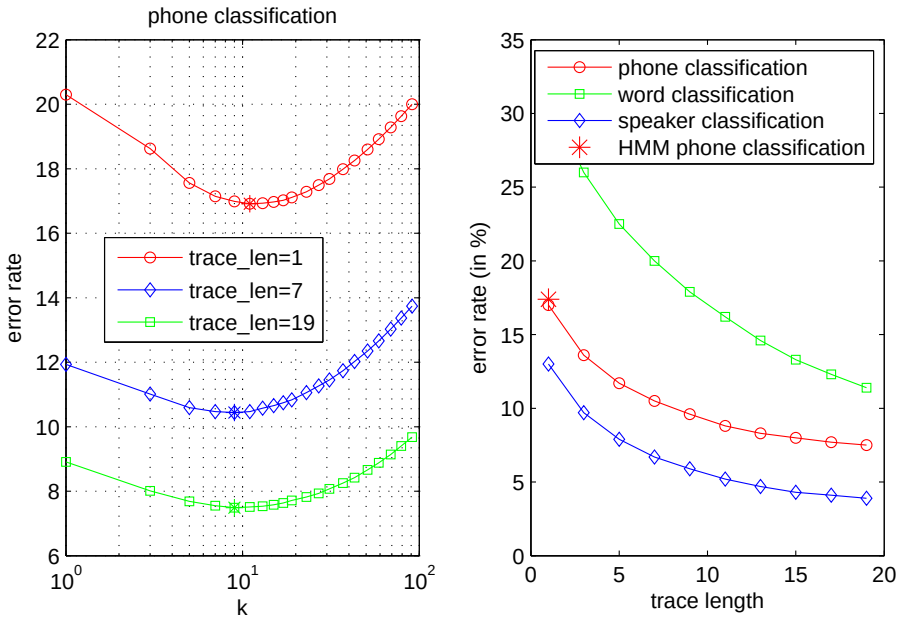


Figure 7: Phone, word and speaker classification error rates (frame classification) on the ACORNS year 2 Dutch database in function of the trace length L for $k = 15$.

Since this is the same HMM-system used to create the phone and word segmentations the results are a bit optimistic, but they nevertheless make a good reference for comparing the k-NN results. The results in figure 7 clearly show that the k-NN approach does not have to yield to the HMM-based system at all.

Figure 8 shows the influence of changing k, the number of neighbours, on the main performance metrics. Using a weighted voting system for classification, i.e. giving each neighbour r_i a weight proportional to $\exp(-\alpha||x - r_i||^2)$ did not show any significant differences in any of the metrics (except for the trivial case where $\alpha \to \infty$). In all experiments (also those reported on in the next sections), the optimal value of k was found to be fairly low (between 9 and 15). This is translated into even fewer (typically less than four) phone labels that are found probable.

Both Cowan [10, 11] and Oberauer [65] assume, based on various studies on working memory [12, 5, 66, 11][2], strong limitation on the capacity of working memory in their computational models. A similar limit on capacity is observed in short term memory [60][2]. Our experiments show that even for low level tasks such as acoustic decoding, the number of concurrent hypotheses can be kept low without adversely affecting the performance.

Figure 9 shows the phone classification results obtained on the Timit database ($k = 15$); given the content of the Timit data base (see section 1.5.2), word nor speaker classification made sense. Compared to the results obtained on the ACORNS year 2 Dutch data, we see that the exemplar based setup has more problems with the Timit data: the HMM system outperforms the k-NN system with a trace length of 1 while at the same time the use of

[2]the limits are typically measured in experiments where different words must be remembered (short term memory) or correlated (working memory), tasks which are not directly comparable to the low-level acoustic decoding process we investigate; in fact later research revealed that the capacity of short term memory depends somewhat on the complexity and familiarity of the items that must be remembered

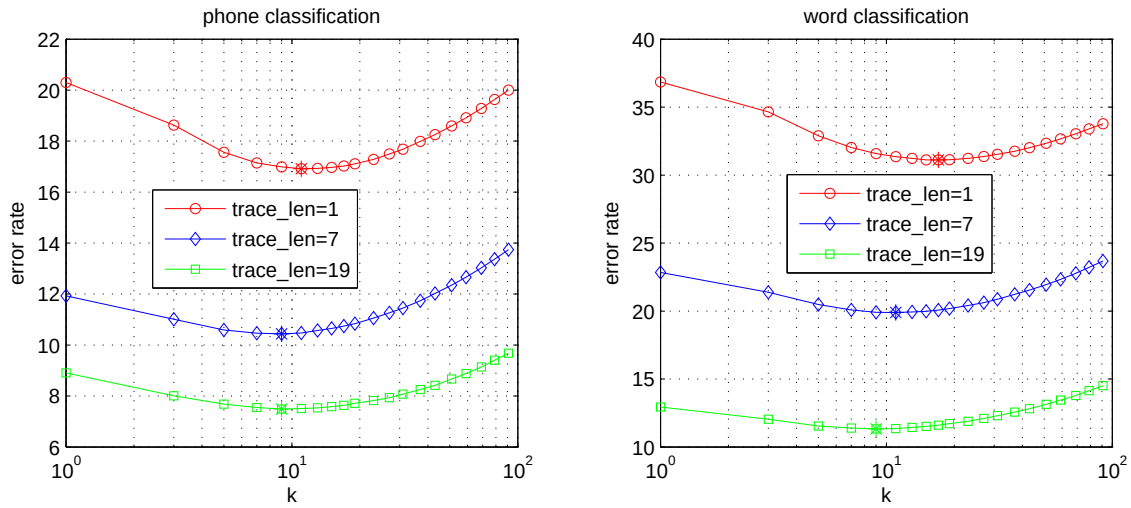


Figure 8: Influence of k (the number of neighbours) on the classification error rates on the ACORNS year 2 Dutch database for 3 difference trace lengths.

longer traces has only limited effect. This behaviour might be explained by the differences in content between the two databases: the Timit databases has substantially less data per speaker (21 seconds versus 942 seconds for the ACORNS year 2 Dutch database) with far more variation in that data (read aloud sentences from diverse sources versus child directed simple sentences using a vocabulary of only 81 words) and has mutually exclusive test and train speakers. This results in far fewer possible neighbours with matching meta-data (matching phone label, and preferably a matching speaker gender and matching surrounding phones as well) and a higher probability of an acoustically similar intruder (and example with a non-matching phone label). In order to achieve optimal results, the k-NN system will need a more powerful inferential system, i.e. more properties of the available data must be used in order to extract the relevant information for the task at hand from the exemplars. Techniques to improve the inferential system will be explored in section 3.5 and 3.6.

3.2 Implications for the automatic detection of acoustic units

Figure 10 shows the confusion matrices for the frame based k-NN phone recognition experiments for the ACORNS year 2 Dutch database. The low confusion overall indicates that the phones correspond to well separable clusters when using acoustic differences measured by means of Euclidean distances as clustering criterion. This observation does not exclude the existence of alternative grouping schemes that would show the same or even a lower confusion overall. In other words, the relation between *phone classes* and `acoustic clusters` corresponding to a single arrow in first-order logic expressions:

$$\textit{phone classes} \rightarrow \textit{acoustic clusters}$$

Acoustic similarity (Euclidean distance), especially when using short traces, thus form an important, but not necessarily sufficient condition when automatically detecting reusable acoustic units (phones). See section 1.4.4 for a list of other aspects that are though to be important and a scheme to integrating all knowledge sources using spectral clustering.

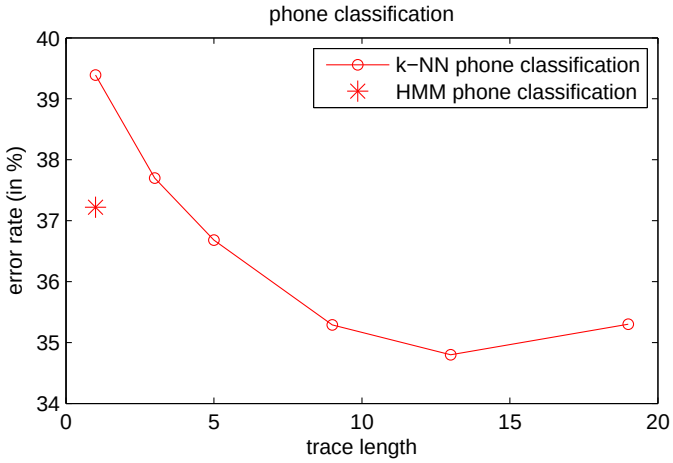


Figure 9: Phone classification error rates (frame classification) on the Timit database in function of the trace length L for $k = 15$.

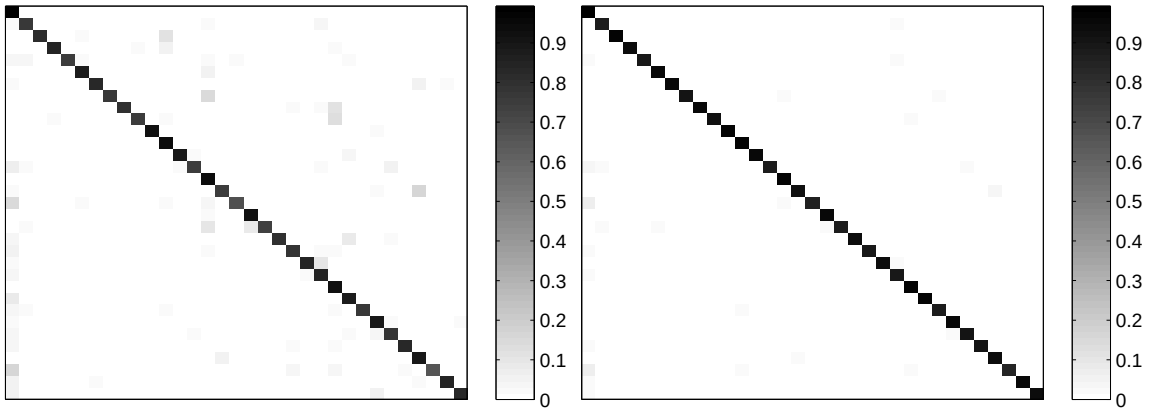


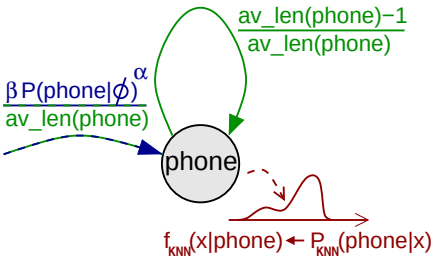
Figure 10: Phone confusion matrices (frame based classification) on the ACORNS year 2 Dutch database for trace length $L = 1$ (left) and $L = 19$ (right).

3.3 Phone recognition using short traces

The k-NN approach, especially when using short traces, yields very competitive results with state-of-the-art HMM systems on the frame level. In this section, we will investigate how the k-NN approach performs when doing continuous recognition of larger units such as phones or words.

This was achieved by making a very simple HMM-system: single state context-independent phones with k-NN based conditional likelihood estimators as state likelihoods functions. A value proportional to the conditional likelihoods $f_{\text{KNN}}(x|\text{phone})$ given the observation x can be derived from the k-NN based posterior probabilities $P_{\text{KNN}}(\text{phone}|x)$ using Bayes' rule for conditional probabilities:

$$f_{\text{KNN}}(x|\text{phone}) \cong \frac{p_{\text{KNN}}(\text{phone}|x)}{P(\text{phone})},$$



av_len(phone)	average length of the phone (in frames)
P(phone|ϕ)	unigram probability of the phone segment
α	LM scale factor
β	phone startup cost
f_{KNN}(x|phone)	likelihood of observing x given phone

Figure 11: The single state HMM's used for phone recognition with non-parametric density estimates.

Acoustic model	**PER**	**Ins**	**Del**	**Sub**
k-NN, $L = 1$, 1 state	10.55%	3.19%	4.41%	2.95%
HMM, ci, 1 state	10.89%	2.29%	5.62%	2.97%
k-NN, $L = 19$, 1 state	3.79%	1.40%	1.48%	0.91%
HMM, cd, 3-state	3.98%	0.80%	1.84%	1.34%
HMM, ci, 3-state	8.73%	1.69%	4.26%	2.77%

Table 5: Year 2 Dutch phone recognition results using single state HMM's with non-parametric density estimates for trace lengths $L = 1$ (a single frame) and $L = 19$ (about the duration of three phones).

i.e. one must divide the phone posteriors by the phone priors. This can be expressed directly as a function of counts as well:

$$f_{\text{KNN}}(x|\text{phone}) \cong \frac{C_k(\text{phone}|x)}{C(\text{phone})},$$

with $C_k(\text{phone}|x)$ the vote counts for each phone class given the k nearest neighbours given observation x, and $C(\text{phone})$ the total number of examples (frames) in the database for the phone class.

Unseen phone classes in the set of k neighbours were assigned a very low likelihood – low enough to avoid that the phone class would be recognised, but high enough to avoid problems with the decoder (which uses log-likelihoods).

The information on the likely phone sequences (cf. the language model in word recognition) was limited to the unigram phone 'LM' for the ACORNS year 2 Dutch setup. Figure 11 illustrates the resulting simple single state hidden Markov models.

Table 5 shows the results obtained on the ACORNS year 2 Dutch database with the k-NN system and with an HMM using 3-state context-dependent phone models (cd) and 3-state and single state context-independent (ci) phone models.

The results for the Timit setup using a 2-gram phone transition model, again compared with a context-dependent (cd) and two context-independent (ci) HMM based systems, are given in table 6.

For the ACORNS year 2 Dutch database, the k-NN based setup performs as good as the corresponding HMM systems. Using a trace length of 19 (about the duration of three phones) brings the k-NN system on parity with the context-dependent HMM system. The Timit results again show that with more diverse data and fewer examples per speaker a more powerful inferential system is needed to achieve parity with our baseline HMM system.

Acoustic model	PER	Ins	Del	Sub
k-NN, $L = 1$, 1 state	31.80%	3.72%	10.41%	17.67%
HMM, ci, 1 state	28.99%	3.27%	9.33%	16.39%
k-NN, $L = 5$, 1 state	31.59%	4.51%	9.38%	17.70%
k-NN, $L = 19$, 1 state	35.01%	3.84%	12.02%	19.15%
HMM, cd, [2-4]-state	23.73%	2.59%	6.26%	14.88%
HMM, ci, [2-4]-state	26.70%	2.70%	7.65%	16.36%

Table 6: Timit phone recognition results using single state HMM's with non-parametric density estimates for trace lengths L of 1 (a single frame), 5 (about the duration of a single phone) and 19 (about the duration of three phones).

Acoustic model	LM	WER	Ins	Del	Sub
k-NN, phone, 1 state, $L = 1$	CFG	0.44%	0.08%	0.02%	0.33%
HMM, ci-phone, 1 state	CFG	0.44%	0.09%	0.01%	0.33%
k-NN, phone, 1 state, $L = 19$	CFG	0.23%	0.07%	0.03%	0.13%
HMM, cd-phone, 3 state	CFG	0.23%	0.03%	0.07%	0.13%
HMM, ci-phone, 3 state	CFG	0.24%	0.02%	0.01%	0.21%
k-NN, concept, 1 state, $L = 1$	CFG	9.17%	1.05%	6.88%	1.25%
k-NN, concept, 1 state, $L = 19$	CFG	6.94%	0.73%	5.57%	0.64%
k-NN, concept, 1 state, $L = 1$	none	11.11%	1.32%	9.10%	0.69%
k-NN, concept, 1 state, $L = 19$	none	7.98%	0.57%	7.08%	0.33%

Table 7: Year 2 Dutch word recognition results obtained with different k-NN based system and with some baseline HMM systems. The k-NN based systems use single state phone or concept models with non-parametric density estimators (see figure 11). Results for two trace lengths are given: $L = 1$ (a single frame) and $L = 19$ (about the duration of three phones). The HMM systems either use context-independent (ci) or context-dependent (cd) phone models consisting of 1 or 3 states. The recognition systems are either completed with a CFG or are used without LM.

Using traces longer than 5 frames even deteriorated the results, indicating that not enough close matches could be found for units longer than a phone.

3.4 Word recognition

Since the Timit database is not well suited for word recognition experiments (see section 1.5.2), all experiments reported in this section are limited to the ACORNS year 2 Dutch data. Several exemplar based approaches have been tested. A first setup consists of the single state phone models presented in the previous section combined with a phonetic lexicon and a context free grammar (CFG) to make a complete recognition system. Both the lexicon and the grammar are identical to those used by the HMM system.
A second setup uses single state concept models, i.e. the k-NN based conditional likelihood estimators that act as observation density functions now work directly with the concept labels. The single state concept models are either combined with a CFG on the concept level or are used without LM.
Table 7 gives the concept error rate obtained by the HMM system and by the different

exemplar-based setups. We see that the exemplar-based setups yield results that are as good as those of state-of-the-art HMM systems. When the k-NN based posterior probability estimators work on a single frame the results match those obtained with an HMM system built around single state context-independent phone models. When upgrading the input of the k-NN based system to short traces (19 frames or approximately 3 phones long) the results match those obtained with an HMM system built around 3-state context-dependent phone models.
These results are very promising, especially considering that the k-NN setups are sub-optimal and can still be improved in many aspects:

- The current setup uses single state models, meaning that there are no explicit constraints on the evolution of the observed acoustics within a phone (phone models) or 'word' (concept models). There is also no duration modelling (a 3-state phone HMM at least imposes a 30msec minimal phone duration). Note that using trace lengths greater than 1 does introduce a preference towards naturally evolving acoustics within a state. This implicit biasing may be (almost) as effective as explicit constraints for "low diversity" databases such as the ACORNS year 2 Dutch database, but for databases such as Timit which express the full diversity of speech, additional constraints are expected to be beneficial.

- There are no explicit "fluidity" constraints. One could for example assign costs to gender switches or incorrect left/right phone contexts. This would be the equivalent to gender-dependent and context-dependent phone models in the HMM world. See section 3.6 for a method of how to incorporate such additional constraints and some results from the literature.

- With sufficient data available per phonetic context, the k-NN approach clearly thrives on longer traces and hence more advanced systems may favour even longer traces, preferably of variable length.

The single state concept (word) models are, as could be expected, clearly inferior to the phone models. On the other hand, such models could be learnt automatically within the ACORNS-framework since the single state word models can be bootstrapped using only the approximate location of a word within a chunk of audio, a problem for which several techniques have already been developed in the ACORNS-project (see section 1.4.2 and 1.4.3). The error rate of the single state word models is comparable to what was obtained with the NMF method [45]. These error rates are low enough to warrant that learning algorithms on the lower and higher levels in the hierarchical memory structure can benefit from the (fuzzy) labelling done by the single state word models. See section 1.4.4 for an example on how approximate concept (word) labelling can be used when learning reusable acoustic units and [45] on how such information was used in the NMF framework to extract information on word ordering (a language model).
The word recognition results must also be put in perspective. The results in table 7 were obtained on a single task: the ACORNS year 2 Dutch database. This database has, by design, a fairly unique content and is characterised by the following properties:

- little variation in phonetic context,
- strict adherence to a very constrained CFG,
- few speakers (10), with much data per speaker

- little variation in environment and voice characteristics (e.g tempo) since everything is recorded in a few highly controlled sessions
- no background noise or music, back channel talk, ...

As was already observed in the phone recognition task, these are optimal working conditions for an exemplar based system.

3.5 Non-Euclidean distances

The experiments presented in the previous section prove a key assumption in the episodic approach: a readily accessible episodic memory and the capability to keep track of a limited number of good matches combined with a (simple) inferential system can replace the precomputed (probabilistic) models typically used today.
However, if the episodic memory is to compete with or even outperform state-of-the-art HMM systems when only a limited number of examples are available, then a more powerful inferential system is needed. In this section, two techniques to improve the inferential system are proposed and evaluated.

3.5.1 Local probability density functions

Even the most simple non-parametric density estimators are known to converge to the true density functions if sufficient examples are available for all relevant *situations* [81]. In the case of speech recognition, a *situation* is characterised by the current phone, the surrounding phones (context-dependency), some speaker characteristics such as gender and accent or dialect and some properties of the environment (signal-to-noise ratio, reverberation, ...).
By construction (few speakers, limited vocabulary, strict grammar, fixed environment), the ACORNS year 2 Dutch train data provides sufficient examples for all situations present in the corresponding test data.
Most other databases (including the Timit database) do not fulfil the above condition. It is also very questionable if humans, especially children, have observed enough speech in different conditions to ensure good coverage over all relevant situations in their memory. In other words, in most realistic situations, data sparseness will be a major problem. Hence, additional methods are needed to assure that the non-parametric density estimator used in our episodic model can cope with data sparseness.
A generic method to reduce the data sparseness problem which is applicable to both parametric (e.g. the gaussian mixtures typically used in HMM systems) and non-parametric probability estimators alike is the extraction of features that are robust with respect to variability. This is typically achieved by normalising the features. Examples of this technique are: cepstral mean normalisation, vocal tract length normalisation, noise masking, spectral subtraction, and speaker normalisation. These techniques reduce the problem, but they do not solve it completely.
A consequence of the data sparseness problem in the high dimensional feature spaces used in the k-NN based method ($36 \times L$, L being the trace length)[3] is that most observations will be located in the at or near the extremes of the class's samples store in memory, i.e. the surrounding near neighbours from a certain class will not be evenly distributed around

[3]The intrinsic dimensionality of the speech feature space is known to be smaller (in between 6 and 13, depending on the phone) than the mathematical feature dimensionality [64]. However, this dimensionality, especially when working with short traces which span multiple phones, is still high enough to assure that most observations will at or near the extremes of the class's samples stored in memory.

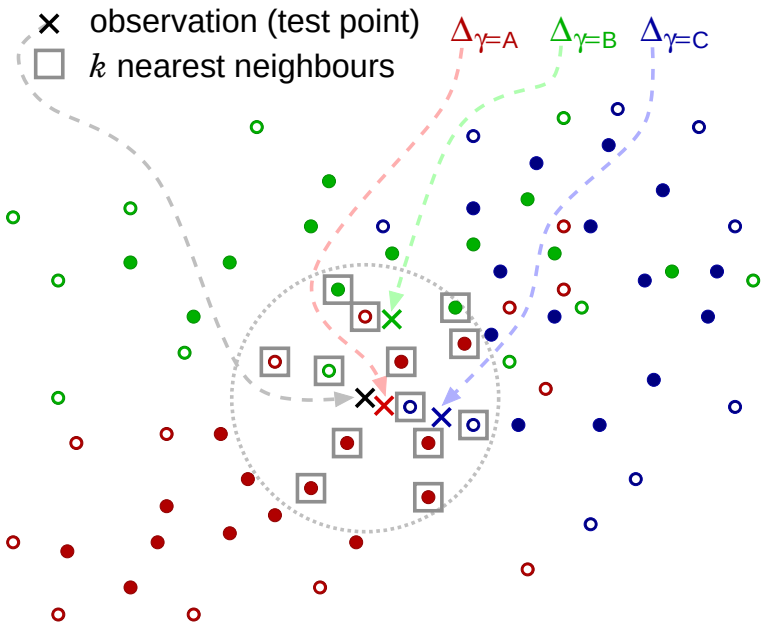


Figure 12: Adjusting the likelihood for points located at or near the extremes of the sample distribution: the hollow points are said to be located at or near the extremes (i.e. not in the bulk) of the sample distribution of their respective class; the local class means Δ_γ (local means based on the K nearest neighbours only) give a good indication whether or not the test point is located at or near the extremes of the sample distribution for that class.

the observation (show a far higher concentration at one side). Such difficult behaviour of high-dimensional data is typically referred to by the term "curse of dimensionality" [7].
Despite its intuitive and methodological importance, there is no natural basis for defining the extremes and quantiles of multivariate distributions [6]. Multiple attempts to explicitly define the idea of multivariate extremes can be found in [6]. For this work, a test point y is said to be at or near the extremes of the sample distribution for a class γ with sample points $\{x_1^\gamma, \ldots x_N^\gamma\}$ if y lays on or near by the convex hull [28] for the set of points $\Omega_\gamma = \{y, x_1^\gamma, \ldots x_N^\gamma\}$. Figure 12 illustrates this for a 3 class problem in a 2 dimensional space. For each class, the extreme samples and those in the bulk of the sample distributions are drawn as hollow and filled points respectively.
In order to make the non-parametric density estimators more robust given the data sparseness problem, we improved upon how exemplars and/or observations near the extremes are treated. Instead of just counting how many examples of a certain class are amongst the k nearest neighbours, we also investigate whether the observation is located at or near the extremes of the sample distribution for that class and how "extreme" the observation is.
Two distinct implementations of this principle were analysed. In the first implementation K, $K \gg k$ nearest neighbours are collected for each test point y. Next, for each of the classes γ present in the k nearest neighbours, a mean exemplar for that class is constructed and a class weight is calculated based on its location with respect to the test point y. Let x_i be the K nearest exemplars ordered from nearest ($i = 1$) to most distant ($i = K$), and let c_i be the corresponding classes and let Ω_γ^k and Ω_γ^K be the sets of indices $i = 1 \ldots k$ or K for which $c_i = \gamma$ and let $|\Omega_\gamma|$ be the number of elements in that set, then the class weight

Acoustic model	PER	Ins	Del	Sub
k-NN, $L = 1$, 1 state	28.40	3.46	9.25	15.69
k-NN, $L = 3$, 1 state	28.26	4.00	8.43	15.83
k-NN, $L = 5$, 1 state	27.86	3.42	9.05	15.39
k-NN, $L = 9$, 1 state	27.33	3.37	8.93	15.03
k-NN, $L = 13$, 1 state	27.46	3.86	8.33	15.27
k-NN, $L = 19$, 1 state	27.87	3.26	9.45	15.16

Table 8: Timit phone recognition results using single state HMM's with non-parametric density estimates, $k = 15$, $K = 255$, $\alpha = 1.0$.

w_γ and the class posterior p_γ are calculated according to the following equations:

$$\begin{aligned}
\Delta_i &= x_i - y \\
\sigma^2 &= \text{diag}\left(\sum_{i=1}^{K} \Delta_i \times \Delta_i^T\right) \\
\Delta_\gamma &= \frac{\sum_{i \in \Omega_\gamma^K} \Delta_i}{|\Omega_\gamma^K|} \\
w_\gamma &= \exp\left(-\alpha \left(\frac{\Delta_\gamma}{\sigma}\right)^T \times \left(\frac{\Delta_\gamma}{\sigma}\right)\right) \\
p_\gamma &= \frac{|\Omega_\gamma^k| w_\gamma}{\sum_{\gamma'} |\Omega_{\gamma'}^k| w_{\gamma'}}
\end{aligned}$$

The first two calculated values (Δ_i and σ^2) are used to transform the neighbourhood spanned by the K points to a unity circle centred around the origin. Next, the local class means Δ_γ are calculated. This is illustrated in figure 12. The remainder of the equations assure that classes for which the selected exemplars are nicely spread around the test point y (point y falls in the middle of the class) get a weight close to 1.0 while classes for which the selected exemplars are all located at one side of y will get a weight near $\exp(-\alpha)$.
The value of α showed to be non critical: the error rates changed only 1% relative when changing α to 0.5 or 2.0 from it optimal value of 1.0. Also note that the use of the larger neighbourhood K is only needed to estimate how the points of a certain class are distributed near the test point y and hence for calculating the class weight. The class posteriors and the related conditional likelihoods are still calculated based on a small neighbourhood k.
A similar strategy –calculating local class means– was adopted in [57] to derive robust predictions. However, instead of focusing on the current frame, they calculate the expected class mean for the next frame. Our method can also be seen as a variant of the "kernel difference-weighted k-nearest neighbour classification" presented in [98], with the constrained least-squares optimisation problem replaced by an intuitive yet robust "goodness of fit" measure. Our method can also be seen as a light-weight approximation of local likelihood regression [54].
Table 8 shows the results obtained using this scheme for coping with the data sparseness problem. Using this technique alone bridges half of the gap between the results of the simple k-NN system (table 6) and that of the baseline HMM systems. In fact, the results

in table 8 already outperform most of the results published on the Timit database (see table 4). Furthermore, traces longer than a single phone are no longer detrimental to the results, indicating that the problem of data sparseness is indeed counteracted.
The need to calculate $K \gg k$ near neighbours and to calculate several mean vectors Δ_γ makes the above described method somewhat less appealing due to the computational overhead and the fact that there is no guarantee that for each class present in the k nearest neighbours, sufficient samples will be present in the larger set of K nearest neighbours to reliably estimate the local class mean. However, by slightly changing the framework, the mean computations can be moved to the learning phase, entailing that only small nearest neighbour problems need to be handled during the recognition phase and guaranteeing that each local mean is based on a sufficiently large set of sample points. If next to each exemplar x_i one also stores a corresponding mean vector μ_i calculated as the mean of the K ($K \approx k$) nearest neighbours (including the vector x_i self), then an exemplar specific weight can be derived as follows:

$$\begin{aligned} w_i &= \exp\left(-\alpha \left(\frac{\mu_i - y}{\sigma}\right)^T \times \left(\frac{\mu_i - y}{\sigma}\right)\right) \\ p_\gamma &= \frac{\sum_{i \in \Omega_\gamma^K} w_i}{\sum_{i=1}^K w_i} \end{aligned}$$

This alternative implementation is expected to behave similarly to the original implementation, and evaluating it was not a priority. The scheme is also very similar to the data sharpening procedure proposed in [16, 15]. The complete setup can also be seen as a generalised kernel estimator [86] where the local kernels are skewed toward the class mean. In any case, the net result is that better usage is made of the available exemplars. Also note that for outliers the mean vector μ_i will differ substantially from the data vector x_i, ensuring that the impact of outliers (e.g. mislabelled data) is close to zero.

3.5.2 Local sensitivity matrices

A second technique for reducing the impact of the data sparseness is to improve the distance metric: not all classes show the same distribution and which distinctions are important may even depend on the context, hence the distance metric should be made adaptive [55].
In [18, 15] class dependent metrics are investigated. The disadvantage of such a scheme is that different metrics are used based on the class of the exemplar one compares with. In order to not give an unfair advantage to classes with a large spread (covariance), the calculated distance must be offset by the Jacobian (the determinant of the covariance matrix). Class dependent metrics are also a hindrance to all known techniques for speeding up the k nearest neighbours problem (see also section 4). Both problems become even more pronounced when using short traces since a trace may span several classes (phones).
We therefore pursue an alternative approach: the distance metric is made dependent on the input data and not on the class of the exemplar one compares with. This requires a module that assigns a (diagonal) sensitivity matrix to each input frame. One possible approach would be using an auditory model to derive these local sensitivity matrices [79, 47, 78]. An alternative is to do some fuzzy clustering of the data space and assign sensitivity matrices to each cluster.
We opted for the latter method. We created a single gaussian model for each of the 554 context dependent states in the baseline model. Each gaussian spans a part of the data space

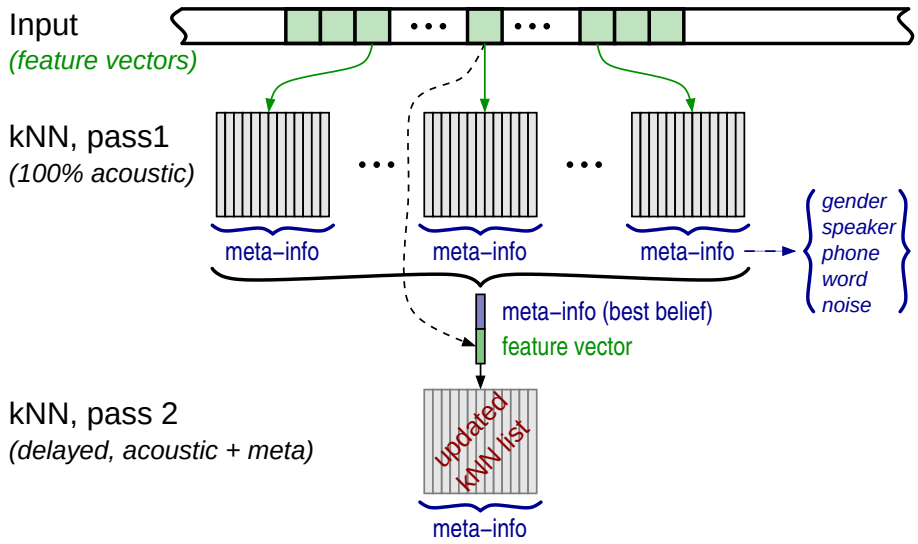


Figure 13: Using belief propagation for imposing continuity constraints.

and at the same time represents the local sensitivity for that part of the space. Since the gaussians overlap heavily, fuzzy clustering based on the posterior probability was used. Let λ_i, μ_i and Σ_i be the a priori probability, the mean vector and the diagonal covariance matrix of state i respectively, then the local sensitivity Σ_y (expressed as a diagonal covariance matrix) is calculated as follows:

$$
\begin{aligned}
w_i &= \frac{\lambda_i \mathcal{N}(y; \mu_i, \Sigma_i)}{\sum_j \lambda_j \mathcal{N}(y; \mu_j, \Sigma_j)} \\
\Sigma_y &= \sum_i w_i \Sigma_i
\end{aligned}
$$

Simulations in Matlab on the artificial problem used in [14] and on limited amounts of speech data showed promising results. Due to lack of time, the approach has not yet been ported to the episodic speech recogniser.

3.6 Belief propagation

The episodic model thus far lacks strong continuity constraints, i.e. there are no constraints on gender transitions, matching phonetic context, or even simple left-to-right traversal of the phone itself when decoding the input.

The time synchronous exemplar-based matching system of Mathias De Wachter (see section 2) promotes continuity by means of transition cost and a two dimensional Viterbi decoding. This however proves to be extremely costly. Continuity constraints can also be incorporated through (loopy) belief propagation [69, 70, 96]. Loopy belief propagation provides a solution to the problem of inference on graphical models, i.e. any model for which the conditional (in)dependency structure between the random variables can be expressed as a graph.

Once the posterior probabilities on the meta information given the continuity constraints are known approximately, the k-NN lists can be updated taking into account the inferred knowledge on the meta-data. This scheme is depicted in figure 13. The overall procedure

has a lot in common with expectation maximisation: a complex global optimisation problem (optimisation with continuity constraints) is solved by storing the current belief (belief propagation) followed by local optimisations (updating the k-NN lists).
Given that the interdependencies imposed by higher level knowledge such as lexicon and language model can be expressed in a graphical model, one could envisage a scheme where the (probabilistic) belief on the higher levels (e.g. word strings) is back-propagated to the lower levels, leading again to an update of the k-NN lists and in turn to a reinforcement (or change) in belief at the higher levels. This setup is very similar to the "hierarchically structured memory of frequently observed temporal sequences with associated labels, combined with top-down predictions" set forth in the memory-prediction framework [38]. Such a scheme could also explain phenomena such as the fact that humans tend to "hear" all phones in a word even in heavily reduced speech [33, 71, 93], or the fact that humans restore masked speech sounds without being able to identify accurately which phones were disturbed [94, 95]. The experiments in [35] show that even semantic knowledge is used for this "restoration" process.
Another promising application of belief propagation in our episodic model is the incorporation of uncertainty decoding [41, 48]. At first, the parts of the audio signal (spectrogram) masked by noise can be treated as unknown (missing). After belief propagation, the uncertain part will gradually be filled in using the information propagated back from the surrounding frames and from higher levels in the memory structure. This would again be in line with human behaviour since humans tend to hear the (reconstructed) speech and the noise as separate sources [13]. Humans believe they hear the missing sound when a sound is masked by noise, but they do notice the gap when a sound is replaced by silence [94].
Belief propagation can be linked to procedural knowledge as well. As is postulated in the paper "The Case for Case-Based Automatic Speech Recognition" [58], a key to more robust and well performing inference may lay in making more "intelligent" use of the expert knowledge (typically of procedural nature) underlying "cognitive architectures". The k-NN system itself is underpinned by the general principle (knowledge) that similar observations tend to result in similar outcomes. Another generally applicable principle is that of continuity: most phenomena are continuous, the frequency of discontinuities (e.g. speaker changes, the slamming of a door) is substantially lower. Loopy belief propagation offers an efficient mechanism to exploit such procedural knowledge either on the primary output of the k-NN based likelihood estimator –shortlists of acoustic units with their corresponding conditional likelihoods– or on any other aspects that can be inferred from the data.
Although belief propagation seems very promising, including it in our current episodic model was beyond the scope of this investigation. Furthermore, the benefits of imposing continuity constraints in an exemplar based system were already clearly demonstrated by the work of Mathias De Wachter [19, 15]. His work showed relative improvements of 11% on the Timit database and 34% on the Wall Street Journal corpus. Requiring matching right and left phone contexts was by far the best performing constraint. Limiting gender transitions only helped marginally.

4 The roadmap algorithm

Making the k-NN search computationally feasible requires a method to quickly find the relevant frames or short traces from the reference data given some new input data. In other words, we need to have some sort of associative memory capable of retrieving the relevant

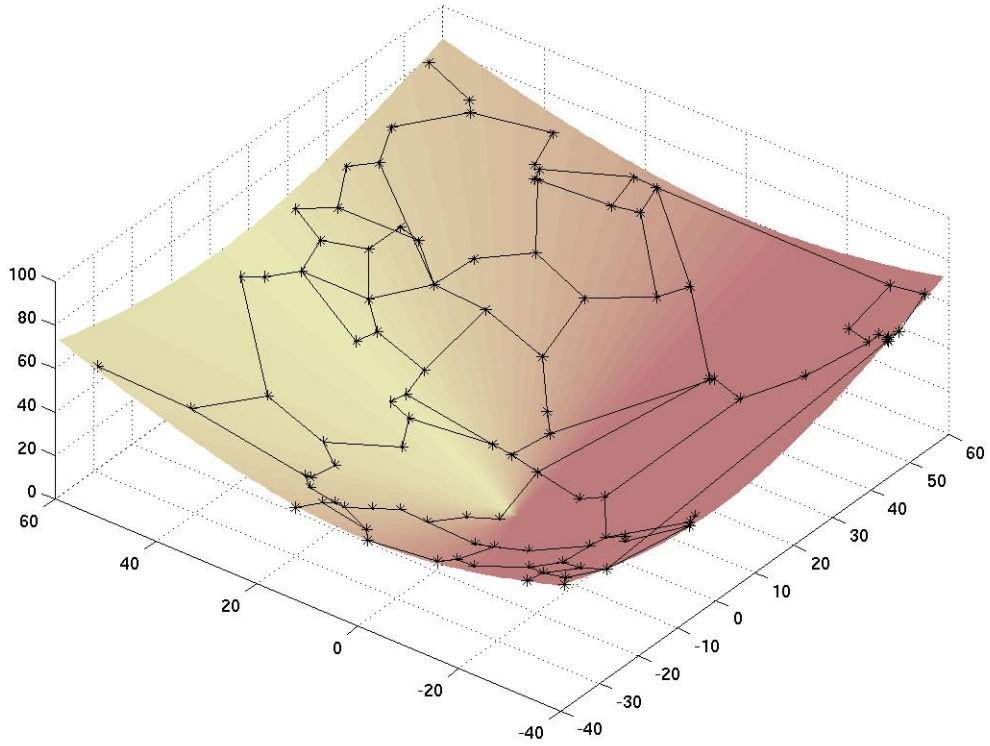


Figure 14: The roadmap algorithm can be seen as a discrete variant of the gradient descent algorithm.

chunks of data from the reference data (episodic memory) in $O(1)$ or at least $O(\ll N)$ time. The roadmap algorithm may provide this.

The basic principle behind the roadmap algorithm is depicted in figure 14. The name "roadmap" stems from the algorithm's original purpose: mapping the shortest road between two points, e.g. in a GPS device. Its usefulness for fast nearest neighbour search in the domain of speech recognition was first recognised by Povey [72]. The roadmap algorithm can also be seen as an extension of the Spatial Approximate Sample Hierarchy algorithm proposed by Houle [42] which in turn is an extension of the more traditional tree-based index structure schemes.

The roadmap algorithm can be seen as a discrete version of the gradient descent algorithm. The possible steps one can take at each point in the graph (one iterations in the algorithm) are limited to an a priori created list of *interesting* (near) neighbours for that point. In the simplified case of searching for a point in the reference database when using Euclidean distances, a perfect roadmap, i.e. a roadmap which guarantees finding the destination point when at each iteration one just jumps to the first neighbour which lowers the distance, can be constructed (see for example the algorithm in appendix B). By adding some back-trace potential [72, 15], the three main constraints of the basic algorithm –Euclidean distance, reference points only, and only finding the single best near neighbour– are readily solved.

We investigated the properties of the basic roadmap algorithm on two types of data: random (normal) distributed data and speech data. For the speech data, we used either a random

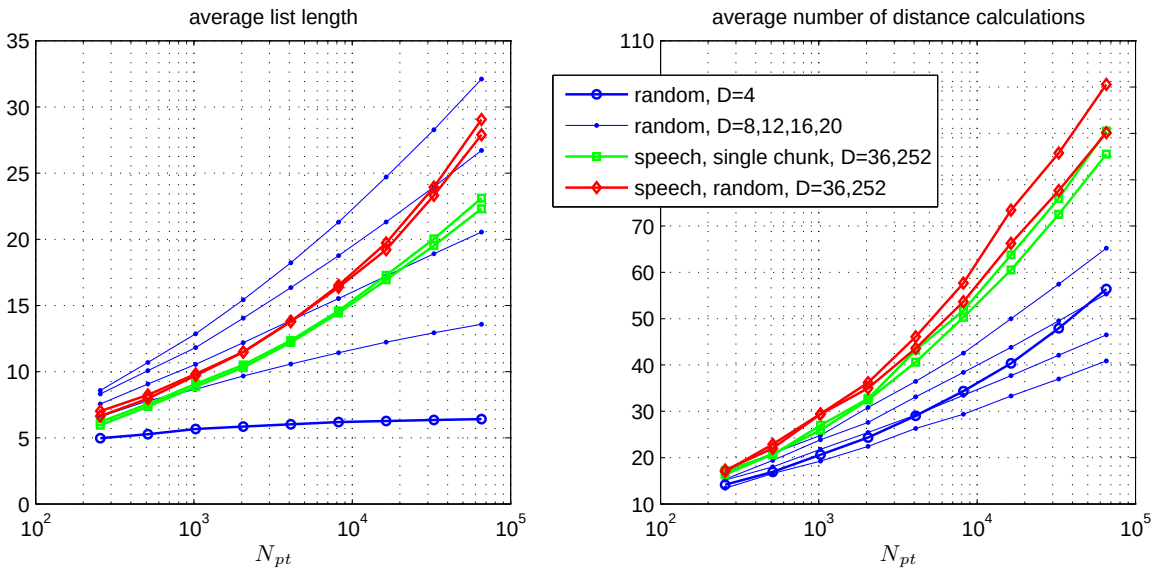


Figure 15: Main characteristics of the roadmap algorithm.

selection of data (called "random") or a continuous block of speech from a single speaker (called "single chunk"). The algorithms used are listed in appendix B).

Figure 15 shows the main characteristics, memory usage and computation time, of the algorithm. The average list length converges to $2 \times D_{\text{int}}$ with D_{int} the intrinsic dimensionality of the data. The evaluation speed was observed to be proportional to $D_{\text{int}} \times \sqrt[D_{\text{int}}]{N_{\text{pt}}}$. As was noted earlier, different phones are known to have different intrinsic dimensionalities. This is expressed in the "mixed" behaviour of the speech data. The difference between the two sets of speech data lays in the content of the data: either taken randomly over all speakers or taken as a single continuous block from the data of one speaker. A real database is expected to behave like the "single chunk" curves. The "random" speech curves should be regarded as worst case behaviour since the added variability (little data per speaker, non-consecutive data) artificially increases the intrinsic dimensionality.

The use of traces (7 frames of 36 features, resulting in vectors of dimensionality 252) seems to have only a limited effect on the intrinsic dimensionality and hence on the main properties: speed and memory. This proves that even for high-dimensional data such as traces, the roadmap can be applied successfully.

Creating a perfect roadmap using the ROADMAPCREATE algorithm from appendix B is very costly: $O(2\ D_{\text{int}}\ D_{\text{data}}\ N_{\text{pt}}^2)$ with D_{int} and D_{data} the intrinsic and the mathematical dimensionality of the data and N_{pt} the number of frame in the database. An extra problem in a self learning system is that new points must be added dynamically. Both problems can be solved by constructing/updating the roadmap iteratively. Since most if not all connections in the roadmap will be to close neighbours, step 2–4 in the ROADMAPCREATE algorithm can be replaced by finding a small set of near neighbours using the roadmap from the previous iteration. Based on this principle, fast methods to create and update a roadmap can be devised. Similar approaches were used in [72] and [15], although there the number of data points is fixed while at each iteration the roadmap is improved for some randomly selected point.

5 Conclusions & future work

In this deliverable, we presented a computational model based on episodic memory capable of doing the low-level acoustic decoding. Key aspects of the architecture are:

- a powerful content addressable memory
- an efficient inference engine
- (loopy) belief propagation as the preferred method for propagating information and constraints

The roadmap algorithm forms the basis of the robust content addressable memory that can cope with large amounts of potentially incomplete (fuzzy) high dimensional data. The intelligent interconnection of the memory cells (data points) with their "interesting" neighbours allows for a quick traversal of the data space, and hence allows for quick access to the memory given new input data. The fact that a cell is either directly or indirectly interconnected with all surrounding cells also allows for checking a hypothesis (verification) by checking the consistency with the surrounding cells. This verification strategy forms the basis of the "local probability density functions" approach (part of the inference engine).
The task of the inference engine is to extract the maximal amount of useful information from the shortlist (k nearest neighbours) returned by the content addressable memory. While the proposed inference engine still has the simplicity and elegance of a k nearest neighbour based likelihood estimator, several enhancements were proposed to increase its information extraction efficiency:

- The use of traces –a short sequence of frames of a fixed and pre-defined length carrying the same meta-information as the central frame– as basic units in the k-NN computation improves the inference since it automatically takes the acoustic context into account.
- The use of local sensitivity measures that indicate how significant acoustic differences in the neighbourhood of the current observation are given the set of acoustic units or given some generic model of the human auditory system.
- The use of local probability density functions. This technique adjusts the class likelihoods based on knowledge about the local neighbourhood: the more evidence there is that the observation falls in the bulk of a class distribution, the higher the likelihoods will be.

The k-NN mechanism reflects a generic principle used in most cognitive architectures: similar input and conditions will lead to similar results. Loopy belief propagation provides an efficient way to exploit this principle even further or to incorporate other generic principles either on the primary output of the k-NN based likelihood estimator –shortlists of acoustic units with their corresponding conditional likelihoods– or on any other aspects that can be inferred from the data. By using loopy belief propagation, continuity constraints or higher level information such as word identity or the gender of the speaker can be exploited to refine the k-NN lists, adding activation verification steps to the inference process.
Belief propagation can be from/to nearby frames (horizontal information flow) and from/to layers higher up or lower down the hierarchy (vertical information flow). Once the vertical information flow incorporates belief propagation, the overall recognition setup is seen to match Hawkins' "memory-prediction framework" very well.
From a "memory architecture" point of view, the system relies primarily on episodic memory for the low-level acoustic decoding. Semantic memory is needed for both labelling the audio data prior to storage in the episodic memory and when integrating the higher level

linguistic information during the final decoding. Procedural knowledge is mainly reflected in the generic principles that underlie the information exchange during belief propagation. The proposed architecture provides a competitive alternative to HMM based systems. A test setup only including a subset of the techniques listed above, was evaluated on both the ACORNS year 2 Dutch database and on the Timit database and showed competitive results with state-of-the-art HMM systems trained specifically for this task.

The fact that stored audio samples enriched with some labels (meta-data) form the main knowledge in the system opens the way to self-learning or adaptive systems. There is no need to retrain complex pre-computed models when new data becomes available, and even large changes such as changing the phone inventory or adding an extra layer of information (e.g. the dialect region of a speaker) can be handled easily. Once belief propagation is used for both bottom-up and top-down information transfer, simple learning techniques such as Hebbian reinforcement learning can be integrated into the system. Bootstrapping and refining the system could be done with techniques capable of clustering (labelling) parallel streams of data (e.g. audio and grounding information), for example spectral clustering.

The fact that all kind of meta-information can be handled in parallel with the primary labels (the acoustic units) opens up perspectives for further research as well. One could for example see how well word or syllable boundaries can be predicted without reverting to the lexicon, i.e. just based on labelled acoustic data. If the boundaries can be readily detected, this information can be propagated up and down the memory hierarchy adding extra verification to the hypotheses at each level. The same can be done for all kinds of prosodic information.

The combination of loopy belief propagation and traces (multiple frames stacked together) also opens up perspectives for noise robust processing. In a first iteration, a robust –missing data based– matching in the spectral domain can be used after which the components that were masked by noise can be imputed using belief propagation. Further iterations of belief propagation (either based on horizontal or vertical horizontal flow) can then operate in the cepstral domain while further refining the imputed spectral components.

References

[1] Guillaume Aimetti and Roger K. Moore. A computational model of preverbal infant word learning. In *International Conference on Cognitive Modeling 2009*, 2009.

[2] Guillaume Aimetti, Roger K. Moore, Louis ten Bosch, Okko Räsänen, and Unto K. Laine. Discovering keywords from cross-modal input: Ecological versus engineering methods for enhancing acoustic repetitions. In *INTERSPEECH 2009*, pages 1171–1174, Brighton, UK, September 2009.

[3] John R. Anderson and Gordon H. Bower. *Human Associative Memory*. Lawrence Erlbaum, Hillsdale, New Jersy, 1979.

[4] Michael A. Arbib. *Action to Language via the Mirror Neuron System*. Cambridge University Press, New York, September 2006.

[5] Alan D. Baddeley. *Working memory*. Oxford University Press, Oxford, 1986.

[6] V. Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A (General)*, 139(3):318–355, 1976.

[7] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1st edition, 1961.

[8] Alexander Bertrand. Zelflerende spraakherkenning via matrix-factorisatie. Master's thesis, K.U.Leuven, ESAT, 2007.

[9] Gilian Cohen. *Memory in the real world*. Lawrence Erlbaum Associates, January 1989.

[10] Nelson Cowan. *Attention and memory: An integrated framework*. Oxford University Press, 1995.

[11] Nelson Cowan. *Working memory capacity*. Psychology Press, New York, NY, September 2005.

[12] Meredyth Daneman and Patricia A. Carpenter. Individual differences in working memory and reading. *Journal of Verbal Learning & Verbal Behavior*, 19(4):450–466, August 1980.

[13] C.J. Darwin. Listening to speech in the presence of other sounds. *Philosophical Transactions of the Royal Society B*, 363:1011–1021, 2008.

[14] Joost De Tollenaere. Zelflerende spraakherkenning: akoestische eenheden en woordmodellen. Master's thesis, K.U.Leuven, ESAT, 2008.

[15] Mathias De Wachter. *Example Based Continuous Speech Recognition*. PhD thesis, K.U.Leuven, ESAT, May 2007.

[16] Mathias De Wachter, Kris Demuynck, and Dirk Van Compernolle. Outlier correction for local distance measures in example based speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume IV, pages 433–436, Honolulu, U.S.A., April 2007.

[17] Mathias De Wachter, Kris Demuynck, Dirk Van Compernolle, and Patrick Wambacq. Data driven example based continuous speech recognition. In *Proc. European Conference on Speech Communication and Technology*, pages 1133–1136, Geneva, Switzerland, September 2003.

[18] Mathias De Wachter, Kris Demuynck, Patrick Wambacq, and Dirk Van Compernolle. A locally weighted distance measure for example based speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Montreal, Canada, May 2004.

[19] Mathias De Wachter, Mike Matton, Kris Demuynck, Patrick Wambacq, Ronald Cools, and Dirk Van Compernolle. Template based continuous speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15:1377–1390, May 2007.

[20] Kris Demuynck. *Extracting, Modelling and Combining Information in Speech Recognition*. PhD thesis, K.U.Leuven, ESAT, February 2001.

[21] Kris Demuynck, Jacques Duchateau, and Dirk Van Compernolle. Optimal feature sub-space selection based on discriminant analysis. In *Proc. European Conference on Speech Communication and Technology*, volume III, pages 1311–1314, Budapest, Hungary, September 1999.

[22] Kris Demuynck, Tom Laureys, Dirk Van Compernolle, and Hugo Van hamme. Flavor: a flexible architecture for LVCSR. In *Proc. European Conference on Speech Communication and Technology*, pages 1973–1976, Geneva, Switzerland, September 2003.

[23] Kris Demuynck, Dirk Van Compernolle, and Hugo Van hamme. Robust phone lattice decoding. In *Proc. International Conference on Spoken Language Processing*, pages 1622–1625, Pittsburgh, U.S.A., September 2006.

[24] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proc. international conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 126–135, Philadelphia, PA, August 2006.

[25] Joris Driesen, Louis ten Bosch, and Hugo Van hamme. Adaptive non-negative matrix factorization in a computational model of language acquisition. In *Proc. International Conference on Spoken Language Processing*, Brighton, UK, September 2009.

[26] Jacques Duchateau, Kris Demuynck, Dirk Van Compernolle, and Patrick Wambacq. Improved parameter tying for efficient acoustic model evaluation in large vocabulary continuous speech recognition. In *Proc. International Conference on Spoken Language Processing*, volume V, pages 2215–2218, Sydney, Australia, December 1998.

[27] Jacques Duchateau, Mari Wigham, Kris Demuynck, and Hugo Van hamme. A flexible recogniser architecture in a reading tutor for children. In *Proc. ITRW on Speech Recognition and Intrinsic Variation*, pages 59–64, Toulouse, France, May 2006.

[28] R.E. Edwards. *Functional analysis: theory and applications*. Holt, Rinehart & Winston, 1965.

[29] Mark Elshaw, Guillaume Aimetti, Robin Hofe, Vicky Maier, Roger K. Moore, Michael Klein, Louis ten Bosch, Okko Räsänen, Joris Driesen, and Hugo Van hamme. Report consolidating all of the results pertaining to memory organisation and access derived in wp3. ACORNS-project deliverable D3.3, ACORNS research consortium, December 2009.

[30] Mark Elshaw and Moore Roger K. A recurrent working memory architecture for emergent speech representation. In *The Bernstein Conference on Computational Neuroscience (BCCN)*, 2009.

[31] Mark Elshaw, Moore Roger K., and Michael Klein. Hierarchical recurrent self-organising memory (h-rsom) architecture for an emergent speech representation towards robot grounding. In *Proc. Conference on Natural Computing and Intelligent Robotics*, 2009.

[32] Mark Elshaw, Vicky Maier, Roger K. Moore, Guillaume Aimetti, Louis ten Bosch, Michael Klein, Hugo Van hamme, and Okko Räsänen. Report focussing on the results of the initial asr experiments comparing episodic and semantic long term memory. ACORNS-project deliverable D3.2, ACORNS research consortium, December 2008.

[33] M. Ernestus, H. Baayen, and R. Schreuder. The recognition of reduced word forms. *Brain and Language*, 81:162–173, 2002.

[34] W.M. Fisher, V. Zue, J. Bernsein, and D. Pallet. An acoustic phonetic data base. In *presented at the 113th Meeting of the Acoustical Society of America*, May 1987.

[35] S. Garnes and Z.S. Bond. The relationship between acoustic information and semantic expectation. *Phonologica 1976*, pages 285–293, 1976.

[36] P. Gaussier, S. Moga, M. Quoy, and J.P.B. Creare. From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 12(8-9):701–727, October 1998.

[37] Stephen D. Goldinger. Echoes of echoes: an episodic theory of lexical access. *Psychological Review*, 105(2):251–279, 1998.

[38] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. Times Books, New York, NY, 2004.

[39] Donald O. Hebb. *The organization of behavior*. Wiley, New York, 1949.

[40] Douglas L. Hintzman. Schema-abstraction in a multiple-trace memory model. *Psychological Review*, 93:411–427, 1986.

[41] J.N. Holmes, W.J. Holmes, and P.N. Garner. Using formant frequencies in speech recognition. In *Proc. European Conference on Speech Communication and Technology*, pages 2083–2086, Rhodes, Greece, September 1997.

[42] M. E. Houle and J. Sakuma. Fast approximate similarity search in extremely high-dimensional data sets. In *Proceedings of the 21st International Conference on Data Engineering (ICDE2005)*, pages 619–630, 2005.

[43] Larry L. Jacoby and C. A. Hayman. Specific visual transfer in word identification. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*, 13(3):456–463, July 1987.

[44] S. Johnson. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Scribner, New York, 2002.

[45] Hugo Van hamme Joris Driesen. Implementation and test of activation-verification mechanisms. ACORNS-project deliverable D4.1, ACORNS research consortium, December 2008.

[46] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 3rd edition, 2001.

[47] C. Koniaris, M. Kuropatwinski, and W.B. Kleijn. Modules for feature augmentation and selection – feature selection based on knowledge of the auditory system. ACORNS-project deliverable D1.2, ACORNS research consortium, December 2008.

[48] T. Kristjansson and B. Frey. Accounting for uncertainty in observations: a new paradigm for robust automatic speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 61–64, Orlando, FL, U.S.A., May 2002.

[49] Unto K. Laine, Gustav Henter, Guillaume Aimetti, and Joris Driessen. Methods for enhanced pattern discovery in speech processing. ACORNS-project deliverable D2.2, ACORNS research consortium, December 2008.

[50] Unto K. Laine, Okko Räsänen, Gustav Henter, and Toomas Altosaar. Pattern discovery with discrete model elements. ACORNS-project deliverable D2.1, ACORNS research consortium, December 2007.

[51] Unto K. Laine, Okko Räsänen, Gustav Henter, and Toomas Altosaar. Pd module with self-directed search, derived segmental quality measures, full integration of cmm. ACORNS-project deliverable D2.3, ACORNS research consortium, December 2009.

[52] L.F. Lamel, R.H. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In Baumann L.S., editor, *Proceeding DARAP Speech Recognition Workshop*, February 1986.

[53] Kai-Fu Lee and Hsiao-Wuen How. Speaker-independent phone recognition using hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(II):1641–1648, November 1989.

[54] Clive R. Loader. Local likelihood density estimation. *Annals of Statistics*, 24(4):1602–1618, 1996.

[55] Viktoria Maier. *Temporal Episodic Memory Model: towards proactive case-based Automatic Speech Recognition*. PhD thesis, Department of Computer Science, University of Sheffield, (to be submitted shortly).

[56] Viktoria Maier and Roger K. Moore. An investigation into a simulation of episodic memory for automatic speech recognition. In *INTERSPEECH 2005*, pages 1245–1248, Lisbon, Portugal, September 2005.

[57] Viktoria Maier and Roger K. Moore. Temporal episodic memory model: An evolution of minerva2. In *INTERSPEECH 2007*, pages 866–869, Antwerp, Belgium, August 2007.

[58] Viktoria Maier and Roger K. Moore. The case for case-based automatic speech recognition. In *INTERSPEECH 2009*, pages 3027–3030, Brighton, UK, September 2009.

[59] Margaret W. Matlin. *Cognition*. Harcourt Brace College Publishers, Orlando, FL, 4th edition, 1998.

[60] G.A. Miller. The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 101(2):240, March 1956.

[61] Vernon Mountcastle. *The Mindful Brain*, chapter An organizing principle for cerebral function: The unit model and the distributed system, pages 7–50. MIT Press, Cambridge, MA, 1978.

[62] Bennet B. Murdock. *Human memory: Theory and data*. Lawrence Erlbaum, Oxford, England, 1974.

[63] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14:849–856, 2001.

[64] M. Nilsson and B. Kleijn. Mutual information and the speech signal. In *Interspeech 2007*, pages 502–505, Antwerpen, Belgium, 2007.

[65] K Oberauer. Access to information in working memory: exploring the focus of attention. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, 28(3):411–421, May 2002.

[66] K. Oberauer, H.-M. Sus, R. Schulze, O Wilhelm, and W. W. Wittmann. Working memory capacity – facets of a cognitive ability construct. *Personality and Individual Differences*, 29(6):1017–1045, December 2000.

[67] Andrew Ortony. How episodic is semantic memory? In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 55–59, Cambridge, Massachusetts, 1975.

[68] M. Ostendorf. Moving beyond the 'beads-on-a-string' model of speech. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pages 79–84, Keystone, Colorado, USA, December 1999.

[69] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second National Conference on Artificial Intelligence*, pages 133–136, Pittsburgh, PA, 1982.

[70] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman, San Mateo, 1988.

[71] M. Pluymaekers, M. Ernestus, and R.H. Baayen. Lexical frequency and acoustic reduction in spoken dutch. *Journal of the Acoustical Society of America*, 118:2561–2569, 2005.

[72] D. Povey and P.C. Woodland. Frame discrimination training of HMMs for large vocabulary speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 333–336, Phoenix, Arizona, March 1999.

[73] V.S. Ramachandran. Mirror neurons and imitation learning as the driving force behind "the great leap forward" in human evolution. In *Edge*, volume 69. Edge Foundation, June 2000.

[74] Okko J. Räsänen, Unto K. Laine, and Toomas Altosaar. A noise robust method for pattern discovery in quantized time series: the concept matrix approach. In *INTERSPEECH 2009*, pages 3035–3038, Brighton, UK, September 2009.

[75] Okko Johannes Räsänen, Unto Kalervo Laine, and Toomas Altosaar. An improved speech segmentation quality measure: The r-value. In *INTERSPEECH 2009*, pages 1851–1854, Brighton, UK, September 2009.

[76] Okko Johannes Räsänen, Unto Kalervo Laine, and Toomas Altosaar. Self-learning vector quantization for pattern discovery from speech. In *INTERSPEECH 2009*, pages 852–855, Brighton, UK, September 2009.

[77] T. Jeff Reynolds and Christos A. Antoniou. Experiments in speech recognition using a modular MLP architecture for acoustic modelling. *Information sciences*, 156(1-2):39–54, November 2003.

[78] Chatterjee S., C. Koniaris, W.B. Kleijn, M. Van Segbroeck, and H. Van hamme. Final modules for features derived from auditory model and a self-learning algorithm. ACORNS-project deliverable D1.3, ACORNS research consortium, December 2009.

[79] Christos Koniaris Saikat Chatterjee and W. Bastiaan Kleijn. Auditory model based optimization of MFCCs improves automatic speech recognition performance. In *INTERSPEECH 2009*, pages 2987–2990, Brighton, UK, September 2009.

[80] Roger C. Schank. Is there a semantic memory. Technical Report 3, Instituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland, 1974.

[81] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[82] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing*, 1:195–281, 1986.

[83] Gisela E. Speidel. *The Many faces of imitation in language learning*. Springer, New York, 1989.

[84] Veronique Stouten, Kris Demuynck, and Hugo Van hamme. Discovering phone patterns in spoken utterances by non-negative matrix factorisation. *IEEE Signal Processing Letters*, 15:131–134, 2008.

[85] I. Sutskever, G.E. Hinton, and G.W. Taylor. The recurrent temporal restricted boltzmann machine. *Advances in Neural Information Processing Systems*, 21, 2009.

[86] George R. Terrell and David W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1992.

[87] Endel Tulving. *Organization of memory*, chapter Episodic and semantic memory, pages 381–403. Academic Press, New York, 1972.

[88] Endel Tulving. Episodic memory: from mind to brain. *Annual Review of Psychology*, 53:1–25, 2002.

[89] Hugo Van hamme. HAC-models: a novel approach to continuous speech recognition. In *Proc. International Conference on Spoken Language Processing*, pages 2554–2557, Brisbane, Australia, September 2008.

[90] Hugo Van hamme. Integration of asynchronous knowledge sources in a novel speech recognition framework. In *Proc. ITRW on Speech Analysis and Processing for Knowledge Discovery*, Aalborg, Denmark, June 2008. 4 pages, ISBN 978-87-92328-00-7.

[91] Hugo Van hamme. Top-down learning of patterns and computation of activations. ACORNS-project milestone M4.1.2, Katholieke Universiteit Leuven, December 2008.

[92] Hugo Van hamme. Asms defined from wp1 and wp2 features and automatic segmentation. ACORNS-project milestone M4.2.2, Katholieke Universiteit Leuven, December 2009.

[93] Natasha Warner, Amy Fountain, and Benjamin V. Tucker. Cues to perception of reduced flaps. *Journal of the Acoustical Society of America*, 125:3317–3327, 2009.

[94] Richard M. Warren. Restoration of missing speech sounds. *Science*, pages 392–393, 1970.

[95] Richard M. Warren. *Auditory Perception: A New Analysis and Synthesis*. Cambridge Univ Press, Cambridge, UK, 1999.

[96] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41, January 2000.

[97] Niklaus Wirth. What can we do about the unnecessary diversity of notations for syntax definitions? *Communications of the ACM*, 20(11):822–823, November 1977.

[98] Wangmeng Zuo, David Zhang, and Kuanquan Wang. On kernel difference-weighted k-nearest neighbor classification. *Pattern Analysis & Applications*, 11(3-4):247–257, 2008.

A Grammar for the year 2 Dutch setup in Wirth syntax notation

```
!grammar acorns_nl_y2 ;

!start <_main_> ;

<_main_>       :== <corrective> | <statement> ;

<corrective>   :== nee ik ( zei | bedoel )
                    ( <pers_adj_mv> | <persoon_mv> | <obj_adj_mv>
                    | <obj_adj_n> | <object_mv> | <object_n> | zie | ) ;

<statement>    :== ( <iemand> ( ziet | pakt | kijkt naar
                              | neemt | geeft | heeft )
                   | ( zie | heb ) je
                   | ( hier | waar | daar ) is
                   ) <object> [en <object>] ;

<iemand>       :== hij | zij | ( de | een ) [<pers_adj_mv>] <persoon_mv> ;

<object>       :== ( de | een ) [<obj_adj_mv> [<obj_adj_mv>]] <object_mv>
                 | ( het | een ) [<obj_adj_n> [<obj_adj_n>]
                 | <obj_adj_mv> [<obj_adj_mv>]] <object_n> ;

<pers_adj_mv> :== blije | boze | huilende | lachende ;

<persoon_mv>   :== mama | papa | vrouw | man | baby ;

<object_mv>    :== appel | auto | bal | banaan | boom | doos | duif
                 | eend |fles | hond | kat | kikker | koe | leeuw | pop
                 | porsche | telefoon | vis | vogel | vrachtwagen ;

<object_n>     :== dier | koekje | paard | roodborstje | vliegtuig ;

<obj_adj_mv>   :== bedroefde | blauwe | blije | eetbare | gele | grote
                 | kleine | rode | ronde | schone | vierkante | vieze ;

<obj_adj_n>    :== bedroefd  | blauw  | blij  | eetbaar | geel | groot
                 | klein  | rood | rond  | schoon | vierkant  | vies ;
```

B The roadmap algorithm as used for evaluating it properties

Algorithm 1 ROADMAPCREATE($\bar{x}_{1...N}, \gamma, \epsilon$)

% Input: $\bar{x}_{1...N}$: N data points
% Output: $L_{1...N}$: the neighbour map (an ordered list of 'direct' neighbours per data point $\bar{x}_i$)
for $i = 0 \ldots N$ **do**
  for $j = 0 \ldots N$ **do**
    $D_j \leftarrow ||\bar{x}_j - \bar{x}_i||^2$
  end for
  $L_i \leftarrow []$
  for j in argsort(D) **do**
    if $\forall k$ in $L : ||\bar{x}_j - \bar{x}_k||^2 > \gamma D_j - \epsilon$ **then**
      $L_i \leftarrow [L, j]$
    end if
  end for
end for

Algorithm 2 ROADMAPFIND($\bar{x}_{1...N}, L_{1...N}, \bar{y}, k$)

% Input: $\bar{x}_{1...N}$: N reference data points
% Input: $L_{1...N}$: the neighbour map (an ordered list of 'direct' neighbours per data point $\bar{x}_i$)
% Input: $\bar{y}$: some test data point
% Input: k : initial guess (may be random) of which reference data point x_k is nearest to the test data point y
% Output: $\bar{x}_k$: the nearest neighbour to $\bar{y}$ in set $\bar{x}_{1...N}$
for $i = 1 \ldots N$ **do**
  for $j = 1 \ldots N$ **do**
    $D_j \leftarrow ||\bar{x}_j - \bar{x}_i||^2$
  end for
  $L_i \leftarrow []$
  for j in argsort(D) **do**
    if $\forall k$ in $L : ||\bar{x}_j - \bar{x}_k||^2 > \gamma D_j - \epsilon$ **then**
      $L_i \leftarrow [L, j]$
    end if
  end for
end for