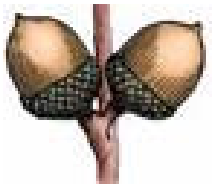

Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

D2.1 PD and DME Modules

Due date of deliverable: 2007-12-31
Actual submission date: 2007-12-21
Post review resubmission date: 2008-02-15

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

Organisation name of lead contractor for this deliverable: TKK

Revision: 0.95

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

VERSION DETAILS	
Version:	0.95
Date:	15 February 2008
Status:	Final - Post annual review version: addresses comments from reviewers

CONTRIBUTOR(S) to DELIVERABLE	
Partner	**Name**
FI-TKK	Unto K. Laine
FI-TKK	Okko Räsänen
SE-KTH	Gustav Henter
FI-TKK	Toomas Altosaar
UK-USFD	Mark Elshaw (Review)
BE-K.U.Leuven	Kris Demuynck (Review)

DOCUMENT HISTORY			
Version	**Date**	**Responsible**	**Description**
0.1	21.11.2007	Toomas Altosaar	writing
0.2	22.11.2007	Toomas Altosaar	writing
0.3	29.11.2007	Toomas Altosaar	writing
0.45	06.12.2007	Gustav Henter	writing, figures
0.5	07.12.2007	Toomas Altosaar	writing, editing
0.6	12.12.2007	Gustav Henter	references, minor edits
0.7	13.12.2007	Toomas Altosaar	ME's review, figs. from GH & OR
0.8	14.12.2007	Toomas Altosaar	Edited in ME's review comments
0.85	17.12.2007	Toomas Altosaar	Added sw table to 2.1.4 from OR
0.85	20.12.2007	Gustav Henter	figs., KD's review, minor edits
0.9	21.12.2007	Toomas Altosaar	KD's review, Final version
0.95	15.02.2008	Toomas Altosaar	Post annual review: addressed comments from reviewers, new document structure

DELIVERABLE REVIEW			
Version	**Date**	**Reviewed by**	**Conclusion***
0.7_gh		Kris Demuynck	
0.7		Mark Elshaw	

Table of Contents

Deliverable D2.1 - PD and DME Modules

(PD/DME Modules & Applicability of CMM Learning for ACORNS)

1 Introduction

This report covers the deliverables set forth in the Technical Annex (TA) for Work Package 2 (WP2) in the ACORNS project covering the first year period from December 1, 2006 to November 30, 2007.

2 Goals, Tasks, and Deliverables

Two tasks are defined in the TA for WP2 that bind with the rest of the ACORNS project. The first task sets guidelines and goals towards pattern discovery (PD) using discrete model elements (DME). The second task defines a study to investigate the applicability of pattern discovery techniques to discover the essential structures and patterns in speech signals.

3 Task 1: PD/DME Modules

The goal of the first task is to study the emergence of basic patterns in speech and the ways in which these patterns can subsequently be used to represent more complex units. The speech signal is to be considered as a sequence of very short acoustic phenomena that are treated as 'data' without reference to phonetics or linguistics. The application of metrics in an articulatory-acoustic space are suggested since biologically-based systems can be viewed as possessing "field-proven" solutions that may well be worthwhile studying. The goal calls out for implementing a segmentation strategy based on attendant measures of constancy utilising classical linear signal processing as well as scale-invariant methods to form DMEs. In later stages of the project, the results of segmentation are to be channelled towards the identification and labelling of clusters.

The first year D2.1 deliverable for WP2, Task 1, is covered in the following four sub-sections.

3.1 M.Sc. Thesis Publication

WP2 was able to employ Okko Räsänen, a M.Sc. student, in June of 2007 (M7) to take on the task of implementing a bottom-up blind segmentation system. Weekly internal WP2 reports where he described his work and progress culminated in his M.Sc. thesis publication (94 pages) during M12, entitled "*Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture*". This thesis has been made available to ACORNS via the ACORNS Wiki and further dissemination of the work to appropriate journals is currently in progress. The thesis describes in further detail the development of the algorithms as well as the experiments carried out to verify its performance.

The bottom-up blind segmentation system that has been developed can be seen as a stable transcoder for the speech waveform, changing sound pressure level variations into a low variable-rate symbol stream. The symbol rate being emitted from this system corresponds well to the acoustic-phone rate of annotated speech corpora, e.g., 10 or so symbols per second. The

low symbol rate makes the system a very attractive alternative on which to perform higher-level pattern matching. For example, instead of having a fixed-rate symbol rate of 100 symbols per second that is typical for a MFCC-based system, a ten-fold decrease in data rate over which to perform pattern matching can be envisioned. This will definitely have positive implications for computationally constrained problems such as sequence discovery that are typically encountered in the pattern-matching field.

3.2 Description of the WP2 software module

Each technical WP of the ACORNS project is responsible for implementing certain foreseen functionality in the form of a computer program that is integrated by WP5. The software module included in the WP2 deliverable produces blind acoustic-phonetic segmentation of speech, as well as segmental data classification and description by incremental clustering. The module processes one speech signal at a time, producing segment boundary locations, segment boundary probability classifications, and segment category indices for each detected segment in the speech stream. Segments with an equivalent category index are considered as belonging to the same acoustic-phonetic category. Segmentation of small sections of longer signals is possible, and the state of the clustering algorithm can also be saved and loaded in at any time, e.g., for data security and experimental efficiency reasons.

The module is implemented in the MATLAB environment, as directed by ACORNS, and consists of 16 MATLAB m-files including auxiliary functions, two directories for data storage and logging, and readme-documentation. Table 1 gives an overview of the functionality included in each of the 16 files.

Table 1: Files included in the WP2 software module.

adjustspace.m	Reserves more memory for clusters if current size of cluster space is exceeded.
acwp2.m	The main module controlling the processing.
cepstrum.m	Used to estimate segment voicing by cepstral analysis.
cleanspace.m	Removes small clusters from the cluster space.
clusterdata.m	The main clustering algorithm.
convergeclusters.m	Calls for cluster space cleaning operations.
fcluster.m	Formats cluster space.
featdata.m	Prepares data for feature extraction.
grabvector.m	Creates feature vectors for segmental data.
loadwp2.m	Loads saved state of the module.
loadwp2settings.m	Loads module settings.
mergespace.m	Merges nearby clusters together.
removepeak.m	Does peak "masking" in boundary detection.
savewp2.m	Saves current state of the module.
sclusters.m	Lists cluster info to command line.
segdata.m	The main segmentation algorithm.

More extensive descriptions of the underlying solutions and algorithm's performance can be found in the ACORNS publication "*Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture"* (Räsänen, 2007), see section 3.1 of this document. The methodology used is also described in a patent application (FIN-20075696) that was filed in M10 (see section 3.4 of this document).

3.3 WP2 Plan & Strategy

During M10 an abridged report (8 pages) entitled "WP2: Plan, Strategy, Methods, Integration, and Current Status" was distributed to ACORNS. The report describes the reasoning behind WP2's approach. In M12 the full report (14 pages) was made available to ACORNS and describes the bottom-up segmentation algorithm developed by WP2 in technical detail. This report is available on the ACORNS wiki.

3.4 Patent Application

Section 7.1.8 of the ACORNS Technical Annex (pp. 45-46) entitled "Dissemination and Standardisation" deals with knowledge protection, e.g.,

> "In accordance with the general trend to favour knowledge protection, ACORNS will keep an open eye for the possibility to apply for patents on the knowledge and technology to be developed in the project. Junior researchers and PhD students in the project will learn to pay proper attention and procedures for knowledge protection."

A patent application (FIN-20075696) was submitted to the *National Board of Patents and Registration of Finland* on September 2, 2007 and covers the methods described in the bottom-up segmentation system mentioned in section 3.1 of this report. This action is well in line with the knowledge protection goals of the project. Since this patent is still in a preliminary and semi-confidential state, it cannot be published in electronic form as of yet. However, a paper copy of the registered application received back from the patent office has been sent to ACORNS WP0 (project management) at the end of November 2007. The *National Board of Patents and Registration of Finland*'s URL is http://www.prh.fi/en.html

4 Task 2: Applicability of CMM Learning for ACORNS

The goal of the second task over the span of the project is to investigate the applicability of automatic pattern discovery techniques to discover essential structures and patterns in speech signals. Specifically, computational mechanics models (CMM) were chosen since its (initial) theoretical indicators seemed a good fit to WP2 goals and ACORNS in general. Specifically, CMM can represent a much richer class of stochastic processes than traditional HMMs. These representations can be recovered from empirical data with few a priori assumptions through a convergent learning algorithm.

The first year subtask (WP2.T2.1) was to apply a CMM-based paradigm to the problem of discovering patterns in signal representations generated by the two approaches defined in WP1 (distortion measure-based and phone-class specific features, respectively). Since the project first needed to achieve an understanding of CMM, work began with a literature review of this and related methods. Initial tests were conducted on artificial, language-inspired data only. The subtask also stressed the development of CMM theory and the creation of tools to enable incremental learning, i.e., a type of learning in which the trade-off between stochastic complexity of the input data and previously acquired knowledge changes as the learning process proceeds.

The first year D2.1 deliverable for WP2, task 2, is covered by three items: i) an introduction to CMM and related methods, ii) a literature review, and iii) a report on the conducted experiments.

4.1 Introduction to computational mechanics models and related pattern discovery techniques

Computational mechanics models [1] provide a way to represent discrete time stationary stochastic processes as a Markov process over the so called causal states plus random noise. Each causal state is defined as an equivalence class of possible observation histories that all give rise to the same probability distribution for the future process evolution. The description can alternatively be considered as a stochastic automaton, the ε-machine, which generates the observations. This description is much richer than what conventional hidden Markov models can offer. In fact it captures all interesting information about the process, in the sense that it enables optimal prediction of the future — it is a minimal sufficient statistic [1].

For practical applications an approximation of the CMM representation may be reconstructed from empirical observations by means of the so called Causal State Splitting Reconstruction (CSSR) algorithm [2]. Given enough data this reconstruction converges on the correct causal states and transitions between them. Unlike regular hidden Markov models, CSSR can therefore learn the structure of the data generating process. The algorithm has two user-set parameters, only one of which, the memory length L_{Max}, is important for asymptotic convergence. The advantages come at a trade-off in worst-case computational complexity and training data size requirements vis-à-vis regular HMMs. As of yet CSSR may only be applied to discrete symbol sequences.

Observable operator models (OOMs) [3] are another way to characterise stochastic processes. They have similar representative power as CMM, but differ significantly in their internal details: unlike the traditional view of HMMs and stochastic processes as a single stochastic operator applied to a hidden state that determines future behaviour, OOMs have one distinct operator for each observed outcome (symbol). The evolution of the system is then a concatenation of operators, essentially a sequence of actions, rather than a trajectory through state space. Like in the case of CMM, a convergent learning algorithm for OOMs exists.

Intermediate methods, richer than HMMs but less powerful than CMM or OOMs, are also available. Particularly notable are variable-length Markov models (VLMMs), sometimes also called context trees or probabilistic suffix trees. These may be considered ancestors of CSSR since both describe stochastic processes as a tree over recently observed symbols (most recent at the top and then on down), where the terminal nodes describe a probability distribution for the next symbol. The tree branches need not all have the same depth. CSSR differs from VLMMs in that different terminal nodes can belong to the same state, and thus share future distributions. The context tree weighting algorithm (CTW) [4] is an important context tree method which has several advantages over other similar VLMMs.

4.2 Literature overview of computational mechanics models and related methods

Preliminary work was carried out at KTH on a literature survey of the computational mechanics models approach and its applicability to the problems faced by ACORNS in pattern discovery. We now present the main results of this review.

4.2.1 Introduction

ACORNS WP2 revolves around strategies for removing or relaxing some a priori assumptions commonly associated with conventional speech recognition systems. WP2 task 2 in particular centers on pattern discovery in speech data using methods related to computational mechanics. As some of these methods are not widely known within the speech recognition community today, a literature review may be in order. This report contains a selection of texts and resources on the causal state splitting reconstruction algorithm and its applications, as well as related approaches. Some of these topics were discussed at the October 3, 2007 one-day workshop and the ACORNS quarterly meeting on October 4–5 in Stockholm.

4.2.2 Variable-Length Markov Models

An overview of hidden Markov models and some more general techniques is available as part of the book chapter [7]. Three of the methods discussed there are of particular relevance for ACORNS WP2 and will be outlined in this and following sections. First to be considered are *variable-length Markov models* (VLMM), also known as *context trees* or *probabilistic suffix trees*. In contrast to regular hidden Markov models VLMMs can also learn the structure of the generating model. They were introduced by Rissanen [8]. A popular VLMM variant is the *context tree weighting algorithm* (CTW) described in [9]. Context trees and CTW are also reviewed in [10].

4.2.3 Background Reading on Computational Mechanics Models

Even though they are more flexible than HMMs, not every regular language can be represented by a VLMM. Several further generalizations are available. ACORNS WP2 Task 2 takes special interest in *computational mechanics models* (CMM), also known as *causal-state models* (CSM), through which observations from any stationary random process can be considered as a Markov process over the so called *causal states* plus random noise. Information on CMM and causal states is available in [11], as well as in the theses [12] and [13].

4.2.4 The Causal State Splitting Reconstruction Algorithm

Causal state splitting reconstruction (CSSR) is a convergent algorithm to infer causal states from empirical observations of a random process with symbolic output. It shares several procedural similarities with context tree methods. An introduction to the procedure is available in [14]. For comprehensive technical detail and proofs it is better to consult the technical report [15].

GPL-licensed C++ source code for a reference implementation of CSSR is provided on the official webpage of the algorithm [16].[1] Contained within the package is a ReadMe file with some additional information on subtle practical issues.

4.2.5 Applications of CSSR

Both the CSSR webpage [16] and a section on Kristina L. Klinkner's research webpage [17] display a collection of practical applications of the algorithm. Among the known applications only the natural language processing experiments of Muntsa and Lluís Padró are close to the

[1] As of 2007–12–12 the source code was "temporarily off-line" in preparation of "a major update." [16]

topics considered within ACORNS. Their work involves tasks such as named entity recognition and noun phrase detection. The experiments were discussed in the October 2007 CSSR presentation in Stockholm, available on the ACORNS wiki [18]. The slides include figures from Muntsa Padró's thesis proposal [19].

The same line of research has also produced several more recent papers involving CSSR, available from Muntsa Padró's web page [20]. One of these describes a modification of the CSSR algorithm [21]. However, the proposed change does not appear to directly address performance in the presence of noise or other primary interest areas of ACORNS WP2.

4.2.6 Observable Operator Models

An alternative to CSMs with CSSR is provided by *observable operator models* (OOMs). Although quite different from CMMs in their inner workings, they have similar power to represent more general processes than VLMMs. An introduction to OOMs is given in [22]. The paper also presents a convergent algorithm for recovering such models from observations. Cosma R. Shalizi, one of the inventors of CSSR, states in [7] that for many processes that require an infinite number of states in their representations, OOMs are likely easier to reconstruct than CSMs.

4.2.7 Summary of Literature Overview

The references included in this literature overview section cover significant portions of the research concerning CMM and CSSR, as well as some related approaches. It is hoped that this brief literature overview will be of use to the ACORNS project.

4.3 Experiments with CSSR on data sequences from simple language models

As mentioned, CMM representations can be inferred from empirical data using an algorithm known as *Causal State Splitting Reconstruction* (CSSR). To get an impression of its behaviour in practical applications, the algorithm was tested in two batches of experiments involving simple, language-inspired stochastic data sequences. All tests were performed using a reference implementation of CSSR in C++, freely available from the homepage of one of the original inventors of the algorithm at URL http://www.cscs.umich.edu/~crshalizi/CSSR/.[2]

In the first set of experiments, CSSR was applied to a symbolic representation of the recording protocol used in generating the first-year ACORNS Swedish speech data corpus. In this representation, each word was assigned a unique symbol. Word variants such as inflections were given different symbols if their spelling differed. The protocol was then converted to a single data stream by concatenating all 1,000 spoken lines with phrases separated by an additional symbol signifying inter-utterance silence. This yielded a data sequence of 4,295 symbols drawn from a 23-symbol alphabet.

As expected, applying CSSR to this data gave different results depending on the parameters of the algorithm. It was found that, for appropriate values of the two user parameters, the algorithm recovered an almost perfect stochastic automata representation of the reading protocol (figure 1).

[2] As of 2007–12–12 the implementation was "temporarily off-line" in preparation of "a major update."

If run generatively, this process would feed out a string of independent utterances from the data corpus with silence symbols ('-' in the figure) in between. The possible output phrases also include "ta en pappa" and "ta en mamma" (not in the training data), with P=0.0018 each.

If the silence markers were removed a data material of length 3,294, containing 22 distinct symbols, was obtained. For suitable parameter values experiments on this data gave similar, if not quite as good, output as for the previous case (figure 2). Although the resulting automata would be able to generate many of the approximately 100 distinct utterances from the original text without error, they would occasionally also make mistakes. As the symbol '-' provides a rather obvious phrase separation cue, it is not unreasonable that reconstruction would be more difficult without it.

In the course of these experiments, unexpected behaviour was observed in the output automata. Most surprising was that the probabilities in the reconstructed transition matrices did not always sum to one for every state. In figure 1, for instance, an inexplicable difference in total probability sum appears to be the only thing preventing the two end states at the bottom being merged to a single causal state. This and other aberrations appear to be due to one or more bugs in the standard CSSR implementation.

The second round of experiments was performed on a lengthy sequence of one million discrete symbols generated by Louis ten Bosch (RU). This data was given by a simple model of speech as a sequence of randomly occurring words, each word comprising a sequence of 'phones' (symbols). The words were taken from a pre-generated random 'wordlist' of ten short symbol strings, each 4–8 symbols long. Eight different phones were used. To be more realistic a deliberate noise component was introduced occasionally (P=0.05) in the form of phones being substituted with a random phone from the alphabet. Note that the same stochastic process may also be considered as a model of ten different sentences repeated at random. Each symbol then represents a word, similar to the material used in the first experiments.

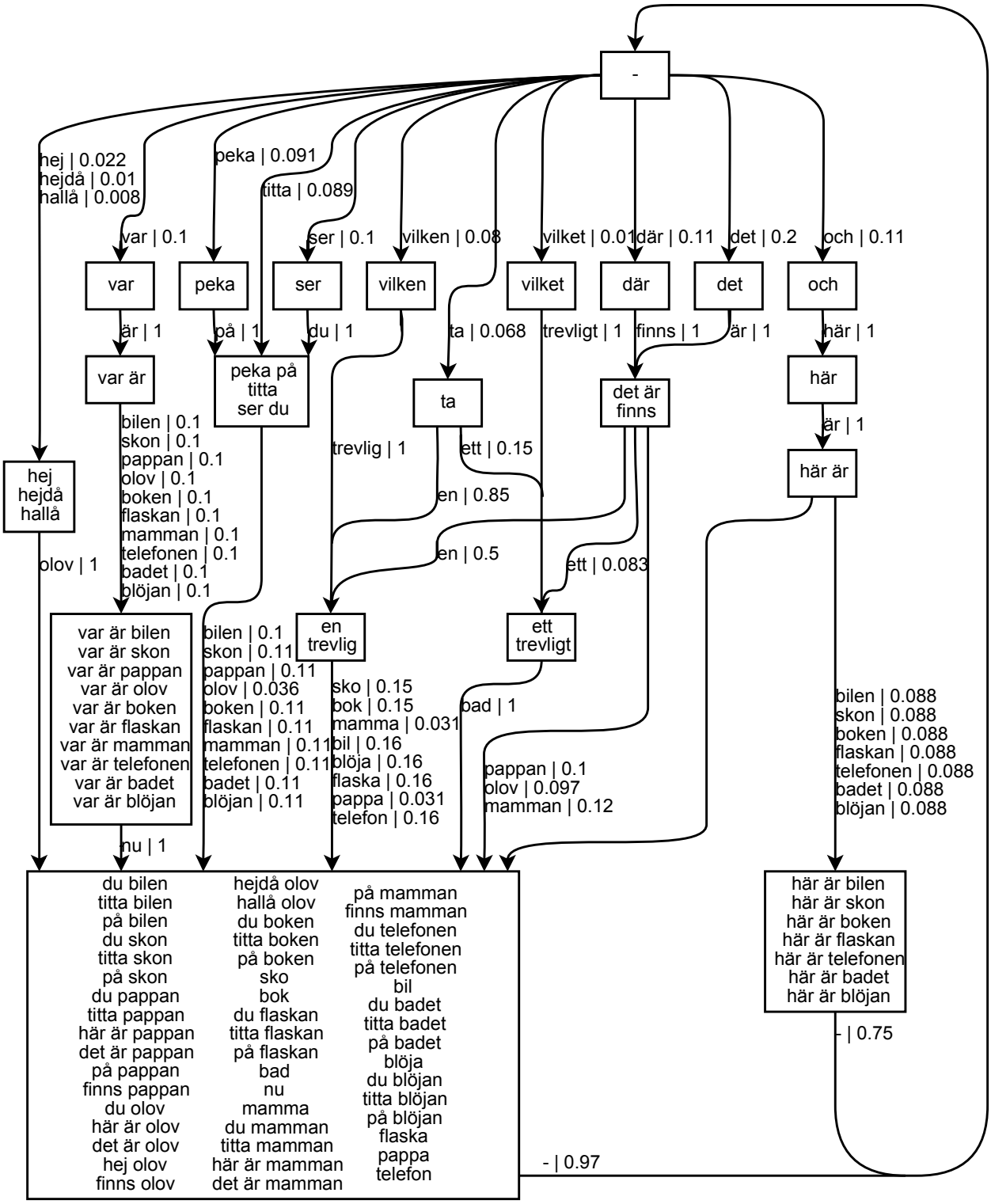

Figure 1. Reconstructed ε-machine with Swedish data including '-' and parameters L_{Max}=4, α=0.002

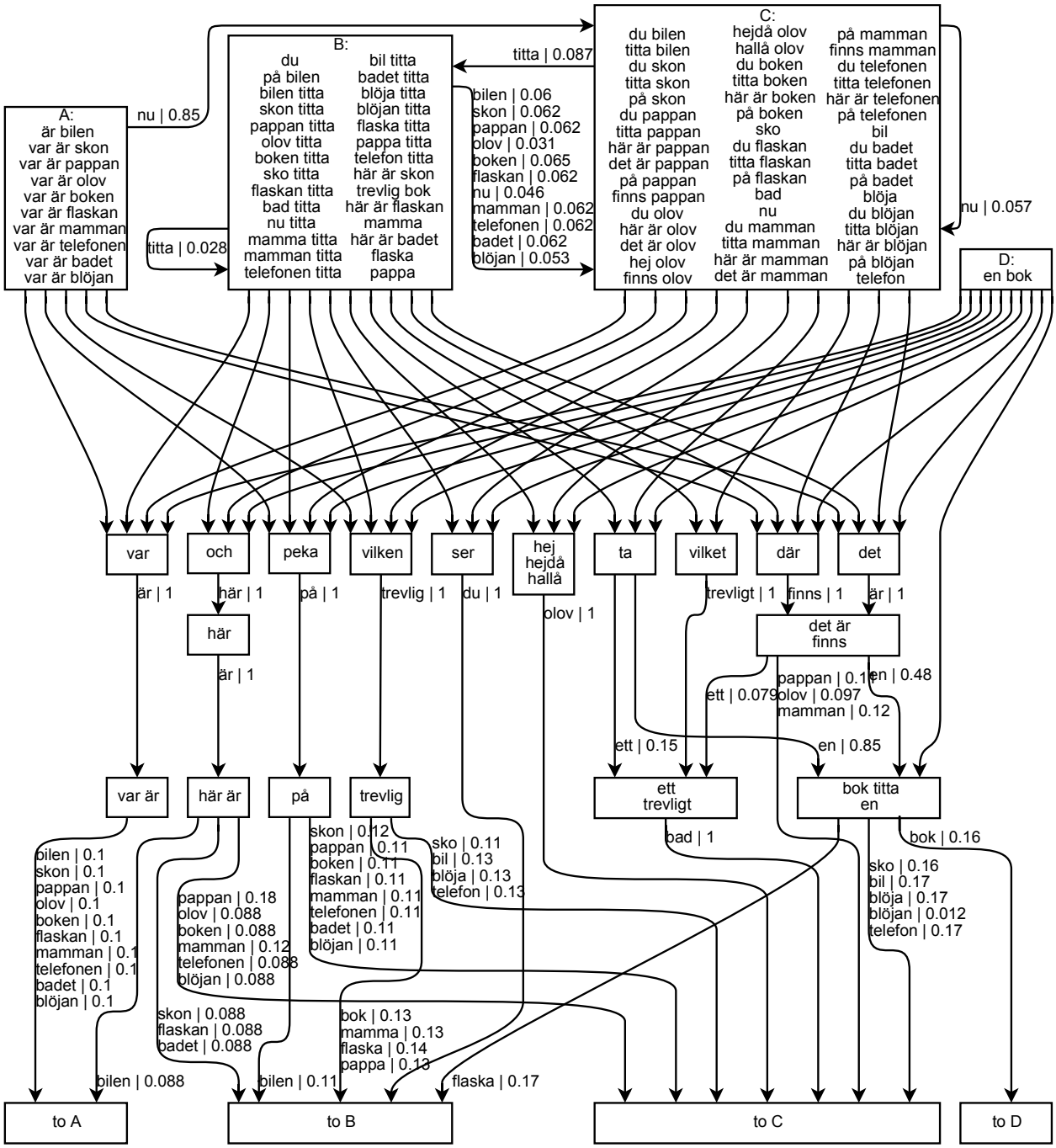

Figure 2. Reconstructed ε-machine with Swedish data, excluding '-', and L_{Max}=3, α=0.01 (some labels omitted)

Results with this data were radically different from those obtained with previous data sequences. Instead of converging on a simple automaton generating the noisy data material, the number of reconstructed states now grew steeply as larger and larger values for the memory length parameter L_{Max} were considered, with no end in sight (figure 3). The computing power requirements also increased prohibitively fast.

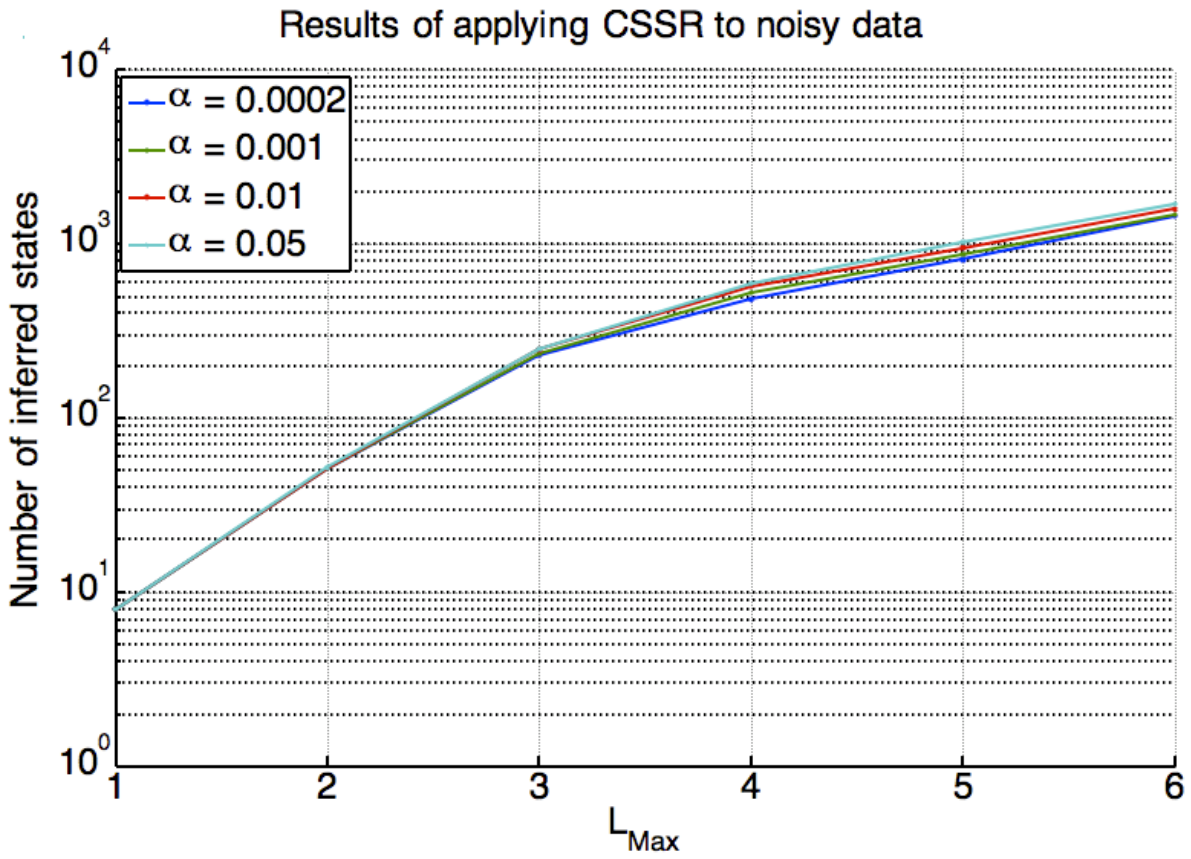

Figure 3. Number of states in ε-machine for noisy data, reconstructed with increased memory length.

Additional experimentation on data generated from similar language models with reduced word lengths, shorter word lists, and smaller alphabet sizes confirmed that the presence of noise was the deciding factor for the unchecked growth to take place. This is consistent with an interpretation that the CMM representation of the noisy process may contain a very large, or even infinite, number of distinct causal states, something that is allowed by the theory [5]. Notably, similar growth has been observed by independent researchers applying CSSR to natural language processing tasks [6].

The current interpretation of the results may be summarised as:

a) Although it works well with clean data, CSSR is very sensitive to noise and cannot in its current form offer any substantial/practical solutions to pattern discovery without modifying parts of the algorithm followed by additional tests.

b) The available CSSR reference implementation used in the tests (written by K. L. Klinkner) contained errors. The extent to which these influenced the results obtained in a) is unclear.

From discussion between the ACORNS partners it was decided that CSSR still holds potential, but that the algorithm should be re-implemented and at the same time debugged to eliminate the uncertainties discussed in b). Furthermore, modifications should also be made to it so that its performance in noise could be better understood and improved.

Currently WP2 is re-implementing the CSSR algorithm of Klinkner/Shalizi in another environment so that modifications and further testing of CMM applicability to pattern discovery in non-ideal noisy conditions can be evaluated more readily. Related discovery algorithms that may offer better resistance to noise or otherwise be more adaptable to the ACORNS framework are also being considered.

5 Conclusion

This report described the work that was carried out by WP2 of ACORNS during Year 1 in accordance to the Technical Annex. The two tasks of WP2 during Year 1, the development of the pattern discovery / discrete model elements (*PD/DME Modules*) as well as computational mechanics models (*CMM*), were described. Knowledge and experience acquired from Year 1 are being used to further direct and refine the goals set out for Year 2 & 3 research.

6 References

[1] C. Shalizi and J. Crutchfield. Computational Mechanics: Pattern and Prediction, Structure and Simplicity, *Journal of Statistical Physics*, 104:816–879, 2001.

[2] C. Shalizi and K. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In: M. Chickering and J. Halpern (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*, pp. 504–511, Arlington, Virginia, 2004. AUAI Press.

[3] Herbert Jaeger. Observable operator models for discrete stochastic time series, *Neural Computation*, 12:1371–1398, 2000.

[4] F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: Basic properties, *IEEE Transactions on Information Theory*, 41:653–664, 1995.

[5] D. Upper. *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. Ph.D. thesis, University of California, Berkeley, 1997. (As of 2007–12–04 available online at http://cse.ucdavis.edu/~dynlearn/papers/TAHMMGHMM.pdf.gz)

[6] M. Padró. *Applying Causal-State Splitting Reconstruction Algorithm to Natural Language Processing Tasks*, Ph.D. thesis proposal, Technical University of Catalonia, Barcelona, Spain, 2005. (As of 2007–12–04 available online at http://www.lsi.upc.edu/~mpadro/publicacions/padro_dea05.pdf)

[7] C. Shalizi. Methods and Techniques of Complex Systems Science: An Overview. In: T. Deisboeck and J.Y. Kresh (eds.), *Complex Systems Science in Biomedicine,* pp. 33–114, New-York: Springer-Verlag, 2006. (As of 2007–12–04 available online at http://arxiv.org/abs/nlin.AO/0307015)

[8] J. Rissanen. A universal data compression system, *IEEE Transactions on Information Theory*, 29:656–664, 1983.

[9] F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: Basic properties, *IEEE Transactions on Information Theory*, 41:653–664, 1995.

[10] M. Kennel and A. Mees. Context-tree modeling of observed symbolic dynamics, *Physical Review E*, 66:056209, 2002.

[11] C. Shalizi and J. Crutchfield. Computational Mechanics: Pattern and Prediction, Structure and Simplicity, *Journal of Statistical Physics*, 104:816–879, 2001.

[12] C. Shalizi. *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. Ph.D. thesis, University of Wisconsin, Madison, 2001. (As of 2007–12–04 available online at http://www.cscs.umich.edu/~crshalizi/thesis/)

[13] D. Upper. *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. Ph.D. thesis, University of California, Berkeley, 1997. (As of 2007–12–04 available online at http://cse.ucdavis.edu/~dynlearn/papers/TAHMMGHMM.pdf.gz)

[14] C. Shalizi and K. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In: M. Chickering and J. Halpern (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*, pp. 504–511, Arlington, Virginia, 2004. AUAI Press.

[15] C. Shalizi, K. Klinkner, and J. Crutchfield. *An Algorithm for Pattern Discovery in Time Series*, Santa Fe Institute Working Paper 02-10-060 (As of 2007–12–03 available online at http://arxiv.org/abs/cs.LG/0210025)

[16] C. Shalizi. *CSSR: An Algorithm for Building Markov Models from Time Series*. Web page, accessed 2007–12–04 at http://www.cscs.umich.edu/~crshalizi/CSSR/.

[17] K. Klinkner. *Research Page --- Kristina Klinkner*. Web page, last accessed 2007–10–23 at http://www.stat.cmu.edu/~klinkner/Research/. (Currently off-line as of 2007–12–05, but accessible through http://www.google.com/search?q=cache:http://www.stat.cmu.edu/~klinkner/Research/)

[18] Website, accessed 2007–12–04 through http://lands.let.ru.nl/wiki.

[19] M. Padró. *Applying Causal-State Splitting Reconstruction Algorithm to Natural Language Processing Tasks*, Ph.D. thesis proposal, Technical University of Catalonia, Barcelona, Spain, 2005. (As of 2007–12–04 available online at http://www.lsi.upc.edu/~mpadro/publicacions/padro_dea05.pdf)

[20] M. Padró. *Publications for Muntsa Padró*, Web page, accessed 2007–12–04 at http://www.lsi.upc.es/~mpadro/research_en.html.

[21] M. Padró and L. Padró. ME-CSSR: an Extension of CSSR using Maximum Entropy Models. In: *Proceedings of Finite State Methods for Natural Language Processing (FSMNLP) 2007*, Potsdam, Germany, 2007.

[22] Herbert Jaeger. Observable operator models for discrete stochastic time series, *Neural Computation*, 12:1371–1398, 2000.