九州大学学術情報リポジトリ
Kyushu University Institutional Repository

EXTENDED EXPERIMENTAL EXPLORATIONS OF THE NECESSITY OF USER GUIDANCE IN CASE-BASED LEARNING

Jantke, Klaus P.
Hokkaido University Graduate School of Enginering

Dötsch, Volker
University of Leipzig Dept. Mathematics & Comp. Sci.

https://hdl.handle.net/2324/3004

出版情報：DOI Technical Report. 135, 1997-03. Department of Informatics, Kyushu University
バージョン：
権利関係：

EXTENDED EXPERIMENTAL EXPLORATIONS OF THE NECESSITY OF USER GUIDANCE IN CASE-BASED LEARNING *

Klaus P. Jantke
Hokkaido University
Graduate School of Enginering
Meme Media Laboratory
Kita–13, Nishi–8, Kita-ku
Sapporo 060
Japan

Volker Dötsch
University of Leipzig
Dept. Mathematics & Comp. Sci.
Institute for Computer Science
P.O.Box 920
04009 Leipzig
Germany

Abstract

This is an extended report focussing on experimental results to explore the necessity of user guidance in case-based knowledge acquisition. It is covering a collection of theoretical investigations as well.

The methodology of our approach is quite simple: We choose a well-understood area which is tailored to case-based knowledge acquisition. Furthermore, we choose a prototypical case-based learning algorithm which is obviously suitable for the problem domain under consideration. Then, we perform a number of knowledge acquisition experiments. They clearly exhibit essential limitations of knowledge acquisition from randomly chosen cases. As a consequence, we develop scenarios of user guidance. Based on these theoretical concepts, we prove a few theoretical results characterizing the power of our approach. Next, we perform a new series of more constrained results which support our theoretical investigations.

The main experiments deal with the difficulties of learning from randomly arranged data in 4 different formal settings. The key insight is that even the right data do not suffice, if they are not arranged appropriately.

The present report aims at presenting a large amount of experimental data exceeding the space available in conference proceedings, usually. We are reporting more than a million of individual learning experiments, each of them comprising several steps of generating hypotheses (2 500 per run[1], in some cases). First results have been presented at the 1996 Pacific Knowledge Acquisition Workshop in Sydney, Australia. A much shorter version of this report will be presented on FLAIRS–97, the Florida AI Research Symposium in Daytona Beach, FL, USA, May 1997.

Contents

*Part of this work has been supported by the German Research Fund (DFG) within the project IND–CBL under contract no. Ja 566/3–3.

[1] In the present form, there are 1 562 440 individual learning runs documented.

1 INTRODUCTION AND SURVEY

Case-based reasoning is deemed an important technology to alleviate the bottleneck of knowledge acquisition in recent computer science (cf. [AP94], [Kol92], [Kol93], and [RS89]). In case-based reasoning, knowledge is represented in the form of particular cases with an appropriate similarity measure rather than any generalized form. Those cases are collected during knowledge processing. For solving particular new problems, cases representing former experience are retrieved. The most similar cases are chosen as a basis for generating new solutions including techniques of case adaptation. There is a widely accepted common understanding of case-based reasoning which is based on a methodological cycle consisting of the main activities *retrieve*, *reuse*, *revise*, and *retain* (cf. [AP94]). Here, there is no need to go into further details.

Within case-based reasoning, case-based learning as investigated in [Aha91] and [AKA91] is a natural way of designing learning procedures. There are even normal form results (cf. [Jan92] and [GJLS97]) explaining that all learning procedures of a certain type may be rewritten as case-based learning procedures. The first task of case-based learning is to collect good cases which will be stored in the case base for describing knowledge and classifying unknown examples. Case-based learning algorithms do not construct explicit generalizations from examples which most other supervised learning algorithms derive. Their hypotheses consist of case bases together with similarity concepts. Both constituents may be subject to learning, i.e. the second task of case-based learning might consist in suitably tuning the similarity measure in use. Both collecting cases and tuning similarity measures is subject of the present investigation.

The specific goal of our research work reported here is to gain a better understanding of the power and limitations of case-based learning where stabilization of the acquired knowledge is essential (cf. [Gol67], [AS83], and [Jan89], e.g., for discussions of the stabilization phenomenon in learning). To allow for precise results which are easy to communicate, we have chosen the problem domain of learning formal languages. There is already a collection of topical results recently published (cf. [JL93], [SJL94], [JL95], and [GJLS97]).

The present paper reports about some comprehensive endeavour comprising a variety of experimentations intended to explore the feasibility of the following fundamental case-based reasoning approach: **Given any CBR system, apply it. Whenever it works sucessfully, do not change it. Whenever it fails on some input case, add this experience to the case base. Don't change anything else.**

This scenario will be discussed in some more detail in chapter 3.2 of this report.

The present investigation is an extended version of [JD97b]. An excerpt will be presented and published as [JD97a].

The present report is extended in a threefold way. (1) The series of experiments reported in chapter 5.1 are extended enormously. (2) There is a completely new setting of experiments reported in chapter 5.5. (3) The new experiments exhibit a couple of further insights in the peculiarities of case-based learning. These insights are summarized in chapter 6.

The approach presented in [JD97b] as well as the present one are exceeding our former publications [DJ96a] and [DJ96b] in two respects. First, we have adopted a much more general perspective which illuminates the relevance of our results to a wide range of logically based approaches. This is briefly described in chapter 2. Second, we have extended the experiments reported in [DJ96a] and [DJ96b] to demonstrate that the key phenomena identified are not sensitive to several changes of the experimental setting.

Towards a better understanding of the power and limitations of case-based learning, we are addressing typical questions like the following:

- When learning by collecting cases, how much does the success or failure of learning depend on the information provided to the learning mechanisms?
- What are the particular difficulties which may prevent some case-based learner from reaching its goal?
- Which role play particular tactics of arranging cases during learning? How robust is case-based learning to slightly changing weights of cases in the case base?

Our subsequent answers to those questions exhibit the importance of user guidance impressively.

There are a lot of more specific questions. Here, we are illustrating only a few of them.

- If one knows already which cases are crucial for learning sucessfully, what about the importance of presenting this information in the right order?
- Is there any hope to compensate for some careless ordering by a sufficiently high redundancy, i.e. by repeating essential cases sufficiently many times?
- Is there any known relationship between structural properties of the target concept to be learnt and the tradeoff relating ordering problems and redundancy?

As a side effect, the investigation may lead us to a better understanding of the importance of so-called good examples in inductive learning. Learning from good examples was introduced by Rusiņš Freivalds, Efim Kinber, and Rolf Wiehagen (cf. [FKW89] and [FKW93]). Further recent publications are [LNW94] and [FKW95], e.g.

Last but not least, we are addressing a quite fundamental issue of artificial intelligence: the embedding of particular automated reasoning procedures into more comprehensive scenarios of knowledge processing. It seems one of the key insights of our findings presented below that case-based reasoning procedures do essentially depend on an appropriate embedding into ensembles of reasoning mechanisms. As stand-alone devices, they will rarely work. This has some immediate implications, as the embedding scenario will usually determine certain constraints to be taken into account.

2 STRUCTURAL SIMILARITY AND PARTIAL ORDERINGS

The following insight lead to our quite fundamental approach towards advanced similarity concepts to be presented in [MJ97]. Like wide areas of computer science, in general, traditional CBR is suffering from the phenomenon of *levelling down*. Although computer applications mostly deal with highly structured objects, their inherent structure is usually levelled down during knowledge acquisition and representation, for fitting into the binary world of computing machinery. Consequently, it is usually extremely difficult to develop and implement automated reasoning procedures on those flat knowledge representations which exploit the structured information of the original objects as efficiently as possible.

In the application domain (cf. [FC93], for a general description), which is belongs to the exciting area of industrial building design, objects are highly structured and may be reasonably understood as graphs, e.g. Representative objects under consideration are fresh air supply networks or water supply pipes, for instance, as illustrated by the following figure.

In many application areas, structured formal concepts like graphs, terms, frames, or patterns, e.g., are more appropriate to represent real objects than lists of attribute/value pairs. In many cases, logical knowledge representation formalisms provide a well-structured background. Frequently used first order formulae, like Horn clauses, e.g., have some natural internal structure somehow related to the semantics they are carrying. This bears evidence for the need of related structural similarity concepts. [Jan94], [DOC+93], and [BJST93] have set the stage for those investigations. [MJ97] develops a first axiomatic approach towards the characterization of fundamental properties of structural similarity concepts.

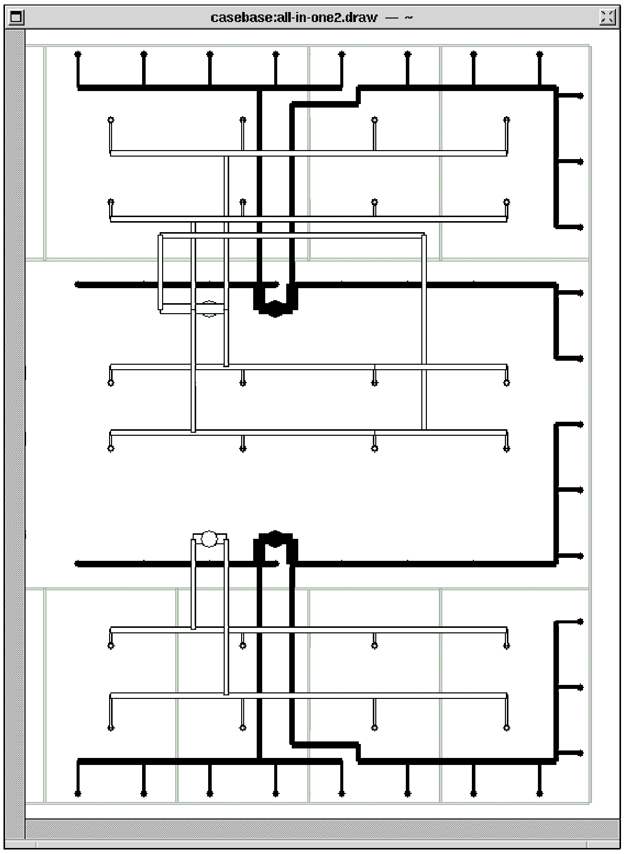

Figure 1: Some Case-Based Design from the FABEL System intended to Illustrate the Need for Structural Reasoning

Recently, [OB96] presented some non-numerical treatment of similarity in which the system's response to some case input is not a most similar case, but a partial ordering of certain cases.

We refrain from a discussion of further details and confine ourselves to the following short summary: In certain application domains and for avoiding several difficulties which mainly result from the loss of structural information in flat knowledge representations, structural similarity concepts based on some partial ordering of cases turn out to be very useful. In many domains, finding some appropriate concept of case similarity essentially means determining some corresponding partial ordering of cases.

Thus, in its right perspective, learning similarity concepts might be understood learning of corresponding partial orderings. This is the focus of our present investigation.

At a first glance, this seems to be a quite abstract construction. But a closer look reveals that this idea is not new at all. In every standard Prolog program, the predicates occurring as heads of clauses are even totally ordered. The reader may adopt this as an illustration, for a moment.

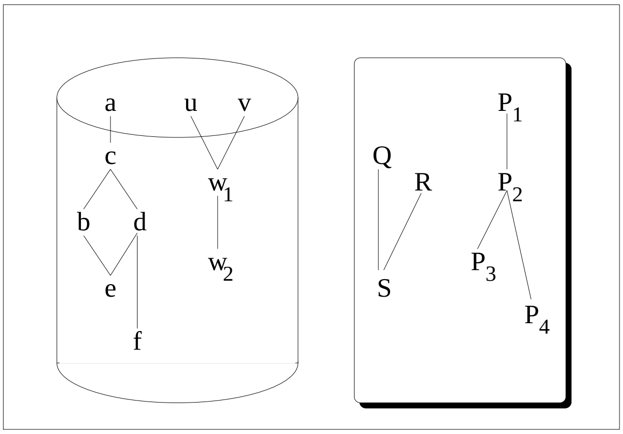

Figure 2: Some Logical Case Memory System

Partially ordered predicates may be taken to represent certain views at a case base with some preference. For a conceptually quite interesting approach to formalize and to process several views within the traditional attribute/value based CBR approach, the interested reader should consult [Sch96].

A system's behaviour, i.e. its semantics, can be specified in several ways (cf. [Jan97]). For the purpose of the present paper, we focus on a very simple approach and refrain from discussing the overwhelming amount of alternatives.

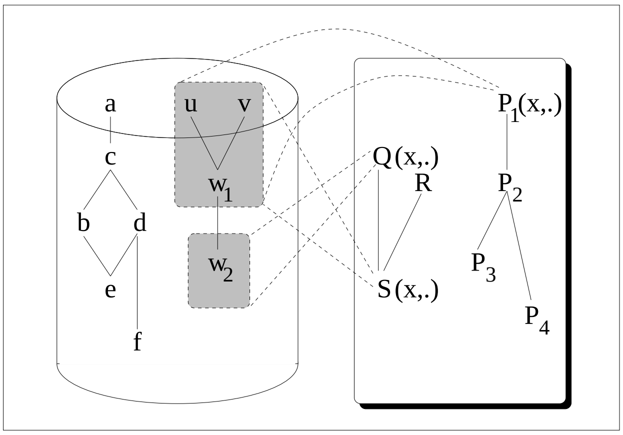

Figure 3: Some Predicate's View Determining a Case Memory System's Semantics

Several knowledge representation formalisms might be reasonably understood as partially ordered units of a certain type. Prolog programs, for instance, are collections of Horn clauses which are partially ordered. Changing this partial ordering is known to be crucial for the overall system behaviour.

We adopt the concept of a *logical case memory system* (cf. [Jan97]). One might imagine a collection of partially ordered predicates as shown in figure 2 above. Every predicate is assumed to be a binary one. Descending lines lead from predicates which are higher ranking w.r.t. the underlying partial ordering $\sqsubseteq$ to those of a lower rank.

Cases are terms. Consequently, a case base is a set of terms which admits a natural partial ordering: subsumption. Thus, approaches like in [BW96] are easily generalized.

The answer to some query x, i.e. to some term, should be any case y such that the highest ranking predicate when applied to these arguments becomes valid (this is just one approach from [Jan97]), i.e.

$$P(x,y) \quad \wedge \quad \exists y'\, Q(x,y') \implies Q \sqsubseteq P \tag{1}$$

The returned case y is understood as a most similar one w.r.t. the query x where the particular predicate P with $P(x,y)$ provides the reason for this choice.

The sample problems discussed in [OB96] might be easily viewed under this perspective.

There are several refinements of this basic idea (cf. [Jan97]) far beyond the scope of the present paper. We focus on the problem of learning the underlying partial ordering. For this purpose, we restrict the type of predicates drastically. Nevertheless, it will turn out that learning remains an extraordinarily difficult problem which seems almost unsolvable without substantial user guidance.

In the remaining part of this chapter, we narrow the problem space under investigation suitably.

Requirement (1) above is somehow of a higher order, as it contains a variable predicate Q. The overall approach becomes conceptually much simpler if one may assume some universal predicate P^* which allows to circumscribe all the other predicates involved via some additional argument.

$$P(u,x,y) \;\wedge\; \forall v,y'\, P(v,x,y') \;\Rightarrow\; P(v,.,.) \sqsubseteq P(u,.,.) \tag{2}$$

As the partial ordering of those predicates is obviously determined by the corresponding indices, this leads to a further simplification:

$$P(u,x,y) \;\wedge\; \forall v,y'\, P(v,x,y') \;\Rightarrow\; v \sqsubseteq u \tag{3}$$

We adopt this simplified setting in the sequel. The particular predicate P is true for three arguments u, x, and y if one of the following two cases holds:
(1) u is a substring of x and $y = 1$ or, alternatively,
(2) u is not a substring of x and $y = 0$.

3 CASE-BASED KNOWLEDGE ACQUISITION SCENARIOS

In its right perspective, the present paper deals with the difficulties of acquiring the knowledge forming logical case memory systems.

More specifically, we have chosen a very specific type of logical case memory systems to focus on. These systems are characterized by a remarkable syntactical simplicity as well as by a considerably simple semantics. They seem particularly suitable for case-based reasoning. Nevertheless, our investigations will exhibit that unsupervised learning will not succeed, usually. These results to be presented in the sequel throw some light at the essential difficulties of learning logical case memory systems, in general.

We might suppress technicalities as much as possible. The key concepts are quite simple.

3.1 The Application Domain

We investigate the problem of learning formal languages in a case-based manner. The reader may interpret *learning* as a particularly ambitious task of *knowledge acquisition.*

A minimal collection of necessary formalisms will be introduced almost informally (cf. [GJLS97], for a detailled discussion of almost all the technicalities we need, and [DJ96a], for a similar, but purely learning-theoretic investigation). [Gol67] is the seminal paper underlying our learning paradigm invoked. From the large number of introductory and survey papers, the reader is directed to [AS83] or [Jan89], e.g. Here, we intend to introduce and clarify the basic concepts in an informal, but precise way.

The target class of formal languages to be learnt is specified via some concept of acceptors: *containment decision lists.* (These are our specific logical case memory systems focussed on throughout the rest of the paper.) The learning theoretic investigation in [SS92] has drawn our attention to this quite simple type of decision lists. Informally speaking, a containment decision list (CDL, for short) is a finite sequence of labelled words (w_i, d_i) $(i = 1, \ldots, n)$, where the labels d_i in use are either 0 or 1. Such a list can be easily understood as an acceptor for words as follows. Any word w fed into a CDL is checked at node (w_1, d_1) first. If any check tells us that w_i is a subword of w, this word is classified as determined by d_i, i.e. w is accepted exactly if $d_i = 1$. If otherwise w does not contain w_i, the input word w is passed to w_{i+1}. All words passing through a containment decision list without being classified at any node (w_i, d_i) are classified complementary to the last node, i.e. they are accepted, if $d_n = 0$, and they are rejected, otherwise.

$$T \quad = \quad [\,(aab, 1)\,,\ (aa, 0)\,,\ (a, 1)\,,\ (b, 1)\,] \qquad (4)$$

is an illustrative example. Roughly speaking, the language accepted by T contains all words containing aab or not containing a square of a. Words in the complement are containing aa, but not containing aab. Containment of words is denoted by the binary relation symbol $\preceq$.

In terms of logical case memory systems, we are faced to the specific case of 5 predicates which can be uniformly generated from two related universal predicate1 P_1^* and P_0^* defined by

$$P_1^*(u, x, y) \iff u \preceq x \ \wedge \ y = 1 \qquad (5)$$

$$P_0^*(u, x, y) \iff u \preceq x \ \wedge \ y = 0 \qquad (6)$$

The particular predicates encoded in the sample CDL T above are named Q_1, Q_2, Q_3, Q_4, and Q_5 defined by $Q_1 = P_1^*(aab, ., .)$, $Q_2 = P_0^*(aa, ., .)$, $Q_3 = P_1^*(a, ., .)$, $Q_4 = P_1^*(b, ., .)$, and $Q_5 = P_0^*(b, ., .)$, respectively. The underlying ordering is $Q_1 \sqsupseteq Q_2 \sqsupseteq Q_3 \sqsupseteq Q_4 \sqsupseteq Q_5$, obviously.

We omit the reduction of these two predicates P_1^* and P_0^* to a single one. Moreover, we mostly refrain from further references to the underlying general concept of logical case memory systems. Another example, which will be used for the first experimental exploration below, is depicted here:

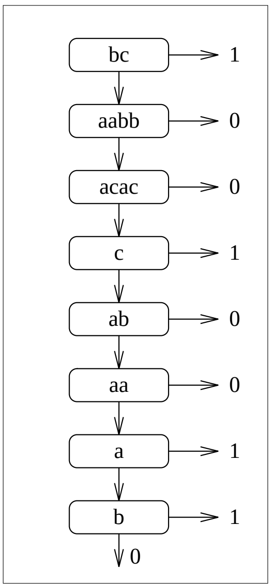

Figure 4: The CDL T^*

For illustration, assume that the word $w = acca$ is fed into T^*. As bc is not contained in w (formally expressed as $bc \not\preceq w$), the word w is passing the first node. The same applies to the nodes labelled by $aabb$ and by $acac$, respectively. At the fourth node, it holds $c \preceq w$. Therefore, w is classified at this node: It is accepted.

This example CDL named T^* will be used below for our four series of experiments. Furthermore, we will take this sample to exemplify a few of our theoretical concepts.

Due to [SS92], arbitrary containment decision lists are known to be learnable. In other words, the knowledge contained in any CDL T can potentially be acquired by processing finitely many cases describing the target language accepted by T.

We will show that this *theoretical result* is *practically valid* only in the presence of *substantial user guidance*.

3.2 The Application Scenarios

There are several ways to present information about formal languages to be learnt. The basic approaches are defined via the concept *text* and *informant*, respectively. A text is just any sequence of words exhausting the target language. An informant is any sequence of words labelled alternatively either by 1 or 0 such that all the words labelled by 1 form a text whereas the remaining words labelled by 0 form a text of the complement of the target language.

When languages are learnt, learning devices have to express their guesses in some particular form. Case-based learners, naturally, generate bases of selected cases and tune similarity concepts (cf. [JL95] and [GJLS97]). There have been published a small number of case-based learning algorithms (cf. [Aha91] and [AKA91]) reflecting the standard case-based reasoning paradigm. An experimental investigation of these algorithms and a comparison to other inductive learning algorithms (cf. [BDF96]) in the setting of formal language learning exhibited a number of difficulties in case-based learning. The present study is an immediate reaction to those phenomena.

In this paper, in its right perspective, we do not intend to analyze, to evaluate, and to criticise some particular algorithm, but some general *paradigmatic idea*. However, when any idea is implemented to become subject not only to theoretical investigations, but also to experimental exploration, it's getting the form of some specific algorithm – at least in computing. Every implementation is concrete. This is an unavoidable dilemma[2]. Consequently, what is tried, what is explored, and what is finally criticised is not the idea itself, but some more operational version. There might be always the argument that the deeper reason for identifying some weakness or even some flaw does not stem from the idea itself, but from implementational details. There is no way out. One can only try to be as careful as possible with any decision about fixing details. That's what we do below.

The paradigmatic idea of case-based learning under investigation can be very briefly expressed as follows: *Given any CBR system, apply it. Whenever it works sucessfully, do not change it. Whenever it fails on some input case, add this experience to the case base. Don't do anything else.*

[2]This even applies to social resp. political ideas. However, we refrain from an in-depth discussion of this issue which is highly interesting, as well.

The simplicity of CBR ideas is charming and has attracted many people, from theory to applications. We suspect it might be sometimes misleading.

In [AKA91] there has been presented some simple algorithm named IB2 for acquiring knowledge like CDLs from finitely many cases. IB2 is selectively collecting cases which are subsequently presented, in case there is any need to do so. It is exactly following the paradigmatic idea circumscribed above.

For our purpose, we extend IB2 to allow for an adaptation of similarity concepts. This is inevitable, as certain case-based knowledge representations do possess some internal structure in contrast to flat case bases which might be understood as sets, only.

Before going into details, we need some similarity measure:

$$\sigma(v,w) = \begin{cases} weight(v) & : \text{ if } v \preceq w \\ 0 & : \text{ else} \end{cases} \tag{7}$$

It is assumed that cases collected in some case base get assigned their individual weight. The reader may imagine that every weight is initially set to 1.

In essence, this is the particular technological version of learning a similarity measure by learning a partial ordering. The cases of the case base are used as indices to the underlying universal predicate. *Thus, collecting those cases means learning predicates in this particular setting. Learning weights means learning the partial ordering among predicates.*

Knowledge acquisition from subsequently presented cases by IB2$^\sigma$ proceeds as follows. Assume any given case base. Whenever a new case is presented and correctly classified by this case base, i.e. its nearest neighbour in the case base carries the same classification value, then nothing is changed. In the opposite situation, there must be some case in the present case base being responsible for the misclassification. The weight of this particular case is reduced from $1/k$ to $1/(k+1)$ and the misclassified case is put into the case base. This is a slight adaptation of IB2.

We have performed 66 800 knowledge acquisition experiments reported in chapter 5.1 below. They exhibit a catastrophic behavior of IB2$^\sigma$.

It turns out that algorithms like IB2 and IB2$^\sigma$ do essentially depend on user guidance. Corresponding formal concepts are sketched in chapter 4 which follows. Chapter 5 reports about more than 1 000 000 particular experiments based on these theoretical concepts.

To say it clearly: Every individual experiment is an attempt to learn the particular CDL from a sequence of correctly classified cases. In certain experiments, a single run means to feed in 2 500 cases. Details will follow.

4 THEORETICAL RESULTS

We have developed some algorithmic principles to generate appropriate cases for presenting CDLs to knowledge acquisition procedures like IB2$^\sigma$. The key concepts are called *sets of good examples*, *lists of good examples*, and *optimzed lists of good examples*, respectively. Instead of a complete formal treatment, we confine ourselves to "a case-based presentation", i.e. we exemplify these concepts by the sample CDL T^* from above. For the basic concepts mentioned, the corresponding notations are $SEX(T^*)$, $LEX(T^*)$, and ${}^{opt}LEX(T^*)$, respectively.

$SEX(T^*)$ =
{ $(a,1)$, $(aa,0)$, $(aabb,0)$, $(ab,0)$, $(acac,0)$, $(acacaabbc,1)$, $(b,1)$, $(bc,1)$, $(c,1)$, $(caab,1)$, $(caabb,0)$ }

$LEX(T^*)$ =
a list of 319 Elements resulting from repetitions of
($(acacaabbc,1)$, $(bc,1)$, $(caabb,0)$, $(aabb,0)$, $(acac,0)$, $(caab,1)$, $(c,1)$, $(aa,0)$, $(ab,0)$, $(a,1)$, $(b,1)$)
which is a particular ordering of $SEX(T^*)$.

${}^{opt}LEX(T^*)$ =
($(acacaabbc,1)$, $(bc,1)$, $(caabb,0)$, $(aabb,0)$ $(acac,0)$, $(caab,1)$, $(c,1)$, $(aa,0)$, $(ab,0)$, $(a,1)$, $(b,1)$, $(acacaabbc,1)$, $(caab,1)$, $(aa,0)$, $(acacaabbc,1)$, $(acac,0)$, $(caab,0)$, $(ab,0)$, $(acacaabbc,1)$, $(caabb,0)$, $(caab,1)$, $(aa,0)$, $(caab,1)$, $(ab,0)$)

Roughly speaking, these sets resp. lists can be effectively generated for any given CDL. Based on information of this type, case-based knowledge acquisition works quite impressively as expressed in the sequel.

It is worth to consult the research work on so-called "good examples" in inductive learning theory (cf. [FKW89], [FKW93], and [FKW95], [LNW94] e.g.). In [Jan97] underlying our present paper, there has been pointed to the same subject from the prespective of some learning scenario.

Theorem 1 [Key Properties of IB2$^\sigma$]

(1) For arbitrary containment decision lists, IB2$^\sigma$ works conservatively, i.e. it is changing its hypotheses only if the current case presented contradicts the current hypothesis.

(2) For arbitrary containment decision lists, IB2$^\sigma$ works semantically finite, i.e. in learning a particular target language it never changes a hypothesis which is completely correct.

(3) For arbitrary containment decision lists, IB2$^\sigma$ does not work consistently, i.e. there are intermediate hypotheses which do not correctly reflect the information from which they have been generated.

Although the first one is a very simple result, it is of some methodological value. First, it is characterizing IB2$^\sigma$ with some clarity not found before. Second, it raises the question for similar characterizations of other algorithms in this area.

Theorem 2

For arbitrary lists $LEX(T)$ and ${}^{opt}LEX(T)$, the algorithm IB2$^\sigma$ is acquiring a case base with weights assigned to each case which equivalently represents the target T.

Our Theorem 2 above is exhibiting that case-based knowledge acquisition may work quite successfully, provided some user is able to provide the necessary guidance by (i) choosing the appropriate information (formally: $SEX(T)$) and by (ii) ordering it suitably (formally: $LEX(T)$ or, even better, ${}^{opt}LEX(T)$).

The following experiments are exhibiting that there is no hope for success without user guidance.

5 EXPERIMENTAL RESULTS

Our experiments have been performed using the system TIC which is not described here in any detail (cf. [BDF96], for a comprehensive description). We have run more than 100 000 experiments of learning the sample list T^*. These results are surveyed first.

The following documentation of our experimental explorations is supported by figures of three types. There are statistical data like in figure 6, e.g., intended to illustrate the development of the ratio of success during some learning process consisting of a sequence of steps. In many cases, this is also illustrating that learning fails, at least within the period of time documented. Another type of figures like figure 5, e.g., is displayng the main interface of the system during experimentation. When such a screen dump is documented, this is usually done to present some collection of related data. A third type of figures like figure 7, e.g., is documenting a particular hypothesis generated during learning.

5.1 First Experiments

In the setting of our first four series of experiments, in every run 2 500 randomly chosen cases are subsequently fed into IB2$^\sigma$. After every 100 inputs, the intermediate hypothesis is documented. Thus, every run is documented via a sequence of 25 hypotheses. Statistics as displayed in figure 6 below refer to these hypotheses.

Figure 5 is illustrating the system state after one experimental run.

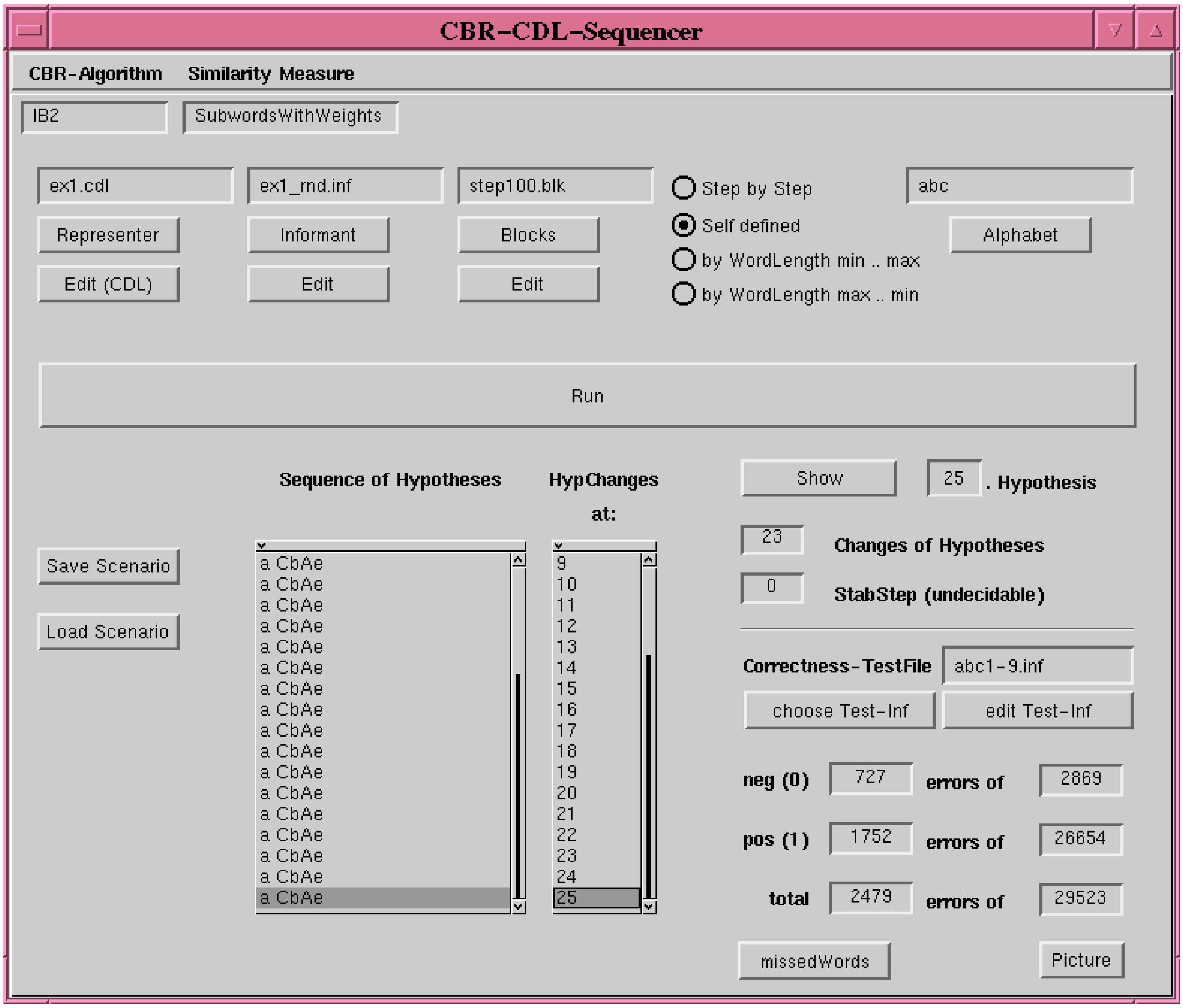

Figure 5: (Not) Learning from 2 500 Cases

The overall error rate of the final hypothesis is 8.40%. The development of errors during knowledge acquisition is displayed by figure 6.

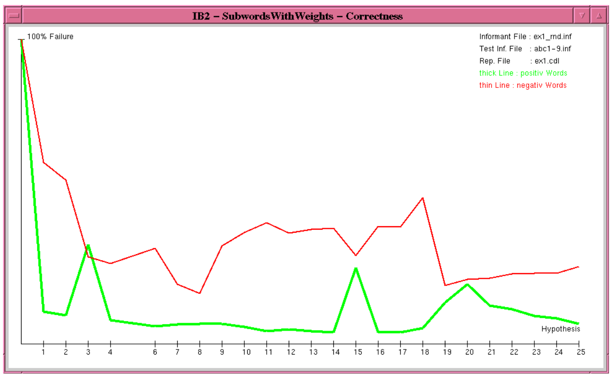

Figure 6: The Ratio of Incorrectly Classified Cases

In figure 5 it might be a little confusing that every hypothesis is mentioned under the same name. This is due to the fact that there is a unique Smalltalk object with this particular name. Nevertheless, there is access to every individual hypothesis and to all relevant data.

In some special display, there are the steps listed at which changes of hypotheses occurred.

In the present window, the 25th hypothesis has been chosen for inspection. Note that in this series of experiments, hypotheses are only documented after every 100 cases. Thus, the 25th hypothesis is based on 2 500 individual cases. It is of an enormous size compared to the target CDL T^* which has only 8 nodes. Its has 134 weighted cases and is (partially) displayed in figure 5.

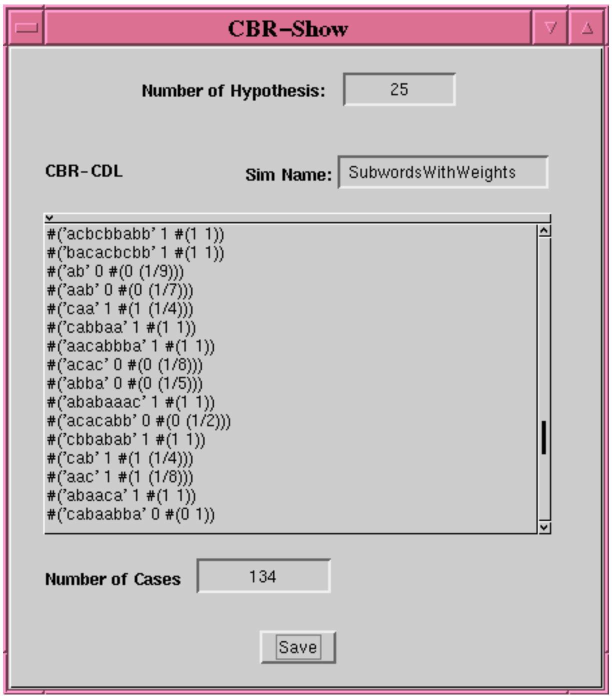

Figure 7: Hypothesis after Processing 2 500 Cases

We conclude this subsection by a survey of 4 series of experiments (different from the survey in [JD97b]).

Series 1		Series 2	
Number of Cases Maximal Length Experiments	2 500 9 5 000	Number of Cases Maximal Length Experiments	2 500 9 6 000
Learning Results			
Success Failure	0 5 000	Success Failure	0 6 000
Size of the Final Hypothesis			
maximal minimal average	247 32 141	maximal minimal average	247 32 141

Series 3		Series 4	
Number of Cases Maximal Length Experiments	2 500 9 7 000	Number of Cases Maximal Length Experiments	2 500 9 48 800
Learning Results			
Success Failure	0 7 000	Success Failure	0 48 800
Size of the Final Hypothesis			
maximal minimal average	247 30 141	maximal minimal average	260 21 141

Very roughly speaking, knowledge acquisition from randomly presented cases did not work in any single attempt out of all together 66 800 experiments.

5.2 Constrained Experiments

In response to the negative results reported in the subsection before, we developed the theoretical concepts introduced in chapter 4.

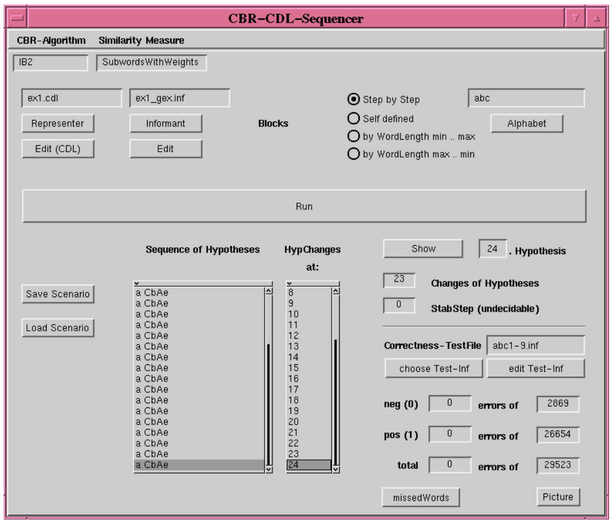

Figure 8: Learning from 24 “Good Cases”

Figure 8 is illustrating the success of learning T^* from ${}^{opt}LEX(T^*)$ according to Theorem 2. The corresponding hypothesis consisting of 11 cases is depicted in figure 9.

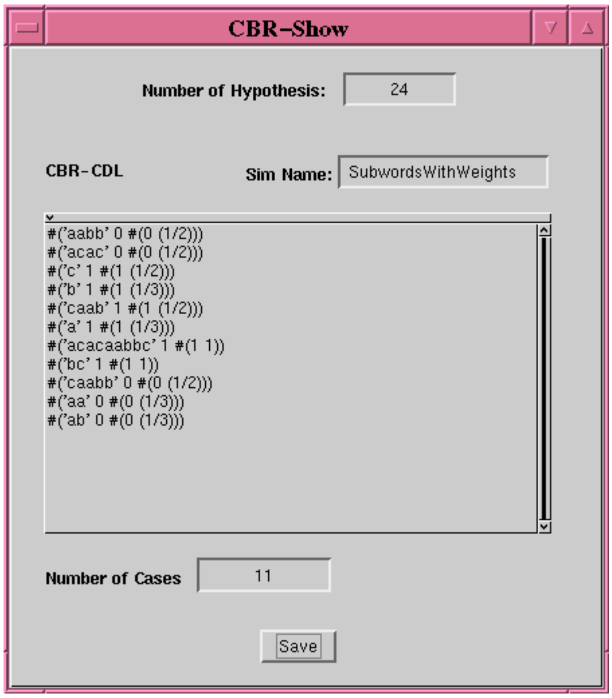

Figure 9: The Result of Learning T^* from Cases

After this result only reflecting the theoretical insights of chapter 4, we asked again for the importance of user guidance. What about randomly rearranging the good cases of $SEX(T^*)$ such that their presentation may differ from ${}^{opt}LEX(T^*)$?

We performed 50 000 experiments with random permutations of ${}^{opt}LEX(T^*)$. The result is impressive: IB2$^\sigma$ learned in only 41 experiments and failed in 49 959, i.e. the rate of success without user guidance, even in the presence of only carefully chosen cases, is only 0.082%.

Number of Permutations	Sucesses	Failures	Rate of Success
50 000	41	49 959	0.082 %

The final figure of chapter 5.2 is showing the rates of misclassifications of positive and negative cases, respectively, during one run of IB2$^\sigma$ on a particular permutation of ${}^{opt}LEX(T^*)$. This is just one sample out of the total amount of 50 000 experiments. It is plain to see how the algorithm is "changing its mind" when being faced to less carefully presented examples, although all these cases come from the collection $SEX(T^*)$ on which IB2$^\sigma$ might learn successfully.

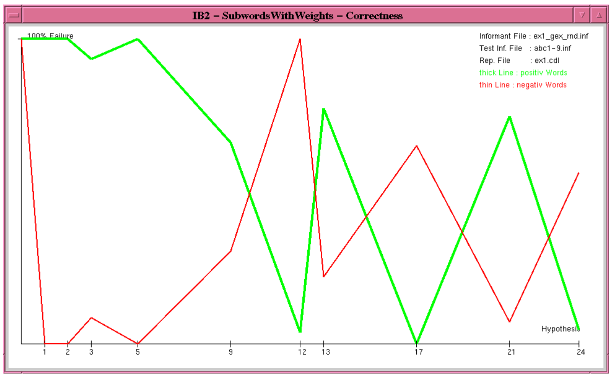

Figure 10: The Ratio of Misclassifications on Randomly Arranged Good Examples

Recall that the particular CDL T^* has been chosen only for illustration. We tried several other CDLs of about the same size and could not find remarkably different experimental results. Thus, it seems not worth to report about in detail. For a more comprehensive treatment, we decided to try two further types of experiments which are considerably different. Before two subsequent chapters will deal with these experiments in detail, we will give some motivation and some overview.

Some reader might argue that the CDL T^* is much to small and quite unstructured to reveal relevant phenomena occurring in pratically interesting settings of case-based learning. Some other readers might want to see more complete experimentations checking all potential stimulus/response pairs up to a certain size. Due to obvious combinatorial reasons, the one desire is ruling out the other. Consequently, we decided to undertake two complementary series of exploratory investigations.

In a first setting, we have chosen a randomly constructed CDL of 70 nodes. Learning and memorizing formal objects of this size is usually far beyond the capabilities of human beings. As decision trees and decision lists of that size might easily occur in practice, the need for automated reasoning, in general, and computer-supported knowledge acquisition, in particular, is obvious. Thus, it is truly relevant to explore the limitations of learning objects of such a complexity in a case-based manner automatically. Chapter 5.3 is presenting our findings.

In a somehow complementary setting, we constructed all – literally all – CDLs up to a certain size, classified them by structural properties, and performed the same experiments with all of them. This will be reported in chapter 5.4 below.

5.3 Exploring Complex Target CDLs

The CDL under investigation throughout the present chapter is named T_{7401} (this notation refers to some indexing of our experiments and is preserved here to avoid confusion with our data sets).

$$
\begin{aligned}
T_{7401} = \; & ((\mathrm{aac},0),\ (\mathrm{aca},0),\ (\mathrm{acb},0),\ (\mathrm{bac},0),\\
& (\mathrm{cac},0),\ (\mathrm{aaaa},0),\ (\mathrm{aaab},0),\ (\mathrm{aaba},0),\\
& (\mathrm{aabb},0),\ (\mathrm{aabc},0),\ (\mathrm{abaa},0),\ (\mathrm{abab},0),\\
& (\mathrm{abba},0),\ (\mathrm{abbb},0),\ (\mathrm{abbc},0),\ (\mathrm{abca},0),\\
& (\mathrm{abcb},0),\ (\mathrm{abcc},0),\ (\mathrm{acca},0),\ (\mathrm{accb},0),\\
& (\mathrm{accc},0),\ (\mathrm{baaa},0),\ (\mathrm{baab},0),\ (\mathrm{baba},0),\\
& (\mathrm{babb},0),\ (\mathrm{babc},0),\ (\mathrm{bbaa},0),\ (\mathrm{bbab},0),\\
& (\mathrm{bbba},0),\ (\mathrm{bbbb},0),\ (\mathrm{bbbc},0),\ (\mathrm{bbca},0),\\
& (\mathrm{bbcb},0),\ (\mathrm{bbcc},0),\ (\mathrm{bcaa},0),\ (\mathrm{bcab},0),\\
& (\mathrm{bcba},0),\ (\mathrm{bcbb},0),\ (\mathrm{bcbc},0),\ (\mathrm{bcca},0),\\
& (\mathrm{bccb},0),\ (\mathrm{bccc},0),\ (\mathrm{caaa},0),\ (\mathrm{caab},0),\\
& (\mathrm{caba},0),\ (\mathrm{cabb},0),\ (\mathrm{cabc},0),\ (\mathrm{cbaa},0),\\
& (\mathrm{cbab},0),\ (\mathrm{cbba},0),\ (\mathrm{cbbb},0),\ (\mathrm{cbbc},0),\\
& (\mathrm{cbca},0),\ (\mathrm{cbcb},0),\ (\mathrm{cbcc},0),\ (\mathrm{ccaa},0),\\
& (\mathrm{ccab},0),\ (\mathrm{ccba},0),\ (\mathrm{ccbb},0),\ (\mathrm{ccbc},0),\\
& (\mathrm{ccca},0),\ (\mathrm{cccb},0),\ (\mathrm{cccc},0),\ (\mathrm{cc},1),\ (\mathrm{ca},1),\\
& (\mathrm{b},1),\ (\mathrm{aa},1),\ (\mathrm{ac},0),\ (\mathrm{a},1),\ (\mathrm{c},1))
\end{aligned}
$$

The target CDL T_{7401} contains 70 nodes. The generater of optimized lists of good training examples (cf. chapter 4) generates some list ${}^{opt}LEX(T_{7401})$ of only 74 cases, i.e. a considerably small set of test cases, when arranged appropriately, suffices to learn the quite complex target object T_{7401} correctly.

Next, we will present the list of words which occur in ${}^{opt}LEX(T_{7401})$, just for completeness. The corresponding class identifiers 0 resp. 1 are omitted, for readability.

The list of words in ${}^{opt}LEX(T_{7401})$ in the correct order: (aac aca acb bac cac aaaa aaab aaba aabb aabc abaa abab abba abbb abbc abca abcb abcc acca accb accc baaa baab baba babb babc bbaa bbab bbba bbbb bbbc bbca bbcb bbcc bcaa bcab bcba bcbb bcbc bcca bccb bccc caaa caab caba cabb cabc cbaa cbab cbba cbbb cbbc cbca cbcb cbcc ccaa ccab ccba ccbb ccbc ccca cccb cccc acc cc ca b aa ac a c acc ac ac)

To perform sufficiently many random experiments with a list of 74 elements is quite difficult, because there are 74! different permutations. The factorial of 74 is an integer number with 108 digits. We peformed only 655 850 individual learning experiments, which means 655 850 times feeding in the 74 words above in another randomly generated order, 655 850 times generating subsequently 74 hypothetical CDLs and finally comparing the result to the target CDL T_{7401}. In fact, each of the final 655 850 comparisons means to decide whether or not the ultimately learnt hypothesis generates resp. accepts the same language as T_{7401} does.

Figure 11 is displaying the learning system's state after successfully learning from good examples.

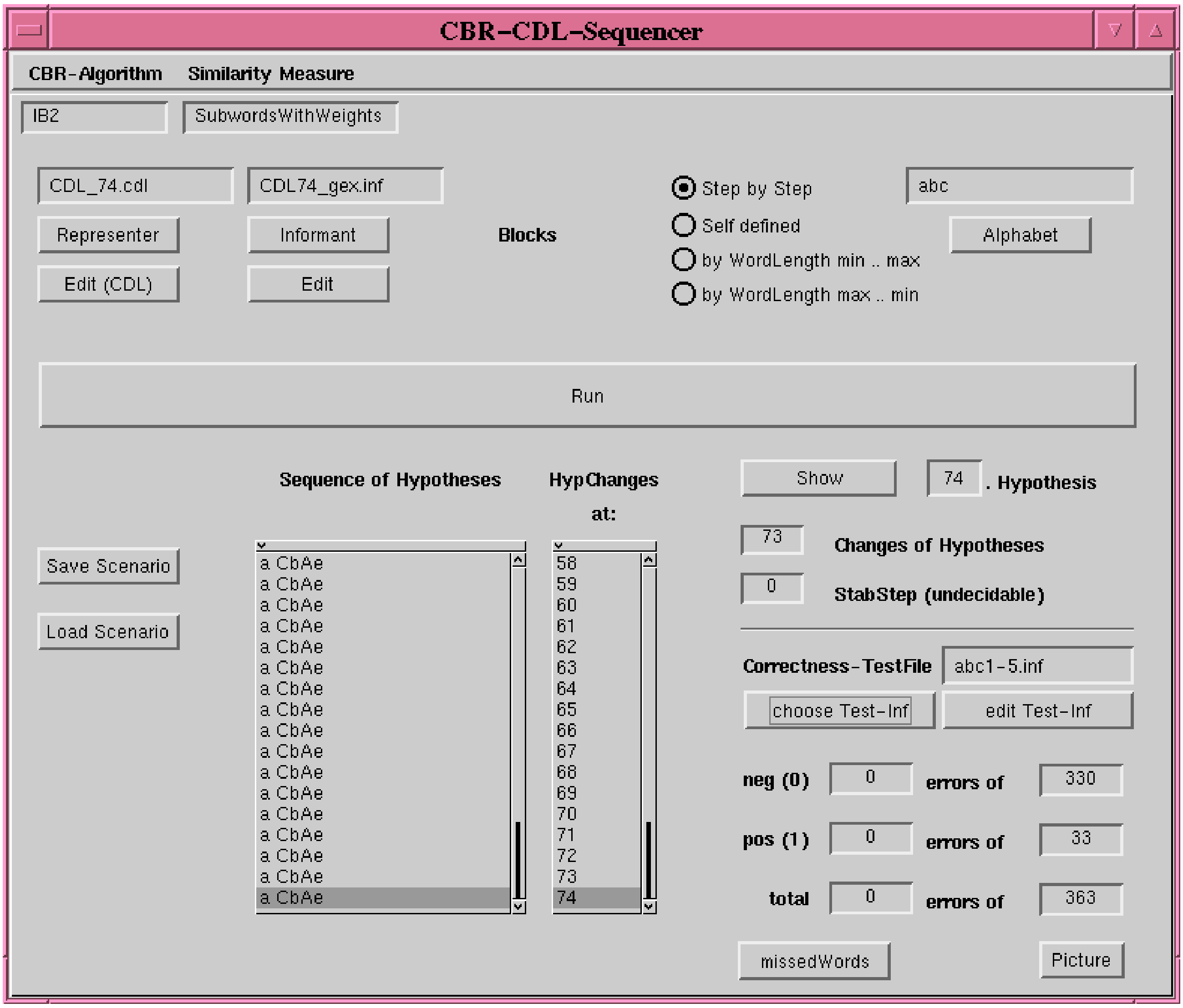

Figure 11: The TIC System after Successfully Learning T_{7401} from the Good Example List “CDL74_gex.inf”

As the figure before shows, the learning goal been reached successfully in this particular case. The next figure is illustrating the progress of the learning system during the 74 steps of this particular run.

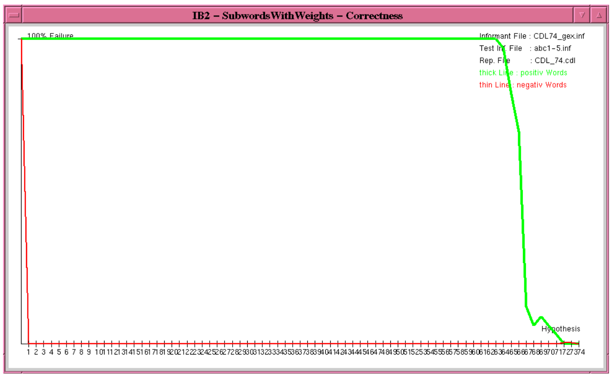

Figure 12: Progress Towards Success in Learning

Two further learning runs are illustrated by the two following displays, respectively. Like in the statistics before, the green line indicates the error rate on positive examples whereas the red line refers to the negative examples.

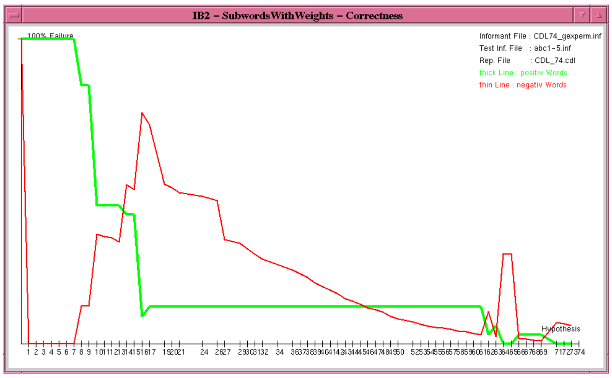

Figure 13: Report on another of the 655 850 Runs

Potentially, we could present a documentation like that about each individual learning experiment.

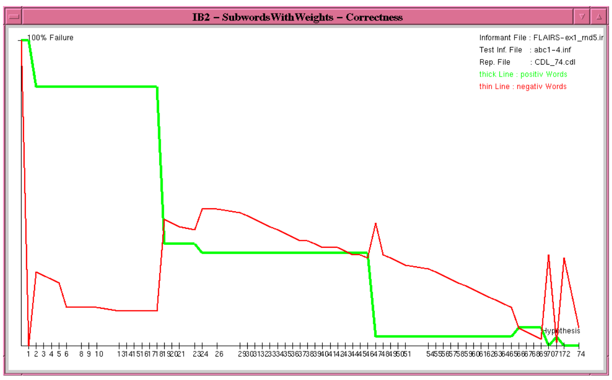

Figure 14: Report on a Third of the 655 850 Runs

Just for illustration, we complete the reported screen dumps from the experiments in learning T_{7401} with a display of the ultimately reached hypothesis after the stepwise learning displayed in figure 14.

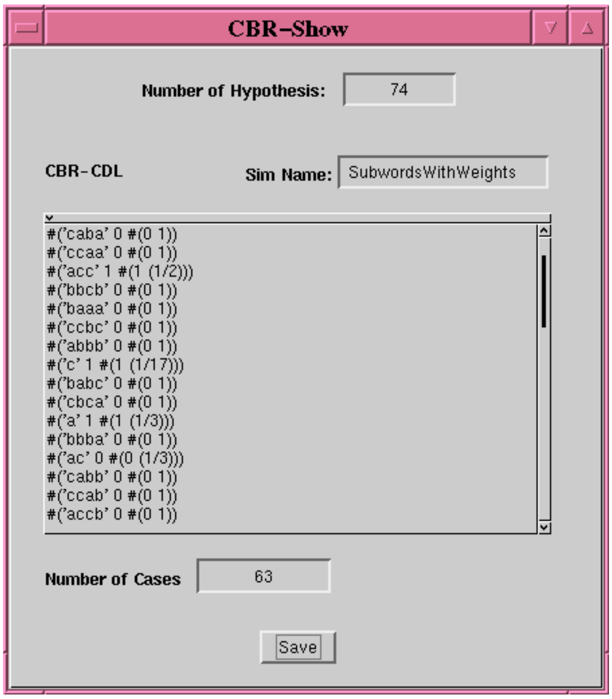

Figure 15: Final Hypothesis after a Third among the Total Number of 655 850 Experiments to Learn the CDL T_{7401}

It follows a complete report on the whole set of 655 850 individual learning runs.

Number of Permutations	Sucesses	Failures	Rate of Success
655 850	3 921	651 929	0.59 %

Recall the results on learning from arbitrarily arranged good examples in chapter 5.2 above. There, the rate of success was also far below 1%, i.e. completely unacceptable as the basis for any effort towards computer-supported knowledge acquisition.

To sum up the report of the present chapter, although the 74 words of the list of good examples ${}^{opt}LEX(T_{7401})$ given in the right order are provably sufficient to learn the target CDL T_{7401}, it is almost impossible to preserve learnability if this underlying order is changed.

In chapter 5.5 we will come back to the long sample CDL T_{7401} investigated here. In this closing part of chapter 5, we will present an aproach exceeding the former presentation of [JD97b]. Before, we proceed as in [JD97b].

5.4 Complete Sets of Elementary Experiments

To contrast the experimental exploration above, we have developed some considerably different setting.

It is quite obvious that the structure of some CDL is not only a matter of predicates located in certain nodes, it is also a matter of relating neighbouring nodes to one another. There might be sublists with identical classification behaviour. The first 63 nodes of T_{7401}, for instance, form an impressive example, in this respect. Those sublists might be located more at the beginning or closer to the end of some given CDL. Structural properties of this type might be of some importance.

For some systematic exploration, we decided to take some list pattern

$$((ab, ?), (c, ?), (a, ?), (b, ?))$$

and investigate all its potential instantiations. In contrast to the experimental explorations reported above, we aimed at a complete coverage of all reasonable learning runs. This intention, naturally, imposes severe restrictions on the size of lists which can be inspected. For this reason, we started with a simple pattern as displayed above.

There are the extremal cases that all nodes are either labelled 1 or 0. The corresponding CDLs will be called T_1 and T_2, respectively. Other instantiation may have one, two, or at most three alternations of classification values. We systematically study all of them. Here is an overview of all possible instantiations of the underlying pattern:

$$\begin{aligned} T_1 &= ((ab,1),(c,1),(a,1),(b,1)) \\ T_2 &= ((ab,0),(c,0),(a,0),(b,0)) \end{aligned}$$

$$\begin{aligned} T_3 &= ((ab,1),(c,0),(a,0),(b,0)) \\ T_4 &= ((ab,0),(c,1),(a,1),(b,1)) \\ T_5 &= ((ab,1),(c,1),(a,0),(b,0)) \\ T_6 &= ((ab,0),(c,0),(a,1),(b,1)) \\ T_7 &= ((ab,1),(c,1),(a,1),(b,0)) \\ T_8 &= ((ab,0),(c,0),(a,0),(b,1)) \end{aligned}$$

$$\begin{aligned} T_9 &= ((ab,1),(c,0),(a,1),(b,1)) \\ T_{10} &= ((ab,0),(c,1),(a,0),(b,0)) \\ T_{11} &= ((ab,1),(c,1),(a,0),(b,1)) \\ T_{12} &= ((ab,0),(c,0),(a,1),(b,0)) \\ T_{13} &= ((ab,1),(c,0),(a,0),(b,1)) \\ T_{14} &= ((ab,0),(c,1),(a,1),(b,0)) \end{aligned}$$

$$\begin{aligned} T_{15} &= ((ab,1),(c,0),(a,1),(b,0)) \\ T_{16} &= ((ab,0),(c,1),(a,0),(b,1)) \end{aligned}$$

By means of the theoretical concepts sketchend in chapter 4, one can easily generate lists of good examples to every of these CDLs. The optimized versions of these lists look as follows:

for T_1: $(c,1),(a,1),(b,1)$
for T_2: $(c,0),(a,0),(b,0)$
for T_3: $(c,0),(abc,1),(a,0),(b,0),(ab,1),(abc,1)$
for T_4: $(ab,0),(c,1),(a,1),(b,1)$
for T_5: $(a,0),(b,0),(ab,1),(c,1),(ab,1)$
for T_6: $(ab,0),(c,0),(a,1),(b,1)$
for T_7: $(b,0),(ab,1),(a,1),(c,1)$
for T_8: $(c,0),(a,0),(b,1)$
for T_9: $(ab,1),(a,1),(bac,0),(c,0),(abac,1),(b,1),(abac,1),(bac,0)$
for T_{10}: $(ab,0),(a,0),(bac,1),(b,0),(c,1),bac,1)$
for T_{11}: $(b,1),(ba,0),(cba,1),(a,0),(ab,1),(c,1)$
for T_{12}: $(ab,0),(b,0),(ba,1),(c,0),(a,1)$
for T_{13}: $(b,1),(bc,0),(abc,1),(a,0),(ab,1),(c,0),(abc,1)$
for T_{14}: $(ab,0),(b,0),(bc,1),(c,1),(a,1)$
for T_{15}: $(a,1),(cba,0),(cbab,1),(b,0),(ab,1),(ba,1),(c,0),(cbab,1),(cba,0)$
for T_{16}: $(b,1),(ab,0),(ba,0),(cba,1),(c,1),(a,0),(cba,1)$

In the most complex situation of T_{15}, there is an optimized list of 9 good examples, i.e., there exist 362 880 possible permutations. So, we have been able to perform learning experiments for each of the CDLs above on every permutation of its specific optimized list of good examples.

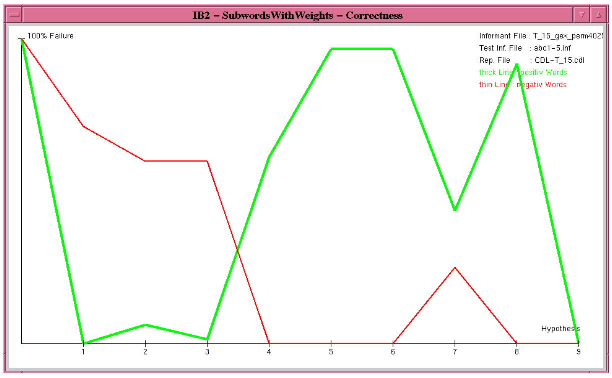

Figure 16: Statistics of an Attempt to Learn T_{15}

Figure 16 is showing the rate of success during one particular run from the more than 400 000 runs of the series of experiments reported here. In fact, this is one of the few runs in which T_{15} has been learnt successfully. The ultimately correct hypothesis after processing this specific permutation of the list of good examples is displayed in figure 15 on the following page.

It is left to the reader to check that the generated hypothesis is indeed semantically equivalent to T_{15}.

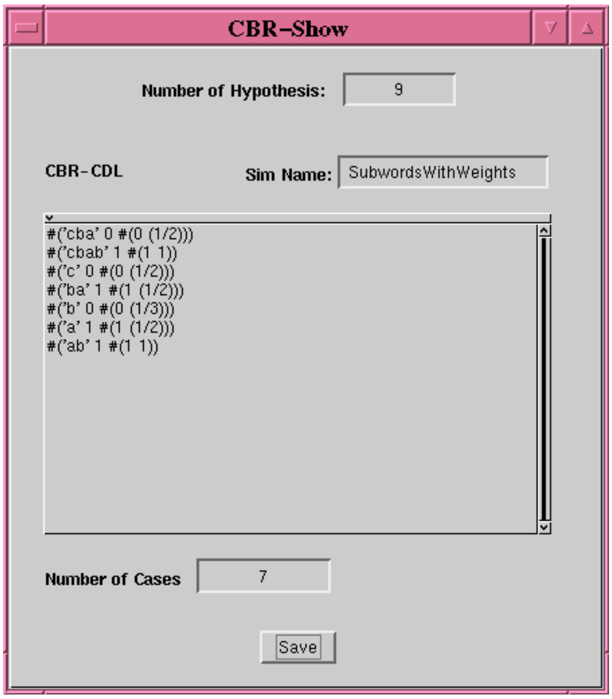

Figure 17: The CDL T_{15} Sucessfully Identified

The following table provides a complete overview of all experimental results. The CDLs are grouped together according to the value of the first classification label. The motivation for this structuring is to isolate a few trivialities which are due to syntactic reasons.

Besides the trivially learnable CDLs T_1, T_2, T_4, T_6, and T_8, learning turns out to be difficult, again.

It is quite difficult to imagine learning problems which are simpler than learning one of the CDLs considered in the present chapter. Practically relevant problems might be usually of a remarkably higher complexity.

CDL	Permutations	Sucesses	Failures	Rate of Success
T_1	6	6	0	100.0
T_3	720	128	592	18.0
T_5	120	40	80	33.0
T_7	24	12	12	50.0
T_9	40320	280	40040	0.7
T_{11}	720	30	690	4.2
T_{13}	5040	142	4898	2.8
T_{15}	362880	1816	361064	0.5
T_2	6	6	0	100.0
T_4	24	24	0	100.0
T_6	24	24	0	100.0
T_8	6	6	0	100.0
T_{10}	720	48	672	7.0
T_{12}	120	35	85	29.0
T_{14}	120	15	105	12.5
T_{16}	5040	930	4110	18.4

Furthermore, we have experimentally investigated learning only under the additional assumption that the examples presented are known to form a list of good examples from which learning provably works. Nevertheless, in all non-trivial cases, learning from unordered cases mostly fails. Even worse, as soon as the number of experiments is exceeding the size of toy examples, the rate of success is becoming catastrophically small.

5.5 The Potentials of Redundancy

In response[3] to the overall impression of the negative results gained from more than 1 000 000 of case-based learning experiments reported above, we searched for any knowledge source to complement the loss of information resulting from only randomly arranging good examples. As sample target we have chosen the CDL T_{7401} investigated in chapter 5.3 above.

Assume that one knows approximately the set of words which might be candidates for becoming good examples. Then we are interested in questions of the following type.

- If these candidates for becoming good examples are frequently repeated, does this compensate for the lack of information about appropriate orderings?
- In case this works, how sensitive is this approach to a change of these examples' frequency?
- How closely is some crucial frequency related to structural knowledge about the target object?

In particular the last question points to some difficulty: It might turn out that some minimal frequency as mentioned does exist, but that finding the right frequency implicitly assumes a priori knowledge about the knowledge to be acquired.

For the readers convenience, we display the optimized list of good examples sufficient for learning T_{7401} again:

(aac aca acb bac cac aaaa aaab aaba aabb aabc abaa abab abba abbb abbc abca abcb abcc acca accb accc baaa baab baba babb babc bbaa bbab bbba bbbb bbbc bbca bbcb bbcc bcaa bcab bcba bcbb bcbc bcca bccb bccc caaa caab caba cabb cabc cbaa cbab cbba cbbb cbbc cbca cbcb cbcc ccaa ccab ccba ccbb ccbc ccca cccb cccc acc cc ca b aa ac a c acc ac ac)

[3]Coincidently, at the time when our experiments have been running in Leipzig, Germany, the question for the potentials of redundancy has been asked during a presentation of our results which was delivered by the first author during his visit to Fukuoka, Japan, in March 1997. In presenting these results here, we are extending [JD97b] substantially.

If these words are presented in exactly this order with exactly the repetitions shown, the algorithm IB2$^\sigma$ is able to acquire some case base and to tune the corresponding weights of cases such that the finally resulting hypothesis represents T_{7401} equivalently.

In the particular problem under investigation, words of length 4 are obviously of some crucial importance. Driven by our a priori knowledge about the target object and about the good examples, we designed three series of experiments.

In every series, one learning experiment means to feed in 1 000 randomly generated cases. Repetitions may occur at random. In the first setting, we have chosen the random cases only among words up to the crucial length of 4. In the second series, the random information ranges over words up to length 5. A final third series takes all words up to length 6 into account. In these series, we have performed 25 900, 16 600 and 12 000 individual learning experiments, respectively.

Series 1	1 000 Cases up to Length 4
Number of Experiments	25 900
Successes	20 069
Failures	5 831
Rate of Sucess (%)	77.5
Maximal Number of Cases	105
Minimal Number of Cases	63
Average Size	82.8

The ratio of success reported above seems quite reasonable. However, the reader should take into account that this result is based on the fact that the randomly chosen examples come as close as possible to the needs of identifying this particular target phenomenon. Due to the fact that there are only 120 different non-empty words up to length 4, every word including every good example occurs about 8 times on the average, if 1 000 cases are presented.

Thus, it seemed reasonable to undertake further experiments.

Series 2	1 000 Cases up to Length 5
Number of Experiments	16 600
Successes	24
Failures	16 576
Rate of Sucess (%)	0.15
Maximal Number of Cases	190
Minimal Number of Cases	90
Average Size	144.4

The second experimental setting illustrates that already the slightest possible deviation from the ideal information results in a break down of learnability. If the case taken into account may vary up to length 6, the results are clearly more disappointing, as the third setting exhibits:

Series 3	1 000 Cases up to Length 6
Number of Experiments	12 000
Successes	0
Failures	12 000
Rate of Sucess (%)	0
Maximal Number of Cases	303
Minimal Number of Cases	122
Average Size	221

There was not a single success of learning among the 12 000 experiments undertaken.

The experiments of the three series reported above are characterized by redundancy which results from repetitions which, on the one hand, are driven by estimates of the target object's size and which, on the other hand, are still only loosely constrained as arbitrary cases are permitted.

To complement this approach, we developed another setting of experiments in which the knowledge about good examples is combined with an attempt to provide redundancy.

The setting is as follows: We take the 74 elements of the good example list describing T_{7401} and construct a list of n times 74 cases by an n-fold concatenation of this list. The particular factor n is called the degree of redundancy. Then, we generate randomly a large number of permutations of these n times 74 cases and run learning experiments as before.

Series 1		Series 2	
Degree of Redundancy	2	Degree of Redundancy	3
Experiments	40 000	Experiments	40 000
Learning Results			
Success	884	Success	1 969
Failure	39 116	Failure	38 031
Ratio	2.2%	Ratio	4.9%
Size of the Final Hypothesis			
maximal	71	maximal	71
minimal	60	minimal	60
average	65.3	average	65.9

The results are presented here in the one table before and in the three tables following. Altogether they are displaying 320 000 individual learning runs. During the shortest 40 000 of these runs, there are fed in 148 cases per run, whereas in each of the longest 40 000 runs there are processed 666 cases.

Series 3		Series 4	
Degree of Redundancy	4	Degree of Redundancy	5
Experiments	40 000	Experiments	40 000
Learning Results			
Success	2 859	Success	36 513
Failure	37 141	Failure	3 487
Ratio	7.1%	Ratio	8.7%
Size of the Final Hypothesis			
maximal	71	maximal	71
minimal	61	minimal	62
average	66.1	average	66.2

The reader may identify some tendency of a slightly growing ratio of success with a growing frequency of repetitions. This is quite understandable as every higher degree of redundancy, in fact, means to perform experiments exactly as before, but afterwards feeding in another set of good examples.

Series 5		Series 6	
Degree of Redundancy	6	Degree of Redundancy	7
Experiments	40 000	Experiments	40 000
Learning Results			
Success	3 602	Success	3 722
Failure	36 398	Failure	36 278
Ratio	9%	Ratio	9.3%
Size of the Final Hypothesis			
maximal	71	maximal	71
minimal	62	minimal	62
average	66.2	average	66.2

Nevertheless, the results are extraordinarily disappointing. Recall that we are taking into account only those cases from which learning is guaranteed provided they are presented in the right order. As one may not know such an ordering in every detail, under realistic circumstances, one may try to compensate this lack of knowledge by repeatedly presenting the information available. The experiments exhibit that this does not work.

Series 7		Series 8	
Degree of Redundancy	8	Degree of Redundancy	9
Experiments	40 000	Experiments	40 000
Learning Results			
Success	3 707	Success	4 037
Failure	36 293	Failure	35 963
Ratio	9.3%	Ratio	10.1%
Size of the Final Hypothesis			
maximal	71	maximal	71
minimal	63	minimal	63
average	66.2	average	66.3

We admit that there might be several combinatorial phenomena not sufficiently well-understood, at the moment. Some deeper investigation and the answers to some more possibly open questions might provide the one or the other explanation of certain phenomena we have exhibited. Nevertheless, those explanations can not cause the disappearance of these difficulties.

A ratio of success of 10% or below (cf. the tables above) is usually completely unacceptable in realistic applications. But – even worse – our testbed does not deal with realistic applications; it deals with some theoretically well-understood domain of formal languages which is characterized by certain properties of simplicity:

- The languages acceptable by systems of the type under investigation are all regular, i.e. quite simple w.r.t. the CHOMSKY hierarchy (cf. [HU79]).
- All the target objects to be learnt are especially simple in structure and – even nicer – are itself composed from cases directly.
- The target languages are known to be PAC learnable (cf. [SS92]).
- The target languages are even known to be learnable in the limit in a case-based manner within some particular setting (cf. [SJL94]).

There might be further questions of interest, but we found our experimentations above of altogether more than 1.5 millions of individual learning runs sufficient, for the moment. Although, we have a bunch of further experiments documented, we refrain from a presentation.

6 CONCLUSIONS

The aim of the present paper is to contribute to a better understanding of case-based reasoning, in general, and of case-based learning, in particular. We focus on quite expressive classes of CBR systems called *logical case memory systems* (cf. [Jan97]).

We are especially interested in those charming CBR paradigms like the one implemented by IB2, for instance, and circumscribed as follows: *Given any CBR system, apply it. Whenever it works sucessfully, do not change it. Whenever it fails on some input case, add this experience to the case base. Don't change anything else.*

For the purpose of an in-depth discussion, we focussed on an extremely simple and well-understood class of logical case memory systems: *containment decision lists.*

For learning containment decision lists, we tried completely unsupervised learning experiments, first. They failed completely. From some critical inspection of the difficulties, we have been lead to the concept of *good example lists.* Those lists are known to be sufficient for learning. On the one hand, they are algorithmically well-defined and can be generated automatically. On the other hand, they might be difficult to find, if the target phenomenon is not sufficiently well-understood. Even if everything needed to build those lists of good examples is known, it might be an additional problem to arrange this knowledge appropriately. We report 1 562 440 learning experiments, some of them consisting of hundreds or even thousands of individual learning steps, to explore the importance of finding an appropriate ordering of information presented as a basis for learning. The results are documented and illuminate the sensitivity of case-based learning to the ordering of information quite well.

We are convinced that case-based learning of containment decision lists is considerably simpler than most problems of knowledge acquisition in the wild. Thus, *user guidance* for acquiring knowledge in a case-based manner is practically at least as important as exhibited in the prototypical domain of our present investigations, it *is just inevitable.*

In their right perspective, the experimental settings developed and explored above may be understood as some methodological *lower bound* to realistic settings of case-based learning. The toy application domain chosen for the present work is extremely simple such that almost every other interesting application domain of some proper relevance is of an intuitively larger complexity. It is quite unlikely that in those realistic domains such very simple algorithmic ideas succeed which provably fail in the toy domain. This circumscribes our understanding of a "lower bound" provided by the present findings.

Furthermore, another methodological aspect seems worth to be briefly discussed. Case-based learning might usually take place in some embedding knowledge processing environment. Under those particular circumstances, case-based learning might work successfully either due to additional knowledge sources or under weakened requirements. There is, obviously, an urgent need to make those assumptions explicit to justify learning approaches and to clearly discriminate those circumstances under which there is not much hope for unsupervised learning as exemplified above.

Acknowledgments

The first author's work has been substantially supported by a visit of more than 2 months to the Meme Media Laboratory of Hokkaido University, Sapporo, Japan. He furthermore acknowledges the fruitful scientific cooperation with colleagues and friends at the Kyushu University, Fukuoka, Japan. This cooperation made the present report possible.

The investigations presented here are part of a more comprehensive research project sponsored by the German research Fund (DFG) und grant no. Ja 566/3–3. Our project partners Michael M. Richter, Christoph Globig (both at Kaiserslautern University) and Steffen Lange (at HTWK Leipzig) have always had a serious impact on our work. We gratefully acknowledge their support. The reader may find many further interesting results relating inductive learning, in general, and case-based learning, in particular, in several publications by these authors.

Udo Burghardt and Stephan Frind have been coauthors of the second author of the present report in developing the TIC system underlying our experiments. Their contribution is highly appreciated. A guided tour to the TIC system is presented in [BDF96].

A shorter version of this paper (cf. [JD97a]) will be presented at FLAIRS–97, the Florida Artificial Intelligence Research Symposium in Daytona Beach, FL, USA, im May 1997. Conference proceedings can usually not provide sufficient space to report comprehensive series of experiments in sufficient detail. Thus, the present report together with its predecessor version [JD97b] should be understood as some addittional source of information related to the more condensed writing in [JD97a].

References

[Aha91] David W. Aha. Case-based learning algorithms. In Ray Bareiss, editor, *Proceedings of the DARPA Case-Based Reasoning Workshop, May 8 - 10, 1991, Washington DC, USA*, pages 147–158. Morgan Kaufmann, 1991.

[AKA91] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991.

[AP94] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.

[AS83] Dana Angluin and Carl H. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–269, 1983.

[BDF96] Udo Burghardt, Volker Dötsch, and Stephan Frind. TIC – ein Testrahmen für IND–CBL. Communications of the Algorithmic Learning Group CALG–01/96, Hochschule für Technik, Wirtschaft und Kultur Leipzig, FB Informatik, Mathematik und Naturwissenschaften, January 1996.

[BJST93] Katy Börner, Klaus P. Jantke, Siegfried Schönherr, and Elisabeth-Charlotte Tammer. Lernszenarien im fallbasierten Schließen. FABEL–Report 14, Gesellschaft für Mathematik und Datenverarbeitung mbH, Forschungsbereich Künstliche Intelligenz, November 1993.

[BW96] Ralph Bergmann and Wolfgang Wilke. On the role of abstraction in case-based reasoning. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning, Proc., 3rd European Workshop on Case–Based Reasoning (EWCBR'96), November 14–16, 1996, Lausanne, Switzerland*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 28–43. Springer-Verlag, 1996.

[DJ96a] Volker Dötsch and Klaus P. Jantke. Good examples in learning containment decision lists. In Werner Dilger, Michael Schlosser, Jens Zeidler, and Andreas Ittner, editors, *Machine Learning, 1996 Annual Meeting of the Special Interest Group of Machine Learning of the German Computer Science Society (GI), Chemnitzer Informatik-Berichte CSR–96–06*, pages 18–23. TU Chemnitz, 1996.

[DJ96b] Volker Dötsch and Klaus P. Jantke. Solving stabilization problems in case-based knowledge acquisition. In Paul Compton, Riichiro Mizoguchi, Hiroshi Motoda, and Tim Menzies, editors, *Pacific Knowledge Acquisition Workshop, Oktober 23–25, 1996, Sydney, Australia*, pages 150–169. University of New South Wales, Department of Artificial Intelligence, 1996.

[DOC+93] Helge Dürschke, Wolfgang Oertel, Carl-Helmut Coulon, Wolfgang Gräter, Bernd Linowski, M. Nowak, Katy Börner, Elisabeth-Charlotte Tammer, Markus Knauff, Ludger Hovestadt, and Brigitte Bartsch-Spörl. Approaches to similarity in FABEL. FABEL–Report 13, Gesellschaft für Mathematik und Datenverarbeitung mbH, Forschungsbereich Künstliche Intelligenz, July 1993.

[FC93] FABEL-Consortium. Survey of FABEL. FABEL–Report 2, Gesellschaft für Mathematik und Datenverarbeitung mbH, Forschungsbereich Künstliche Intelligenz, February 1993.

[FKW89] Rusiņš Freivalds, Efim B. Kinber, and Rolf Wiehagen. Inductive inference from good examples. In Klaus P. Jantke, editor, *Analogical and Inductive Inference (AII'89) Proc. 2nd International Workshop, Reinhardsbrunn Castle, GDR, October 1–6, 1989*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 1–17. Springer-Verlag, 1989.

[FKW93] Rusiņš Freivalds, Efim B. Kinber, and Rolf Wiehagen. On the power of inductive inference from good examples. *Theoretical Computer Science*, 110:131–144, 1993.

[FKW95] Rusiņš Freivalds, Efim B. Kinber, and Rolf Wiehagen. Learning from good examples. In Klaus P. Jantke and Steffen Lange, editors, *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 49–62. Springer-Verlag, 1995.

[GJLS97] Christoph Globig, Klaus P. Jantke, Steffen Lange, and Yasubumi Sakakibara. On case-based learnability of languages. *New Generation Computing*, 15(1):59–83, 1997.

[Gol67] E Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[HU79] John E. Hopcroft and Jeffrey. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[Jan89] Klaus P. Jantke. Algorithmic learning from incomplete information: Principles and problems. In Jürgen Dassow and Jozef Kelemen, editors, *Machines, Languages, and Complexity*, volume 381 of *Lecture Notes in Computer Science*, pages 188–207. Springer-Verlag, 1989.

[Jan92] Klaus P. Jantke. Case based learning in inductive inference. In *Proc. 5th Annual ACM Workshop on Computational Learning Theory, (COLT'92), July 27–29, 1992, Pittsburgh, PA, USA*, pages 218–223. ACM Press, 1992.

[Jan94] Klaus P. Jantke. Nonstandard concepts of similarity in case-based reasoning. In Hans-Hermann Bock, Wolfgang Lenski, and

Michael M. Richter, editors, *Information Systems and Data Analysis: Prospects – Foundations – Applications, Proceedings of the 17th Annual Conference of the GfKl, Univ. of Kaiserslautern, 1993*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 28–43. Springer-Verlag, 1994.

[Jan97] Klaus P. Jantke. Logical case memory systems: Foundations and learning issues. Technical report, Forschungsinstitut für InformationsTechnologien Leipzig e.V., Forschungsbericht 97–1, January 1997.

[JD97a] Klaus P. Jantke and Volker Dötsch. The necessity of user guidance in case-based knowledge acquisition. In *FLAIRS–97, Proc. Florida AI Research Symosium, Daytona Beach, FL, USA, May 11–14, 1997*, 1997.

[JD97b] Klaus P. Jantke and Volker Dötsch. Theoretical investigations and experimental explorations of the necessity of user guidance in case-based knowledge acquisition. Technical Report MEME–IMP–3, Hokkaido University Sapporo, Meme Media Laboratory, February 1997.

[JL93] Klaus P. Jantke and Steffen Lange. Case–based representation and learning of pattern languages. In Klaus P. Jantke, Shigenobu Kobayashi, Etsuji Tomita, and Takashi Yokomori, editors, *Proc. 4th International Workshop on Algorithmic Learning Theory, (ALT'93), November 8–10, 1993, Tokyo*, volume 744 of *Lecture Notes in Artificial Intelligence*, pages 87–100. Springer–Verlag, 1993.

[JL95] Klaus P. Jantke and Steffen Lange. Case-based representation and learning of pattern languages. *Theoretical Computer Science*, 137(1):25–51, 1995.

[Kol92] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6:3–34, 1992.

[Kol93] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.

[LNW94] Steffen Lange, Jochen Nessel, and Rolf Wiehagen. Language learning from good examples. In Setsuo Arikawa and Klaus P. Jantke, editors, *Algorithmic Learning Theory, Proc. 4th International Workshop on Analogical and Inductive Inference (AII'94) and the 5th International Workshop on Algorithmic Learning Theory (ALT'94), October 10–15, 1994, Reinhardsbrunn Castle, Germany*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 423–437. Springer-Verlag, 1994.

[MJ97] Daniel Matuschek and Klaus P. Jantke. Axiomatic characterizations of structural similarity for case-based reasoning. In *FLAIRS–97, Proc. Florida AI Research Symosium, Daytona Beach, FL, USA, May 11–14, 1997*, 1997.

[OB96] Hugh R. Osborne and Derek G. Bridge. A case base similarity framework. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning, Proc., 3rd European Workshop on Case–Based Reasoning (EWCBR'96), November 14–16, 1996, Lausanne, Switzerland*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 309–323. Springer-Verlag, 1996.

[RS89] Christopher K. Riesbeck and Roger C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Assoc., 1989.

[Sch96] Jörg Walter Schaaf. Fish and shrink. A next step towards efficient case retrieval in large scaled case bases. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning, Proc., 3rd European Workshop on Case–Based Reasoning (EWCBR'96), November 14–16, 1996, Lausanne, Switzerland*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 362–376. Springer-Verlag, 1996.

[SJL94] Yasubumi Sakakibara, Klaus P. Jantke, and Steffen Lange. Learning languages by collecting cases and tuning parameters. In Setsuo Arikawa and K.P. Jantke, editors, *Algorithmic Learning Theory, Proc. 4th International Workshop on Analogical and Inductive Inference (AII'94) and the 5th International Workshop on Algorithmic Learning Theory (ALT'94), October 10–15, 1994, Reinhardsbrunn Castle, Germany*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 533–547. Springer-Verlag, 1994.

[SS92] Yasubumi Sakakibara and Rani Siromoney. A noise model on learning sets of strings. In *Proc. of the 5th ACM Workshop on Computational Learning Theory, COLT'92, July 27-29, 1992, Pittsburgh, PA, USA*, pages 295–302. ACM Press, 1992.