Fishing in a Speech Stream - Angling for a Lexicon

Peter Juel Henrichsen
Center for Computational Modelling of Language (CMOL)
Copenhagen Business School
pjh.isv@cbs.dk

Abstract

We present a learning device able to deduce a set of Danish color and shape terms. Only two data sources are available to the learner: A phonetic transcription of a human informant solving a description task, and a minimal formal model of the picture being described. The system thus contains no preconceived lexical, morphological, or semantic categories. The test data are from the phonetic corpus DanPASS, a standard Danish reference corpus. The learning device, called InShape-2, is an early result of an ambitious research programme at CMOL on data-driven language learning.

1 Introduction

Imagine a device able to learn the lexical units and linguistic structures occurring in a natural language discourse. The device would have access to data of only two sorts: a sound recording of a language user[1] and a formal representation of the scene (physical or mental) being talked about. In particular, the device would have no built-in language model, no grammatical or lexical expectations, no phonological bias. Such a device would not only be of practical value, it could also play a role as evidence in the still unsettled debate about linguistic universals and innate language capacities. In addition, it would have obvious use as an instrument for first language (L1) acquisition studies.

In this paper we present a toy system called InShape-2, intended as a small step towards the general learning device.[2] We begin with a short introduction of the speech data that we have used, followed by some methodological considerations, a presentation of our implementation, and some test results. The paper concludes with a discussion of the limitations of the current framework – and how to remove those limitations.

2 A learning experiment

We take our starting point in a simple-minded world of geometrical figures.

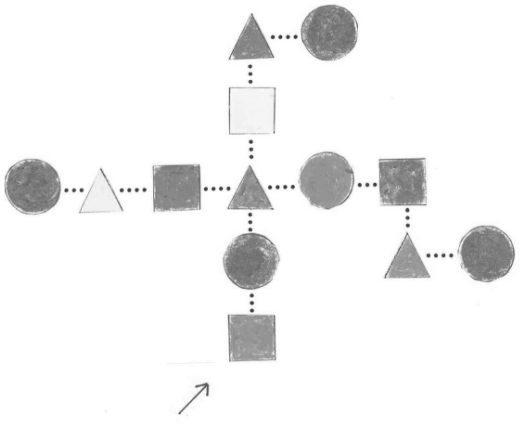

Figure 1. Corpus DanPASS: The geometrical network.[3]

The design in fig. 1 is borrowed from the phonetic corpus DanPASS (Danish Phonetically Annotated Spontaneous Speech, Grønnum (2009)). The corpus was collected and annotated by phonologists at Copenhagen University, and it has mainly been used for phonetic research. As a matter of fact, DanPASS comes with a disclaimer: "The intention was to supply a corpus for acoustic and perceptual phonetic investigations. That is, the primary goal is neither syntactic, pragmatic, socio-linguistic, psychological, nor whichever other aspect of spoken language one might wish to investigate." (http://www.danpass.dk). We are thus dealing with data that were not prepared for, or biased towards, experiments addressing lexical and semantic issues.

[1] This paper, and the associated line of research, is primarily aimed at the auditory part of the speech situation, not the related visual and tactile cues (gestures, gaze directions, body movements, etc).

[2] InShape-2, here presented for the first time, is a complete rewrite of the earlier program InShape-1 (Henrichsen 2010, now deprecated). More details below.

[3] The 13 objects are colored; consult the original graph at http://www.cphling.dk/~ng/danpass_webpage/figs/geometry.pdf. Symbols e1..e13 are not included in the original.

Bolette Sandford Pedersen, Gunta Nešpore and Inguna Skadiņa (Eds.)
NODALIDA 2011 Conference Proceedings, pp. 90–97

DanPASS consists of a monologue part and a dialogue part, each containing a number of lab recordings of Danish informants solving various language tasks. The design in fig. 1, call it **G** (for "geometrical network"), was used in a series of description tasks. Eighteen male and female informants were asked to give a complete description of **G** as if to a person who could not see it. The informants were instructed to begin with the object pointed to by the arrow, but otherwise no clues were given as to what specific terms should be used for e.g. colors and shapes, nor to the sequence in which the objects should be named. As a result, the sessions vary considerably with respect to term selection, description strategy, rhetorical and grammatical style, informational redundancy, and duration. The shortest session is only 45 seconds, the longest more than four times longer.

The goal of the InShape-2 project was to design a robust learning device able to determine for each **G** session its basic vocabulary (the shape and color terms being used) and its temporal composition (the sequence of naming events) based on two information sources only, (i) the phonetic transcription data available with DanPASS and (ii) a minimal formal representation of **G**.

At this point we are not making claims of cognitive isomorphism. We simply wanted to demonstrate that linearly ordered phonetic data can be sufficiently rich for extraction of structured lexical and semantic information in a process governed by a specific learning purpose.

2.1 Formal preliminaries

We refer to the geometrical objects of **G** as **e1**, **e2**, .., **e13**, reading the branches of the network clockwise (see fig. 1). For reference to individual sessions, we use the DanPASS identifiers m_*n*_g (m='monologue', *n*=informant code, g='geometrical network').

For reasons of computational tractability, some basic constraints had to be hardwired into the implementation. Some of them concern the naming events of the sessions, i.e. those utterances by which informants refer to the particular objects. Examples of naming events are "en gul firkant" (a yellow square), "den lilla trekant" (the purple triangle), but also "en rund grøn" (a round green) using two adjectives instead of a standard noun phrase, and "en grøn tre- en blå trekant" (a green tri- a blue triangle) including a self-correction. Since the informants were instructed to include all objects in their descriptions, the naming events are expected to occur in sequences of 13. Such a complete sequence referring to each object in turn, we call a path.

[**e1**,**e2**,**e3**,**e7**,**e8**,**e9**,**e4**,**e5**,**e6**,**e10**,**e11**,**e12**,**e13**]

The path above happens to be the one most frequently used by the informants, but by no means the only one. Perhaps surprisingly, no less than seven different paths are represented in the 18 sessions.

Formal requirements

- A session has (at least) one path consisting of 13 unambiguous naming events
- In any path, the first naming event denotes **e1**

In addition to the formal requirements, we also imposed some linguistic conditions for reasons to be discussed shortly.

Linguistic conditions

In any path, all color terms and shape terms must be used consistently in the following sense:

- no synonymy (i.e. no two distinct terms for one color or for one shape)
- no homonymy (no single term can denote a particular color *and* a shape)
- no material inclusion among the terms within a domain (no color term can be a part of another color term, excluding term pairs like "light green"+"green"; similarly for shapes)
- adjacency of terms for shape and color (e.g. not "a circle <u>in a</u> green color")

In **G** sessions there are often multiple naming events referring to one particular object, or several uses of the color and shape terms not referring to any object at all. Some naming events may even refer to non-existing objects, sometimes (but not always) followed by a self-correction. Most of this variation is not critical with respect to InShape learning (and shouldn't be!) as long as at least one well-formed path exists. It should be noticed, though, that the significant naming events may be very unevenly distributed; in session m_009_g, the initial naming event (**e1**) thus occurs after no less than 74 seconds.

Finally, to limit the search space an integrity constraint is adopted formalizing the observation that informants tend to move from object to adjacent object whenever possible rather than jumping arbitrarily around in the network. The constraint is defined in terms of *path complexity* (PC), defined informally as the minimal number of instructions in {'up', 'down', 'left', 'right', 'reset'} needed in an explanation. For instance, the path mentioned above can be explained as: "**e1** *UP* **e2 e3 e7 e8** *RIGHT* **e9** *RESET* **e3** *LEFT* **e4 e5 e6** *RESET* **e3** *RIGHT* **e10 e11** *DOWN* **e12** *RIGHT* **e13**", making PC=8 for this path. Only one other path has PC=8, while ten paths has PC=9. The PC<10 paths include all actually occuring DanPASS naming sequences with the sole exception of the quite entangled m_31_g (PC=10). Moreover, the PC=8 paths are generally preferred (7+3 cases) over the PC=9 (3+2+1+1+0+0+0+0+0+0), adding some psychological plausibility to the complexity measure.[4]

Integrity constraint

- Path complexities >9 are not considered

Following the requirements, conditions, and constraints, a number of sessions must be excluded from the InShape test material leaving a test corpus of 13 sessions. The excluded sessions (see table 1) do however play an important role as negative evidence in the testing of the implementation.

Session	Disqualifying feature
m_003_g	**e9** is wrongly described as "en gul cirkel" (*a yellow circle*)
m_005_g	**e10** is wrongly described as "en rød cirkel" (*a red circle*)
m_008_g	**e3** is wrongly described as "en firkantet lilla" (*a square violet*); the mistake is corrected, but too late to restore the path.
m_013_g	Homonymy: two terms for *square* ("firkant", "kvadrat")
m_031_g	Integrity constraint violation: Path is too entangled

Table 1. Discarded sessions.

It is worth mentioning that even the formally well-formed **G** sessions may be quite informal in style with lots of repetitions and quite verbose explanations. Several sessions contain one or more self-corrections, such as

"en violet en lilla trekant" [m_007_g]
(*a violet a purple triangle*)

"en lilla nej en brun en brun firkant" [m_014_g]
(a violet no a brown a brown square)

"til en blå eller til en grøn cirkel" [m_016_g]
(*to a blue or to a green circle*)

Deducing the lexical paradigms and semantic relations in **G** thus calls for parsing strategies that are not usually taught in courses on formal grammar.

2.2 Color blindness

A curious property of **G** – probably unintended by its designers – is the similarity of the left branch [**e4**,**e5**,**e6**] and the upper branch [**e7**,**e8**,**e9**] concerning shapes and colors. Both branches share the shape sequence [*square*, *triangle*, *circle*] while the color sequences are [*X*, *Y*, *green*] and [*Y*, *X*, *green*], respectively. As colors *X* and *Y* do not occur elsewhere in **G**, an inference engine needs to learn some location terms in order to deduce the intended terms for *red* and *yellow*. Picking the wrong color terms for *X* for *Y*, for the reasons mentioned, we shall refer to as *color blindness*.

3 InShape-2 – the implementation

Skipping over irrelevant programming details, we present the implementation as a pipeline of functional modules. The chain is strictly feed-forward (no backtracking between modules).

Extract_phones > Find_siblings > Assign_shapes > Assign_colors > Build_lex

Figure 2. The InShape-1 functional modules.

Each of the stations in fig. 2 is introduced below, with special emphasis on those implementing the semantic inference system, viz. Assign_shapes and Assign_colors. Other papers expanding on the data preprocessor (Extract_phones) and the growth of lexical structure (Build_lex) are in preparation; Find_siblings is treated at length in Henrichsen (2004).

[4] In InShape-1 the set of well-formed paths were hard-wired into the system whereas InShape-2 derives it directly from the **G**-geometry, preparing the system for experiments with varying geometries.

3.1 Extract_phones

The phonetic data are derived from the TextGrid files (Boersma 2001) included in the DanPASS corpus. Fig. 3 shows a sample from the orthographic tier of m_007_g.[5]

> du starter nederst på papiret med= en= + blå firkant + dernæst går du= + opad= + mod toppen af papiret + og lægger en= grøn= cirkel + og endelig= lægger du= en= = violet en lilla + trekant + fortsætter du opad + og lægger en gul firkant + og en rød trekant + og til højre for den røde trekant men + på= niveau + med trekanten + lægger du en + grøn = cirkel + så vender du tilbage til den= + lilla + trekant= der nu er i midten + til højre + men= på niveau + med den lilla trekant + lægger du en + blå cirkel +

Figure 3. Orthographic transcription.

Extract_phones reads a TextGrid, returning the phonetic transcription in compact form excluding pauses, hesitation, stress pattern, stød, and syllabification, since these features tend to be less consistently annotated by transcribers than vowels and consonants. For m_007_g, the string returned thus begins with [dusdAd0neD0spcp0piCDmEDenblcfiRkant...].

3.2 Find_siblings

Which is the easier to learn, the shape domain or the color domain? Two observations on **G** suggest the answer. Firstly, there are six color types and only three shape types. Secondly, each shape type has a substantial amount of occurrences in the geometrical network (5+4+4), as opposed to the thinner spread of the colors (4+3+2+2+1+1). As data-driven learning typically relies on repetition and limited diversity, the shape domain should, arguably, be addressed first.

The phone stream from Extract_phones must be segmented into potentially meaningful units, interpreted as recurring substrings of phones. For the segmentation analysis we used the Siblings-and-Cousins algorithm of Henrichsen (2004). Due to space limitations, we have to introduce the S&C algorithm very briefly; a thorough presentation is in the paper mentioned. Originally, S&C was suggested as a way of clustering the lexical items (words) occurring in an unannotated text corpus based on their distributional similarity. A main ingredient in the S&C framework is the *proximity* measure comparing the similarity of two types based on the distribution of their adjacent tokens in the corpus.

Modifying the S&C framework to accommodate the current data type, the proximity of two *n*-grams *X* and *Y* occurring in a phonetic transcription *T* is given by the *Prox* formula.

$$Prox(X,Y,T) = \frac{\sum_{z \in Voc} c(z) \cdot \left(1 - \frac{(L_1 - L_2)}{L_1 + L_2}\right)}{c(X)} \cdot \frac{\sum_{z' \in Voc} c(z') \cdot \left(1 - \frac{(R_1 - R_2)}{R_1 + R_2}\right)}{c(X)}$$

Figure 4. Proximity of two n-grams X and Y.

Voc is the set of all tokens in *T*; c(*g*) is the *T* count function, i.e. the number of occurrences of *n*-gram *g* in *T*, and

$$L_1 = c([z\ X])/c([X]);\ \ L_2 = c([z\ Y])/c([Y]);$$
$$R_1 = c([X\ z'])/c([X]);\ \ R_2 = c([Y\ z'])/c([Y]).$$

Intuitively, *Prox* measures the similarity of two *n*-grams occuring in a transcription, based on their left and right context functions. *Prox*-values always range between 0 and 1 for valid input. Kindred *n*-grams (such as two color terms, or two shape terms) tend to score high, while less associated *n*-grams (such as one color term and one shape term) score lower. *Prox*=1 occurs for pairs of *n*-grams with identical distribution of tokens in their immediate surroundings, while a pair of *n*-grams not sharing a single left-side token or right-side token makes *Prox*=0.

```
#62          'f_i_R_k_a_n'          5
1.000000     f_i_R_k_a_n            5
0.321535     t_r_z_k_a_n            8
0.140000     s_i_R_g_0_l            5
0.124675     b_l_c                  4
0.098937     z_R                    4
0.098209     t_C                    4
0.080672     g_r_O_n_s_i_R_g_0_l    4
(...)
```

Figure 5. Sample from the S&C log for m_019_g

[5] '+' = pause, '=' = hesitation with phonation. English word-by-word gloss, with underlining of significant and non-significant naming events: *You start lowest on the-paper with= a= + blue square + thereafter go you= + upwards= + towards the-top of the-paper + and put a= green= circle + and finally= put you= a= = violet a purple + triangle + continue you upwards + and put a yellow square + and a red triangle + and at right of the red triangle but + at= level + with the-triangle + put you a + green = circle + then turn you back to the= + purple + triangle= that now is in the-middle + at right + but= at level + with the purple triangle + put you a + blue circle +*

Sets of *n*-grams with mutually high *Prox*-values are informally called *siblings*. Consider the sample in fig. 5, quoted from the S&C analysis for *X*='f_i_R_k_a_n'. As reported, this particular *n*-gram has five occurrences in the transcription, and it was analysed as the 62th item. In the quoted list, the *Y*s (i.e. the siblings of *X*) are sorted by their associated *Prox* values. Speakers of Danish will notice the difference between this phonetic string and the standard pronunciation for "firkant" (*square*), especially concerning the final part of the word. Whereas the Danish standard phonetic dictionary prescribes a final stop [d], the pronunciations in m_019_g show some variation, with [fiRkan] being the invariant part. Therefore (only) this part is suggested by the Find_siblings module as a potentially meaningful unit – and similarly for *triangle,* t_r_z_k_a_n.

Observe that the 10-gram g_r_Q_n_s_i_R_g_0_l (*greencircle*), even though it is not an acknowledged Danish lexeme, is also suggested as a potentially meaningful unit in the specific learning context of InShape. The prediction of g_r_Q_n_s_i_R_g_0_l as a semantic atom is an effect of the **G** model where circles are generally green, with only one exception. As we will argue, predictions like this should be seen as signs of lexical flair rather than just errors.

As demonstrated in fig. 5, lexical types belonging to the same semantic category, e.g. shape terms, color terms, or direction terms, tend to appear near each other in S&C log tables. This property is used by Find_siblings for output generation. All sets of three siblings above a certain *Prox* threshold (typically >0.1) are thus extracted and exported as input for the Assign_shapes module.

```
( f_i_R_k_a_n , t_r_z_k_a_n , s_i_R_g_0_l )
( f_i_R_k_a_n , t_r_z_k_a_n , b_l_c       )
( f_i_R_k_a_n , s_i_R_g_0_l , b_l_c       )
( t_r_z_k_a_n , s_i_R_g_0_l , b_l_c       )
    (...)
```

Figure 6. Exported tri-sets for m_019_g.

Of course, algorithms other than S&C could be used for segmentation and grouping of the phonetic data. Most of those known to us would however force us to split up the n-gram formation and the paradigm-formation in two more or less independent steps, which is why we settled on the S&C framework with its simultaneous and inter-dependent chunking and clustering. More discussion on the segmentation methodology is to follow in the final section.

3.3 Assign_shapes

The Assign_shapes algorithm is implemented in the programming language Prolog (e.g. Bratko 2000). In this language, propositional knowledge is particularly easy to formalize and to reason about, as exemplified by the **G** model below.

```
prop(color,blue,[e1,e10,e12]).
prop(color,green,[e2,e6,e9,e13]).
prop(color,red,[e4,e8]).
prop(color,yellow,[e5,e7]).
prop(color,purple,[e3]).
prop(color,brown,[e11]).

prop(shape,square,[e1,e4,e7,e11]).
prop(shape,circle,[e2,e6,e9,e10,e13]).
prop(shape,triangle,[e3,e5,e8,e12]).
```

Figure 7. Formal model of **G**.

The Assign_shape algorithm is perhaps best presented by an example. Consider a particular tri-set T3 of shape term candidates as delivered by Find_siblings, and transformed to the shape lexicon T3'.

```
T3 =
 (f_i_R_k_a_n, t_r_z_k_a_n, s_i_R_g_0_l)

T3' = (  square:   f_i_R_k_a_n ,
         circle:   s_i_R_g_0_l ,
         triangle: t_r_z_k_a_n )
```

We trace the program execution at a point where T3' is to be evaluated with respect to the session transcription, m_019_g, and a particular path *P′*.

```
P′ = [e1,e2,e3,e7,e8,e9,e4,
          e5,e6,e10,e11,e12,e13]
```

Consulting prop/3 (fig. 7), the Prolog engine infers that *P′* has the related shape sequence [*square*, *circle*, *triangle*, *square*, *triangle*, *circle*, *square*, *triangle*, *circle*, *circle*, *square*, *triangle*, *circle*], so T3' is evaluated by searching for a 13-section of the transcription m_019_g faithfully representing the shape sequence (that is, its T3' mapping). As it turns out, such a 13-section does exist, verifying T3' in this case.

In general, each tri-set delivered by Find_siblings is evaluated for each of its six permutations, and for each formally well-formed path (cf. 2.1). Each combination of path and tri-set for which a 13-section was found is then

passed on to Assign_colors for further evaluation.

3.4 Assign_colors

In this part of the Prolog script, a partly instantiated variable `Table` is declared.

```
  Table = [green:_,blue:_,red:_,
           yellow:_,purple:_,brown:_],
```

Using Prolog backtracking, a solution is sought in the form of a fully instantiated `Table` structure. A slightly simplified version of the central Prolog predicates is shown below (excluding some performance improving modifications).

```
eval(Tran,[T1,T2,T3]):-
  path(Path),
  perm(Shapes,[T1,T2,T3]),
  Table = [green:_,blue:_,red:_,
             yellow:_,purple:_,brown:_],
  traverse(Tran,Path,Shapes,[],Con),
  deduce_colors(Con,Table),
  write_result(Table,Path).

traverse( Tran_in, [E|Path],
          [Tr,Sq,Ci], ConIn,ConOut):-
  prop(color,Col,Colset),
  member(E,Colset),
  prop(shape,Shp,Shpset),
  member(E,Shpset), member(Shp:Shpname,
  [triangle:Tr,square:Sq,circle:Ci]),
  occur(Shpname,Tran_in,Tran_out,Con),
  traverse( Tran_out, Path, [Tr,Sq,Ci],
            [Col:Con|ConIn], ConOut
  ).
traverse(_,[],_,Con,Con).

deduce_colors([],_).
deduce_colors([Col:Txt|More],ColTable):-
  append(_,ColName,Txt),
  member(Col:ColName,ColTable),
  deduce_colors(More,ColTable).
```

Figure 8. Central Prolog predicates of Assign_colors

Two sessions contain small variations of the pronunciation for a specific color term, viz. m_029_g (*yellow*: `g_u_l`, `g_u`) and m_033_g (yellow: `g_u_l`, `g_u_l_0`). As these two are otherwise fit for InShape-2 analysis, we accommodate the phonetic variation replacing

```
  Table = [green:_,blue:_,red:_,
         yellow:_,purple:_,brown:_],
```

by

```
( Table=[green:_,blue:_,red:_,yellow:_,
         purple:_,brown:_], Extra=none
;
  Table=[green:_,blue:_,red:_,yellow:_,
         purple:_,brown:_,Extra:_]
),
```

in `eval/2` (fig. 8). Notice the logical disjunction (the connective `;/2`) ensuring that a proper one-to-one mapping (`Extra=none`), if any, is preferred over versions with an `Extra` color term. This way, a single unspecified lexical deviation can be accommodated in a controlled manner. Of course, more licenses could be issued by adding more `Extra`s to the `Table` list, at a price of extra processing load.

4 Results

The inference engine delivers satisfactory results for all sessions, however with some interesting twists. Before we go into the details, we present an example of an output from the Assign_colors module.

```
  m_017_g

  triangle : [t,r,z,k,a,n,d]
  square   : [f,i,R,k,a,n]
  circle   : [s,i,R,g,0,l]

  blue     : [b,l,c]
  brown    : [C,t,0,h,Q,j,C,f,C,d,
              0,n,b,l,c,s,0,R,g,
              0,l,h,A,d,u,e,n,b,r,o,n]
  green    : [g,r,Q,n]
  purple   : [C,X,
              0,n,X,i,g,E,n,
              h,A,d,u,e,n,l,e,l,a]
  red      : [n,r,x,D]
  yellow   : [g,u,l]

  Extra = none

  PATH : [e1,e2,e3,e7,e8,e9,e10,
                e11,e4,e5,e6,e12,e13]
```

Figure 9. Output from Assign_colors.

The `PATH` is correctly identified: informant 017 did name the thirteen objects in the order shown. Concerning the deduced vocabulary, several unusual phonetic forms are encountered. Perhaps most surprising are the very long terms for colors *brown* and *purple*. With a little bit of reflection, it is easy to see why the inference engine, with each of these colors occuring only once among the significant naming events, has too sparse data to determine their usual delimitation. As expected, the standard color names are identified by the *right* edge of the proposed strings (shown in

bold in fig. 9). Like English, Danish usually has adjectives in pre-nominal position.

Notice also that *red* translates to [nrxD] rather than the expected form [rxD] due to the fact that the latter on all its occurrences in m_017_g is preceded by [n]. More examples in the same vein can be studied in table 2, showing the variety of color terms picked for *yellow*.

Such non-standard delimitations are the fingerprints of a truly data-driven learning automat. Of course, several cosmetic operations could be applied post festum, arriving at tokens much more like the dictionary forms – for example by relating the deduced color-terms to the frequency distribution of the *n*-grams in the S&C log. We have chosen not to do so. Actually, we find the deduced terms quite beautiful as they are.

The results of all **G** analyses are summarized in table 2. Paths are explained using the symbols A=[**e1**,**e2**,**e3**], B=[**e4**,**e5**,**e6**], C=[**e7**,**e8**,**e9**], D=[**e10**,**e11**], and E=[**e12**,**e13**].

Session	**Path**	**Instantiation of `yellow`**	**Status**	**Diagnosis**
003	ABDEC	-	*OKneg*	*Anomaly*
005	ABCDE	-	*OKneg*	*Anomaly*
006	ABDEC	`[e,n,g,u,l]`	*OK*	-
007	ACDEB	`[n,r,x,D]`	*OK*	*ColBlind*
008	ACDEB	-	*OKneg*	*Anomaly*
009	ABDEC	`[g,u,l]`	*OK*	-
013	ACBDE	-	*OKneg*	*Violation*
014	ACBDE	`[g,u,l]`	*OK*	-
016	ACBDE	`[N,g,u,l]`	*OK*	-
017	ACDBE	`[g,u,l]`	*OK*	-
018	ACBDE	`[d,C,e,n, g,u,l]`	*OK*	-
019	ACDEB	`[e,n,r,x,D]`	*OK*	*ColBlind*
021	ACBDE	`[n,g,u,l]`	*OK*	-
027	ACBDE	`[e,n,g,u,l]`	*OK*	-
029	ADEBC	`[g,u,l] Extra:[g,u]`	*OK*	-
031	AC*1*D BC*2*E	-	*OKneg*	*I.con.viol.*
033	ABCDE	`[n,r,x,D]`	*OK*	*ColBlind*

Table 2. Learning results for InShape-2

In table 2, 'ColBlind' stands for an instance of color blindness (cf. 2.2); 'Violation' for a violation of a linguistic constraint (2.1); 'I.con.viol.' for an integrity constraint violation; 'Anomaly' for a factual description error.

As seen, all sessions were successfully analysed, in the sense that the same paths were identified by the inference engine (IE) and by a human listener (HUM). The sessions are marked as *OK* if the vocabulary and the path deduced by IE is the same as those reported by HUM, modulo color blindness, i.e. possibly with confusion of the terms for *red*/*yellow* and the related confusion of branches B and C. Sessions for which no well-formed paths could be found either by IE or by HUM, are marked as *OKneg* . These anomalous cases either contain factual description errors with late or no self-correction, or are in conflict with the well-formedness criteria of 2.1.

4.1 Linguistic constraints revisited

From a general linguistic point of view, the conditions of 2.1 are not very attractive. Which language does not have instances of synonymy or homonymy? How, then, could an L1 acquisition model afford to reject it?

We did a few test runs with manufactured sessions copied from real ones, but with certain vital elements changed, e.g. replacing each occurrence of the term for *blue* by the term for *triangle* (creating homonymy), and replacing every second occurrence of the term for *green* by a fresh term (synonymy). As it turned out, the InShape-2 system is actually fully capable of learning homonymy in this sense, and even synonymy with one synonymous term allowed for each extra uninstantiated element in the `Table` structure of the Assign_colors module – however at the cost of a heavy overhead in processing loads, especially in the case of synonymy.

Concerning the adjacency condition and the material inclusion condition, these are perhaps even more weakly motivated than the synonymy and homonymy constraints from a linguistic point of view. Again it is easy to modify the program to make it accept non-contiguous naming events, but the processing cost is high.

5. Discussion

InShape-2 is a simple-minded learning device, but nevertheless quite successful on its own terms. Based on unbracketed strings of phones representing a great variety of speech styles, the

system robustly derives a set of lexemes and semantic categories in a combined process of low-level data clustering and high-level semantic inference. However, still some refinement is needed. Only property terms were learned (in Danish mostly associated with adjectives and nouns) while locations and spatial relations were not (prepositions and adverbials), causing symptoms of color blindness. We are currently working on an enhanced learner with improved color vision, to be presented in the near future.

Greater challenges are waiting further ahead. Shifting from pre-digested phonetic symbols to uninterpreted acoustic data will soon force us to reconsider the whole regime of speech sound segmentation. It is by no means given that the phonematic level, of all possible levels, will provide the optimal domain for information extraction. In contrast, we expect that the most fertile segmental level will vary dynamically with the purpose of the learning session. "Meaningful units" thus cannot be identified a priori with phonemic or syllabic or prosodic elements, or any other independently defined domain, since the very meaning of *meaningful* depends crucially on the purpose and the success criteria of the task at hand. Simple examples of situated meaningful units are the terms extracted from the **G** sessions which did not always coincide with Danish dictionary items, simply because they were generated as handles for deductive reasoning under very specific conditions, rather than as speaker and purpose independent abstractions. We hence need to develop methods for sound analysis able to lock in on a particular domain of segmentation in a semantically informed feedback-loop. This means that the current phonetic transcription data must be abandoned and replaced by data derived from the sound signal directly (e.g. Henrichsen et al 2009).[6]

An acoustically based learning device would provide a number of interesting spin-offs, both of theoretical and practical nature. One of our immediate goals is to extend the InShape experiment to data from other languages. At CMOL, we have built a large collection of **G** session recordings for languages within the Indo-European family (German, Bulgarian, Hindi, ...) as well as typologically unrelated languages (Tamil, Xhosa, Khmer, ...). For most of these recordings by far, we have no phonetic transcription, so sound-driven learning is the natural approach towards genuine language-independency.

Even more difficult than the segmentation problem is however the model-theoretic challenge. The scene around the purple triangle must eventually be replaced by something of greater psychological relevance in order for us to approach a claim of cognitive realism. This does not mean, however, that the use of very simple models should be frowned upon. Even infants begin their linguistic career by developing primitive, highly personal sound units as names for a small number of concrete objects.

References

Ando, R. and L. Lee. 2003. Mostly-Unsupervised Statistical Segmentation of Japanese Kanji Sequences. *Natural Language Engineering*, 9(2).

Belkin, M. and P. Niyogi. 2004. Semi-supervised learning on Riemannian manifolds. *Machine Learning, Special Issue on Clustering*, 209–239.

Boersma, P. 2001. Praat, a system for doing phonetics by computer. *Glot. International* 5:9/10, 341-345.

Bratko, I. 2000. *Prolog Programming for Artificial Intelligence.* Third Edition, Addison-Wesley.

Church, K.W. and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1), 22 29

Cover, T.M.; J.A. Thomas. 1991. *Elements of Information Theory*. Wiley Series in Telecommunications, New York.

Grønnum, N. 2009. A Danish phonetically annotated spontaneous speech corpus (DanPASS). *Speech Comm.* 51, 594-603.

Henrichsen, P. J. 2004. Siblings and Cousins, statistical methods for spoken language analysis. *Acta Linguistica Hafniensia*, 36, 7-33.

Henrichsen, P. J. 2010. Den lilla trekant - learning Danish shape and color terms from scratch. *Linguistic theory and raw sound. Copenhagen Studies in Language*, 40. 27-44.

Henrichsen, P. J. and T. U. Christiansen. 2009. Fishing for meaningful units in connected speech; Proceedings of *ISAAR-2009.*

[6] An interesting investigation which we leave for others to explore, would be to compare the current S&C based chunking algorithm to alternative unsupervised, statistical methods using mutual information (e.g. Cover et al 1991), *t*-score (e.g. Church et al 1990), or newer frameworks as Ando (2003) and Belkin (2004).

ISSN 1736-6305 Vol. 11
http://hdl.handle.net/10062/16955