Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D6.3 Open source software

Period covered: from 1 December 2008 to 30 November 2009
Start date of project: 1 December 2006 Duration: 36 months
Submission date Nov 27, 2009
Due date Dec 1, 2009
Version: final

Project coordinator name: Prof. Lou Boves
Project coordinator organisation name: Radboud University, Nijmegen
Project co-funded by the EC, FP6, 2002-2006

Dissemination level

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

High-level description of the final learning system – the ACORNS computational model of language acquisition

Version 1.5 (15.11.2009)

Created by Radboud University Nijmegen
The Netherlands

Description of the deliverable

This deliverable consists of a set of MATLAB packages that have been designed, used and tested in the ACORNS project (www.acorns-project.org). The entire package is available as a zip file from www.acorns-project.org
All the software has actually been used for experiments. Before we present the contents of the packages, we first list a number of papers that may be of interest for interpreting these packages.

NMF has been described in Van Segbroeck and Van hamme (2009), in Stouten and Van hamme (2009), and especially its incremental adaptive variant in Driesen, ten Bosch, Van hamme (2009).

DP-Ngrams is extensively described in Aimetti (2009), in Aimetti, ten Bosch, Moore (2009), and in Aimetti, Moore, ten Bosch, Räsänen, Laine (2009).

Concept Matrices and Segmentation and **Self Learning VQ** are in detail described in various papers by Räsänen, Laine, Altosaar (2009x).

Experiments with **novel features** have been described in Chatterjee, Koniaris and Kleijn (2009).

Various other experiments have been described in ten Bosch, Driesen, Van hamme, Boves (2009) (about the ability to generalize towards novel speakers), in ten Bosch, Räsänen, Driesen, Aimetti, Altosaar, Boves (2009) (about contrasting NMF, DP-Ngrams and CM on the same dataset), in Moore & ten Bosch (2009) (about human word acquisition curves), and in ten Bosch, Boves and Räsänen (2009) (about less than 'ideal' interaction strategies in both learner and caregiver).

See ten Bosch, Van hamme, Boves, Moore (2009) for a general overview of the ACORNS project.

As observed above, knowledge of the ideas underlying the experiments (as described in the literature) may be helpful in interpreting the input and output of the various software packages.

The software packages

Most of the software runs on standard versions of MATLAB, without special requirements for specialized MATLAB toolkits. The other software is written in C.

The entire ACORNS package consists of the following packages:

- CA-LA-interaction (MATLAB)
- Concept Matrices and Segmentation (C)
- SCCR_RCSD (C)
- DP_Ngrams (MATLAB)
- Features (MATLAB)
- Interaction Platform (MATLAB)
- Self Learning VQ (MATLAB)
- NMF (MATLAB)

Of these, 'Concept Matrices and Segmentation' and 'SCCR_RCSD' will not be directly available. With respect to Concept Matrices (and the segmentation), the authors will provide these algorithms as standalone packages for academic research purposes upon request. The use of software requires signing of a license agreement by the representatives of both TKK and the other party requesting permission of use. Requests of software use should be sent to Okko Räsänen (okko.rasanen@tkk.fi) or Unto Laine (unto.laine@tkk.fi).

The software SCCR_RCSD will be made available after the corresponding papers have been published (Gustav Henter, ghe@EE.KTH.SE).

CA-LA-interaction

This package contains the MATLAB scripts that have been used for the experiment described in ten Bosch, Boves and Räsänen (2009). CA and LA are abbreviations for 'Caring Agent' ('caregiver') and 'Learning Agent' ('learner'), respectively.
In *most* ACORNS experiments, we have assumed that the caregiver always present complete and consistent stimuli. Each stimulus consists of an audio part and a 'visual' (grounding) part. In the experiments so far, the learner was presented stimuli in which audio and grounding part were consistent with each other.
At the same time, the learner takes each stimulus 'as it appears'. That is, it assumes the stimulus to be consistent, and does not doubt the consistency within the stimulus.

Many scenarios can be invented that deviate from this idealized-world scenario:

- At the caregiver side: the caregiver may present stimuli that are inconsistent. The probability of presenting an inconsistent stimulus is denoted *p*. ($0 \le p \le 1$)
- At the learner's side, an *internal confidence mechanism* is active such that if the confidence about a certain hypothesis exceeds a certain threshold θ, the learner assumes that its own hypothesis is true and discards the information in the grounding section of the input stimulus. The own hypothesis is put in short term memory for later reuse. ($0 \le \theta \le 1$)

ten Bosch, Boves and Räsänen (2009) describes an experiment in which the effect of these parameters is investigated using the NMF framework.
This package contains the MATLAB scripts to run this experiment, and a description of the meaning of the various MATLAB parameters.

Concept Matrices and Segmentation

The Concept Matrix approach (and the derived segmentation) is one of the major computational approaches applied in ACORNS. The other approaches make use of Non-negative Matrix Factorisation (NMF) and DP-Ngrams, NMF and DP-Ngrams are included in the package.

The current situation (Nov 11, 2009) is that the **segmentation algorithm** is protected by a patent FI120223 ("Automatic, unsupervised segmentation of audio signals"), and patent application for **Concept Matrices** (CM) has been filed ("Method for pattern discovery and recognition"; application number FI20086260). Therefore they cannot be released as free software packages to the public audience with the other ACORNS software.

Instead, the authors will provide these algorithms as standalone packages for academic research purposes upon request. The use of software requires signing of a license agreement by the representatives of both TKK and the other party requesting permission of use. Requests of software use should be sent to Okko Räsänen (okko.rasanen@tkk.fi) or Unto Laine (unto.laine@tkk.fi).

SCCR_RCSD

The consortium decided that the Software on *Causal State Splitting Reconstruction* (CSSR) and *Robust Causal State Discovery* (RCSD) will be made publicly available once the associated paper is accepted.

The CSSR algorithm and its ACORNS-based improvement, RCSD, are both examples of algorithms in the field of computational mechanics.

Computational mechanics is a research programme with the aim of using automata theory to describe patterns and the complex, stochastic processes that generate them. Central to these descriptions is the concept of *causal states*. The causal states are equivalence classes defined over the possible *histories* (sequences of observations from negative infinity until the current time *t*) of a given stationary, discrete-time stochastic process. Two histories belong in the same causal state if and only if they give the exact same beliefs about the future, i.e., if they imply the same probability distribution over all *futures* (sequences of observations from *t*+1 and on to infinity). The causal states are thus a partitioning of the set of possible histories.
One can show that the causal state representation is a *minimal sufficient statistic* for the observation sequence; it retains precisely all information from past observations relevant for predicting the future, and nothing more. Moreover, appending a symbol to a history string gives a new history string that also belongs in some causal state. This way it is possible to define transitions between the states. Interestingly, the states and their transitions together constitute a Markov process—even if the original process is not Markovian. Unlike HMMs, the current state can, in this description, be uniquely identified from the available sequence of observations.

Causal states are a theoretical construct based on full knowledge of the underlying process. Fortunately, in practice an approximation of the causal states can be learned from one or more empirical data sequences using the so-called *causal state splitting reconstruction algorithm* (CSSR) by Shalizi, Klinkner, and Crutchfield. Given enough data, this procedure converges on the true causal states. However, the algorithm only operates on sequences of discrete symbols from a finite alphabet.

Our implementation of CSSR improves the original by Shalizi et al. The development of the robust version, RCSD, has been based on a combination of the best of both worlds of CSSR and of HMM.

Stockholm_RCSD/rcsd.zip

This zip file provides source code and a manual for the ACORNS WP2 C++ implementation of the CSSR algorithm from computational mechanics and the RCSD algorithm that improves on it.

The two algorithms build predictive finite automata representations of discrete, conditionally stationary processes, given data sequences sampled from such a process. RCSD extends CSSR and can recover the original structure of these automata even if the data has been disturbed by random insertions, deletions, and substitutions. The manual provides additional information, usage examples, and references to relevant research papers.

For details, the reader is referred to the literature (Gustav Henter, www.acorns-project.org)

DP_Ngrams

DP-Ngrams is the name of one of the methods used in the ACORNS project to computationally model the acquisition of language, by focusing on the detection of word-like elements.

In contrast to the other methods, CM and NMF, DP-Ngrams is the most episodic of the three. This has interesting consequences, both for the comparison of the performance of DP-Ngrams with the other approaches, and for the cognitive plausibility of the results. For details, the user is referred to the literature.

Knowledge about the principles of Viterbi decoding, distance matrices, alignment in ASR will be helpful (and sometimes essential) in order to properly apply and understand DP-Ngrams.

The input for the algorithm consists of a corpus file (cor file), which specifies the utterances to be processed. The corpus file is used in all algorithms (NMF, CM, and DP_Ngrams). Also a parameter initialization file is required. Examples are provided in the DP-Ngram package.

Figure 1 shows an example of how DP-Ngrams locates potentially interesting word-like elements, by aligning two utterances. Red lines in the matrix connect the sequences of vectors with low local alignment scores.

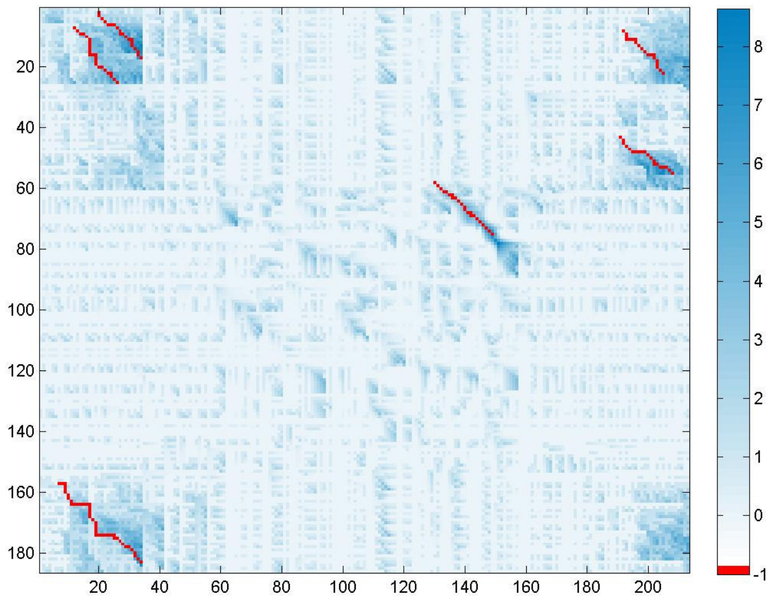

Figure 1. Example of an alignment by DP-Ngrams.

Features

In the ACORNS project we have been experimenting with features different from the MFCC features. MFCC features are almost by default chosen in all automatic speech processing, and certainly in (high band) ASR.

We have explored alternatives in two directions: Modified MFCC, and 'Static Adaptive' MMFCC. These modifications have been inspired by numerical optimisations using a computational model of human audio processing (see papers by Saikat Chatterjee on www.acorns-project.org).
Compared to MFCC, MMFCC introduces changes in parameter settings, e.g. in de definition of the MEL-frequency curve. SAMMFCC goes further than this: here entirely new features are appended to the MMFCC, resulting in 39+12=51 features.
The 39-dimensional MFCCs and MMFCCs were tested by a standard HTK system for phone and TIDIGIT recognition tasks. According to these experiments (see public deliverable D1.3), the MMFCCs provide a better performance than standard MFCCs (both for clean and noisy conditions).

The package provides both MATLAB scripts for MFCC, MMFCC and SAMMFCC.
Some basic knowledge of MATLAB and of signal processing is required to understand the underlying research line leading to these scripts. For details the reader is referred to the literature (S. Chatterjee, see also www.acorns-project.org).

Interaction Platform

The Interaction Platform has been designed to facilitate a number participants (‘caregiver’, ‘learner’) to be engaged in an interaction. An interaction can be described as a (finite) process during which information is exchanged between participants. Such an interaction may be a computational modeling of language acquisition, in which a young infant and a caregiver interact.

The Platform comes in different versions: **‘Interaction Platform 2007’, ‘Interaction Platform 2009’,** and **‘Interaction Platform 2009b’**.

The 2007 and 2009 packages contain versions of the Platform for two participants (caregiver and learner – this has been the usual set up in the ACORNS experiments). The 2007 package is the basic one that was taken as starting point for follow-up versions. The 2009 package provides an explicit example of NMF being integrated into the platform (the availability of a database is required).

The 2009b package contains an implementation for three participants. This 2009b version can run without the availability of an actual database, and probably best demonstrates the functionality of the Platform itself.

All ACORNS experiments are based on the use of models for word learning or language acquisition. The learning of a language (and of a new word) takes place within an interactive loop between caregiver and learner. Independent the nature of the messages that the learner and caregiver exchange, the way the interaction takes place already poses quite some constraints on the types of learning that can take place, and on the natural ness of the learning simulation.

For example, in human-human interaction, it is common that people interrupt each other. This is obvious from studies on telephone conversations. Interruption is a natural property of human-human interaction. A second, also prominent, property of human-human communication is the fact that information from speaker to listener may be on several distinct levels (‘backchannel’, shallow disagreement/agreement, fully-content-based). These levels have a different role in maintaining the naturalness and spontaneity in a human-human dialogue (see e.g. studies by Kristinn Thorisson on human-human dialogues).

The Platform provides a way for the learner and caregiver to interact with each other in various ways. A simple example is message-passing, such that each party waits until the other party is ready with processing the previous input and has provided a result (next output). A more complex example would use interruptions and different levels of messages from learner to caregiver and reverse.

It is important to say that the Platform does not bring new functionality to the content of the dialogue between learner and caregiver. It is a facilitating module, able to pass around complex messages, able to mainly a complex logging. The Platform does not limit the number of participants in an interaction.

Self Learning VQ

The VQ package contains one zip file SLVQ_v01.zip that includes the Self Learning Vector Quantization algorithm and a readme file for documentation. The Self Learning Vector Quantization (SLVQ; Räsänen et al., 2009) is an incremental and computationally effective clustering method used for vector quantization in WP2 word learning experiments. The SLVQ does not require the overall number of clusters as a parameter, but adjusts to the properties of the training data according to the learning rate defined by the user, making it plausible for unsupervised learning applications.

As said, the zip file contains a readme file. This readme file gives explicit examples of a MATLAB call to the SLVQ. The user must have global knowledge about VQ procedures in general in order to properly use and interpret the various parameters.

NMF

Non-Negative Matrix Factorisation (NMF) has been one of the major three computational approaches in ACORNS. The other two approaches are DP-Ngrams and Concept Matrices.

NMF is a technique based on linear algebra. NMF decomposes a large matrix **X** as a product **WH**. The matrices **X**, **W** and **H** all have non-negative elements. Usually the number of columns of **W** and the number of rows of **H** in NMF are selected so the product **WH** will become an approximation to **X**. The full decomposition of **X** then amounts to the two non-negative matrices **W** and **H** as well as a residual **U**, such that: **X** = **WH** + **U**. The elements of the residual matrix can either be negative or positive.

In ACORNS, NMF is used to decompose a large data matrix into two smaller matrices. NMF takes place inside the learner. The matrix **W** resembles the internal representations for word-like elements inside the learner (in the learner's long-term memory).

There are different types of non-negative matrix factorizations. The different types arise from using different cost functions for measuring the divergence between **X** and **WH** and possibly by regularization of the **W** and/or **H** matrices. Two simple divergence functions studied by Lee and Seung are the squared error and an extension of the Kullback-Leibler divergence to positive matrices (the original Kullback-Leibler divergence is defined on probability distributions). Each metric leads to a different NMF algorithm. In ACORNS we have used the extended KL distance, in nearly all experiments.

The original description of NMF was batch-based. Joris Driesen and Hugo Van hamme have made an *incremental* version of the original batch-NMF. This incremental version is available in this package, and is implemented in the learner module. The package presents caregiver and learner and their communication, implemented using the ACORNS Interaction Platform.

General remarks

The software packages have been used for performing experiments in a research project. Often, a certain knowledge of the ideas underlying the experiments is helpful (even necessary) to understand the settings of parameters and the input-output relations of the algorithms.
Also knowledge of MATLAB is essential. The use of toolkits is not required.
Some of the specialized algorithms need knowledge about the underlying methods. The literature lists provided may be of help in such cases.
The ACORNS data that have been used in the ACORNS project will be distributed via another channel see (www.acorns-project.org). Some but not all packages provide a small number of ACORNS (audio) files to show the principle of an algorithm. In general, we expect the collaborative user to be able to find her way, be able to accommodate MATLAB scripts to local circumstances and data, and hopefully these implementations can be of great help.

ACORNS references, with links to pdfs

Louis ten Bosch, Hugo Van hamme , Lou Boves, Roger K. Moore "A computational model of language acquisition: the emergence of words", *Fundamenta Informaticae*, Vol. 90, (2009), pp. 229-249*,* [pdf]

Maarten Van Segbroeck and Hugo Van hamme "Unsupervised learning of time-frequency patches as a noise-robust representation of speech", *Speech Communication*, volume 51, pp. 1124-1138, 2009 [pdf]

Veronique Stouten and Hugo Van hamme "Automatic voice onset time estimation from reassignment spectra", *Speech Communication*, Vol. 51, pages 1194-1205, 2009 [pdf]

Guillaume Aimetti "Modelling Early Language Acquisition Skills: Towards a General Statistical Learning Mechanism", *Proc. EACL-2009*, [pdf]

Okko Räsänen & Joris Driesen "A comparison and combination of segmental and fixed-frame signal representations in NMF-based word recognition", *Proc. 17th Nordic Conference on Computational Linguistics,* 2009 [pdf]

Louis ten Bosch, Joris Driesen, Hugo Van hamme, Lou Boves "On a computational model for language acquisition: modeling cross-speaker generalisation", *Proc. Text, Speech and Dialogue, 12th Intern. Conference, TSD 2009* [pdf]

Guillaume Aimetti, Roger K. Moore, Louis ten Bosch, Okko Räsänen, Unto K. Laine "Discovering Keywords from Cross-Modal Input: Ecological vs. Engineering Methods for Enhancing Acoustic Repetitions", *Proc. Interspeech 2009* [pdf]

Roger K. Moore, Louis ten Bosch "Modelling Vocabulary Growth from Birth to Young Adulthood", *Proc. Interspeech 2009* [pdf]

Okko J. Räsänen, Unto K. Laine, Toomas Altosaar "A noise robust method for pattern discovery in quantized time series: the concept matrix approach", *Proc. Interspeech 2009* [pdf]

Okko J. Räsänen, Unto K. Laine, Toomas Altosaar "An Improved Speech Segmentation Quality Measure: the R-value", *Proc. Interspeech 2009* [pdf]

Okko J. Räsänen, Unto K. Laine, Toomas Altosaar "Self-learning Vector Quantization for Pattern Discovery from Speech", *Proc. Interspeech 2009* [pdf]

Viktoria Maier, Roger K. Moore "The Case for Case-Based Automatic Speech Recognition", *Proc. Interspeech 2009* [pdf]

Saikat Chatterjee, Christos Koniaris and W. Bastiaan Kleijn "Auditory Model Based Optimization of MFCCs Improves Automatic Speech Recognition Performance", *Proc. Interspeech 2009* [pdf]

L. ten Bosch, O. Räsänen, J. Driesen, G. Aimetti, T. Altosaar, L. Boves, A. Corns "Do Multiple Caregivers Speed up Language Acquisition?" *Proc. Interspeech 2009* [pdf]

Joris Driesen, Louis ten Bosch, Hugo Van hamme "Adaptive Non-negative Matrix Factorization in a Computational Model of Language Acquisition", *Proc. Interspeech 2009* [pdf]

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

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

Louis ten Bosch, Lou Boves and Okko Rasanen "Learning meaningful units from multimodal input – the effect of interaction strategies", *Proc. Wocci2009*, [pdf]

Guillaume Aimetti, Louis ten Bosch, Roger K. Moore "The emergence of words: Modelling early language acquisition with a dynamic systems perspective", Proc. EpiRob-09, [pdf]