Internal Use Only

002

TR-IT-0119

Layered Language Understander: Generic Abductive Inference for Language Processing

Kevin Lenzo † John R. Josephson †* Christopher Bailey-Kellogg *

1995.6

Abstract

This report describes the design an current implementation of LLU (Layered Language Understander). LLU is application-building software (a "shell") for natural language processing using abduction, but designed to be independent of any particular language or level of processing (syllables, words, sentences, etc.). It is a specialization of a generic abduction mechanism with support for layered abduction and for handling temporally-bounded hypotheses.

ATR Interpreting Telecommunications Research Laboratories

† ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan.
* Laboratory for Artificial Intelligence Research, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio, USA

Contents

Introduction

Abduction, or inference to the best explanation, is a pervasive phenomenon. It appears in medical diagnosis, legal reasoning, perception, and hypothesis testing, just to mention a few areas; the first use of the word Abduction is generally ascribed to the American philosopher, Charles Saunders Peirce.

The general pattern of reasoning has been described as follows:

D is a collection of data (facts, observations, givens).
H explains D (would, if true, explain D).
No other hypothesis can explain D as well as H does.

Therefore, H is probably true. (Josephson & Josephson, 1994)

While this description covers a lot of territory, our work focuses on implementing a generic machine that is procedurally abductive, and is applicable to a variety of domains and knowledge sources. In particular, we are currently focusing on LLU, or Layered Language Understander, which is intended to be a generic, layered abduction framework for language processing.

LLU is implemented in C++, and is the sixth in a series of abduction machines (see Josephson & Josephson, ch. 9, 10). It is specifically targeted towards speech recognition and language understanding, and contains vestiges of the previous machines, as well as specific improvements. The first machines were designed for diagnosis, and LLU is a direct descendant of that legacy.

The current implementation runs on a Power Macintosh 8100/80AV, and is compatible with MetroWerks CodeWarrior 6.0. The core code is ANSI compatible, and so is largely portable to other systems.

Abduction

While the general form was given above, abduction itself bears closer examination. The term "best explanation," in particular, needs exploring. We take it that the "best" explanation is one that is parsimonious, confident, and consistent: parsimonious, in the sense that the data is explained with as few hypotheses as possible; confident, in that the component parts surpass competitors to a convincing degree' and consistent, in that the accept parts of a composite explanation to not contradict each other.

> Having evolved in a world of mixtures, humans have developed heuristic mechanisms capable of decomposing them. Because the conditions under which decomposition must be done are extremely variable, no single method is guaranteed to succeed. Therefore a number of heuristic criteria must be used to decide how to group the acoustic evidence. These criteria are allowed to combine their effects in a process very much like voting. No one factor will necessarily vote correctly, but if there are many of them, competing with or reinforcing one another, the right description of the input should generally emerge. (Bregman, 1990)

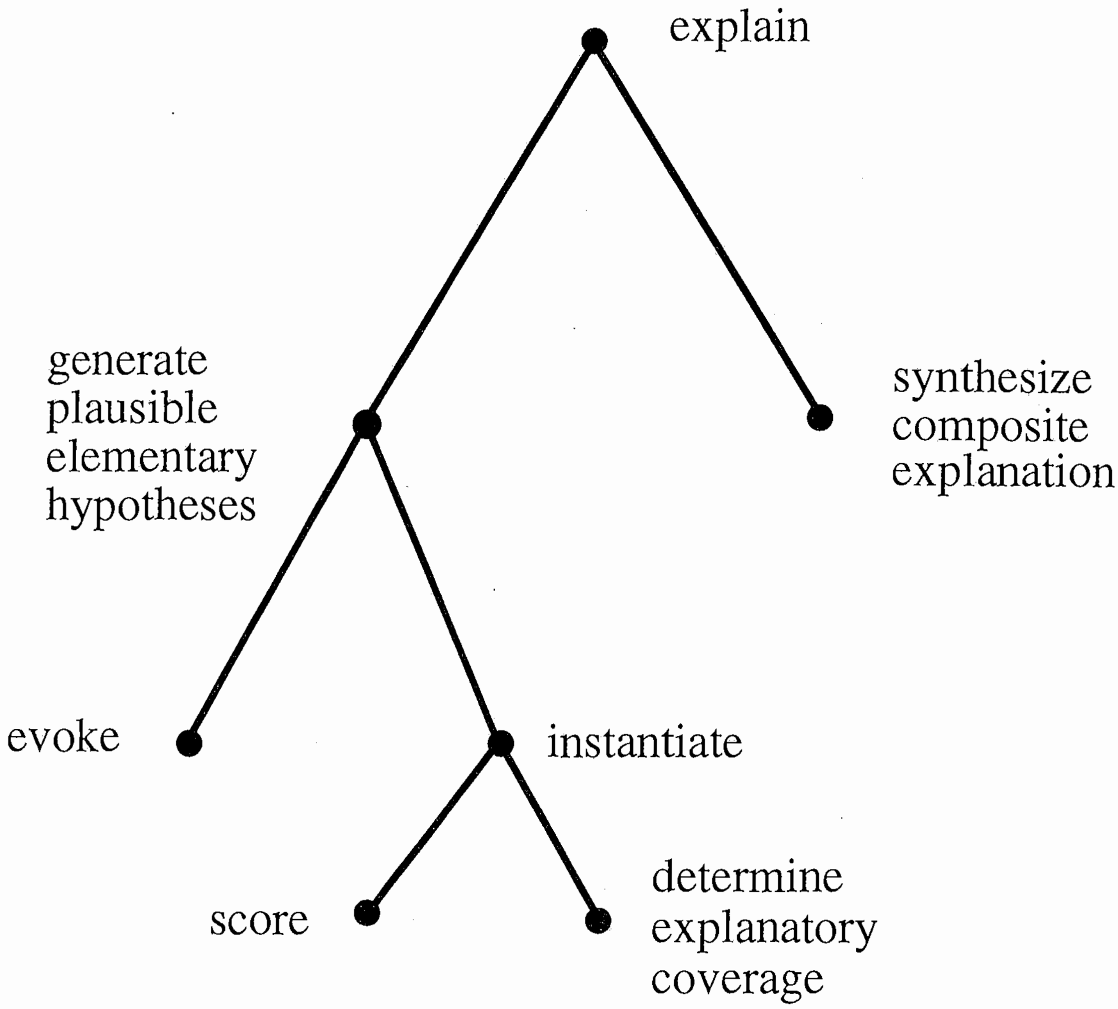
Hypotheses

In this formulation, a hypothesis is a volatile entity created during problem solving that instantiates if certain preconditions are met. It is updated during problem solving, and carries a Belief Status ("doxastic status"), as well as a confidence score based on a priori probabilities and the current state of problem solving.

During processing, hypotheses go through a cycle of evocation, instantiation, and composition.

Generating explanations by instantiation and composition

task-subtask breakdown



Evocation generally occurs bottom-up, as a hypothesis becomes "stimulated" for consideration; however, a hypothesis may also be evoked from above, as an expectation. In LLU, the bottom-up evocation is done using triggers, while the top-down evocation is performed using expectations.

Instantiation is the process by which a hypothesis gains an initial confidence score and determines how much of the data it can account for or cover. It combines a priori probabilities with how well the data "fits" the hypothesis. At instantiation time, no consideration has yet been given to rival hypotheses that may offer to explain or account for some or all of the data covered by the particular hypothesis; it is a logically parallel act.

Composition is the phase in which hypothesis interactions come into play, and, under good conditions, a coherent, "best" explanation emerges. LLU uses a least-commitment strategy to exploit essential hypotheses and make easy decisions first, and propagating the results of acceptance. Initial confidence scores, set at instantiation, become modified as the abducer tentatively accepts the essentials and clear-best explainers, and the interactions between hypotheses are leveraged to dynamically update the confidence scores. This is discussed in more detail later.

Layered Abduction

The loci of hypothesis formation an are named **agora** after the marketplace where the ancient Greeks gathered for dialog and debate. An agora is a place where hypotheses of a certain type gather and contend and where, under good conditions, a consensus hypothesis emerges.

In typical cases the emerging hypothesis will be a composite, coherent in itself, and with different subhypotheses accounting for different portions of the data. For example a syllable agora is the presumed location where syllable hypotheses are formed and accepted; each specific syllable hypothesis accounting for certain specific data from lower level agoras.

Hypotheses go through a cycle of evocation, instantiation, updating, and acceptance; results at one agora become data to be explained by another

Note that abduction, as a logical form, encompasses many hypothesis relation topologies that already exist, such as neural networks (NNs), rule-based resolution systems, hidden Markov models (HMMs), and directed acyclyc graphs. Each of these could be a particular instantiation of the truly generic abduction machine.

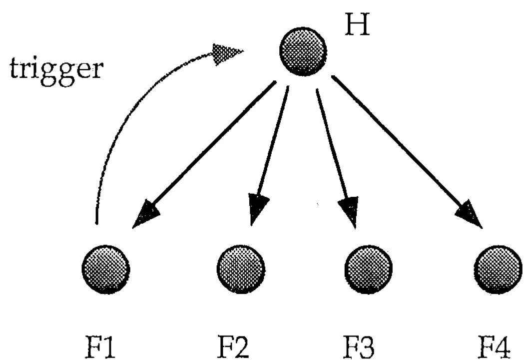


Figure 1. Neural Net-type topology

In the neural-net type topology (see Figure 1), the abduction machine can implement a feed-forward net by constraining the hypothesis relationships to be those of the appropriate network; i.e., findings (show as F1 through F4) stimulate covering hypotheses ("nodes"), and the covering nodes may have damping relations toward each other. However, transition probabilities are not considered.

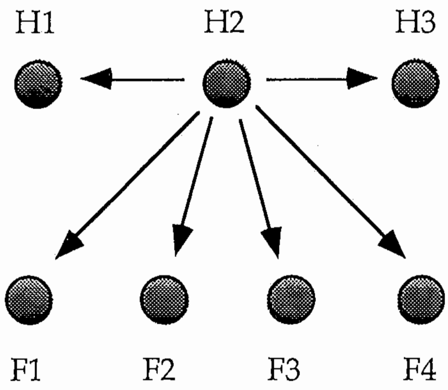


Figure 2. HMM-type topology

In another type of topology, an HMM-type construct or a grammar-based set of relations can be constructed. The hypothesis under examination (here, H2) depends on the transition probabilities of temporally adjacent hypotheses. If the link between H2 and H3 is replaced by a trigger, the relationship becomes more clear. Of course, the statistics required must be initialized and updated based on learning to satisfy the requirements of an HMM to actually be an HMM.

In the current system, hypotheses exist in time. The beginning and end times of a hypothesis reflect their uncertainty; the times are represented in interval representations (see Figure 3).

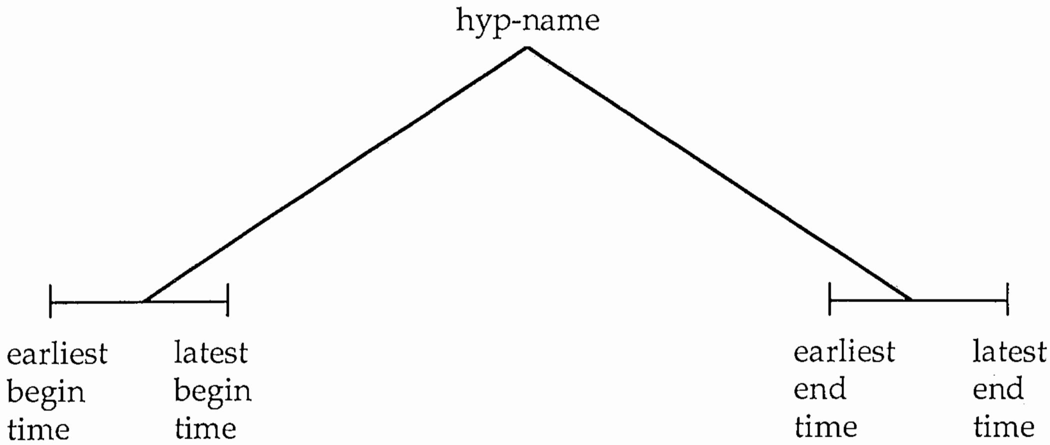


Figure 3. Temporal Hypotheses in LLU

Often the regions covered by a piece of data to be explained may be accounted for by overlapping hypotheses. Consider for instance "six stockings". If the s-fix frication region at the end of "six" and the s-frication at the beginning of "stockings" is forced to one side or the other, the parse will be wrong. "sick stockings" or "six talkings". The <u>duration</u> of the frication may not be longer than it would be for a single word, especially in rapid or casual speech. This forces an overlapping representation, where more than one hypothesis can account for the same data. Overlapping hypotheses were possible in the abduction machine, but knowledge is not present from higher-order parsing such as syntax or semantics to disambiguate these sorts of errors.

Rather than specifying absolute time intervals, a user who wants to explicitly specify the knowledge should be able to use some notation for a partial ordering. It would be highly desirable to characterize partially-ordered sets with constraints rather than actual time values in some cases, considering the amount of temporal variability under various prosodic conditions. Otherwise, the fuzzy boundaries used for temporal specification spread out so far as to become almost meaningless.

Hypotheses can be mutually incompatible, or they can coexist and overlap in time. For instance, in the feature-based machine, a syllable explains a set of overlapping, articulatorially-based features around a region of time. In the phonemoid-based machine, however, one phoneme is posited by the HMM front-end for each time slice, and thus they do not overlap.

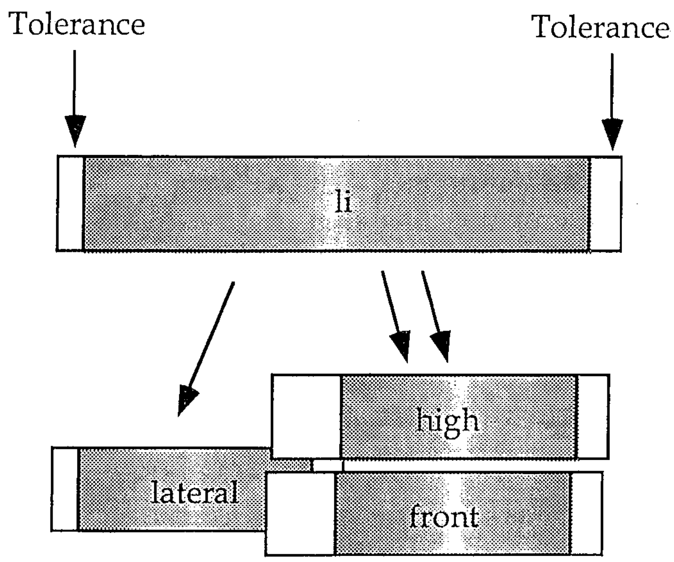


Figure 4. Overlapping, non-contradictory hypotheses

For the edge-driven, 2-layer abduction machine, a trigger from the layer below instantiates a hypothesis at the next level, and these triggers are positioned along the left edge of the higher-level hypothesis. An efficient, cost-based abductive parser has also been discussed in (Den, 1994).

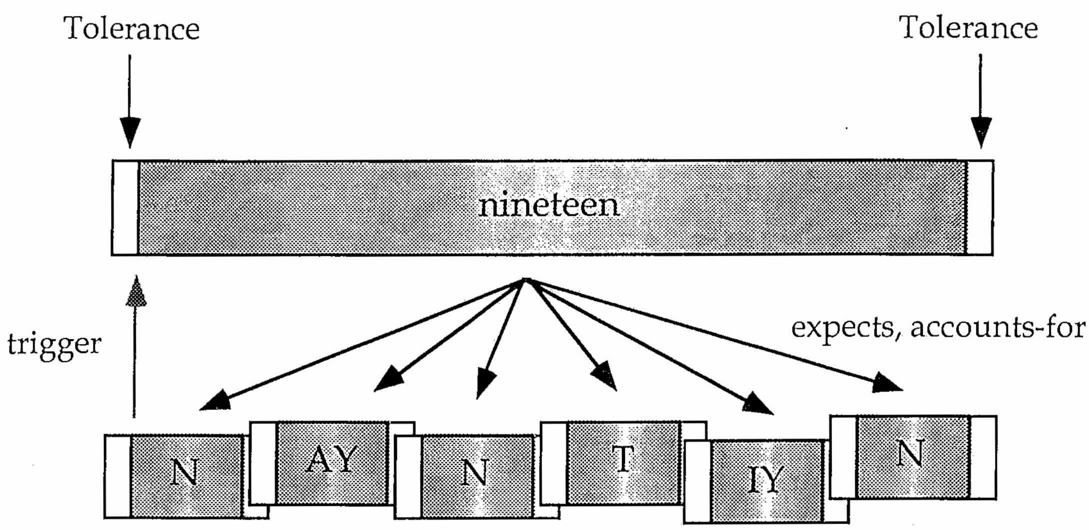


Figure 5. Some relationships between hypotheses in a 2-level machine

In our work, an "abducer" is a software agent that manages problem solving within a defined hypothesis space, known as an agora. An agora may be considered a "marketplace of ideas," where hypotheses gather and contend for the data to be explained. The collection of abducers are, in turn, managed by an agent called an "abducer manager."

Conflation

Among the other duties on the abducers, one is hypothesis conflation. When two hypotheses of the same type exist, and the boundaries of their temporal coverage completely intersect, hypothesis 'conflation' occurs, with one of two possible results: 1) one hypothesis adjusted and the other is eliminated, or 2) one hypothesis is adjusted and the other is left untouched.

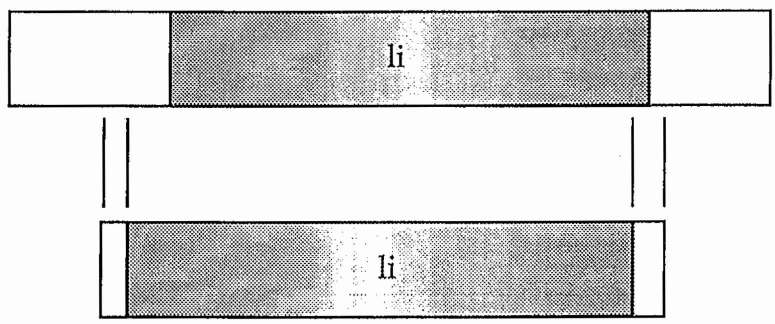


Figure 6. Overlapping boundaries of temporal coverage for two hyps of the same type

The EFLI strategy

The acceptance agenda follows a strategy we call EFLI, or Essentials First Leveraging Incompatibilities. Hypotheses that offer unique explanations are considered Essentials. Essentials are added to the composite as they appear, and the consequences of their acceptance are propagated through the system.

Hypotheses are added to the composite in order of certainty, where certainty is determined by

- the confidence rating of the hypothesis
- how well the confidence rating surpasses competing hyps
- the distribution of alternate explainers

See (Josephson, Smetters, et al., 1989) for a discussion of the criteria.

The threshold for acceptance is gradually lowered, and hypotheses are added to the composite explanation; their consequences are propagated through the system. Consistent with least commitment and island-driven strategies, hard decisions are delayed until further evidence can be evaluated, and may not be hard decisions if a conclusion is forced by confident hypotheses. As the system moves down the confidence gradient, the composite explanation covers more but may reduce in overall confidence.

The EFLI strategy is used for control of hypothesis assembly (Josephson & Josephson, 1994, ch. 9). EFLI may be briefly described as:

- Find the data with the lowest ambiguity. This may be either a datum with only a unique possible explainer, an "essential hypothesis," or with one explanation being much better than all others.

- Accept the best explanation for each low-ambiguity data point. That is, make a local, confident abduction.

- Propagate the consequences of acceptance by using known hypothesis relationships. This includes the rejection of incompatible hypotheses which compete with accepted hypotheses, and rescoring those that have expectations towards the accepted hypothesis.

- If necessary, lower the standards for acceptance and continue. (Lowering the standard for acceptance means accepting hypotheses that are best explanations, but surpass their competitors by a degrading margin.)

Please refer to ATR ITL Tech Report TR-IT-0075, "Generic Software for Language Understanding: a design based on layered abduction," for a more complete discussion the LLU design.

Automatic Generation of Abduction Machines from Pronunciation Dictionaries

In our work, a perl script was used to automatically generate 2- and 3-level abduction machines from pronunciation dictionaries. This allowed for testing under a variety of conditions, and also allowed for the rapid creation of a very large recognition machine. A list of words, and a pronunciation dictionary, are given to the script, and it generates all of the appropriate hypothesis types and triggers between them.

To create the abduction machine automatically, the script first finds all pronunciations for the target words in the dictionary. Then, it creates hypothesis types for each of them, including information about which constituents the expect and account for. Thereafter, a hypothesis type is made for each type of constituent, in a recursive descent fashion, and the relationships of triggering are described at the appropriate level as the levels descend. References which are not in the dictionary but may appear in the input can be included in a separate file of forced references; this was done for the PAU and brth (pause and breath, respectively) tokens, which did not appear in the dictionary but did in the input tokens.

Unfortunately, there was no direct automatic generation of abduction machines based upon the Converter/Distributor (C/D) Model (Fujimura, 1994), which was an initial goal of the work. The reason for this is that, as the project progressed, the generic abduction processing gained in precedence in comparison to the signal processing and stochastic database techniques that would be useful for making feature detectors for the features posited by the C/D model.

Results

More experiments need to be done in order to capture precise statistics; however, we report here some preliminary results using the Boston University Radio Data, speaker f2b. The corpus contains 116 utterances.

- 42 phonemoid types (including PAU and brth)
- 2315 unique syllables
- 3283 pronunciation dictionary entries

Utterances can be quite long, and have been broken into sentences. Each test was done on a sentence-length unit.

short dictionary: uses only words in the utterance, 12-20 words
moderate performance in initial tests. approaching 85%.

confounding words: uses all the words in the paragraph; up to 4x as many words as in the target
reasonable performance, with some degradation. ~75%

full dictionary: uses 3283 words
just beginning experiments now, but experiencing problems of scale. Abduction machine is refusing to commit due to the entertainment of so many hypotheses.

Conclusion

Abduction is a generic inference mechanism that can be exploited explicitly for speech recognition. While the set-covering aspects of the system seem to be functioning effectively, there is a word-subword problem if the subword confirms first and marks the data as explained. The machine generates about 200 hypotheses, excluding the input hypotheses, for the short dictionary and confounding words case, while it generates several thousand for the complete dictionary. Some more aggressive pruning may be in order; see (Den, 1994) for a cost-based approach to search control.

While almost all recognition devices are logically abductive, there are benefits to be obtained by generic, procedurally abductive systems -- these include:

- hybrid integration of knowledge sources
- multiple levels of abstraction
- explicit coding and learning in the same mechanism
- explanation-based reasoning
- multiple partial explanations
- consistent, generic mechanism
- cognitively plausible

But, pre-processing and the sensory periphery offer important challenges. Without good pre-processing, the system will fail -- as would any. Models of hypothesis-hypothesis interaction provide important knowledge for the system.

In our opinion, future research should be specifically aimed and prosodic control, using the assumptions of the C/D Model. Coherent integration of stress and focus are goals we were unable to achieve in the short span of one year, but the mechanism allows for it.

Future possibilities also include

- Auditory Scene Analysis processing project with Guy Brown
- Rhythmogram and prosodic incorporation with Neil Todd
- incorporation of other knowledge sources (syntax, semantics, etc.)

Acknowledgments
This work was supported mostly by ATR Interpreting Telecommunications Research Laboratories.

Bibliography

Bregman, A. (1990). *Auditory Scene Analysis*, A Bradford Book, MIT Press, Boston, Massachusetts.

Den, Y. (1994). Generalized Chart Algorithm: An Efficient Procedure for Cost-based Abduction. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico.

Fujimura, O. (1994). Syllable Timing Computation in the C/D Model. In *Proceedings of the 3rd International Conference on Spoken Language Processing*, Yokohama Japan.

Josephson, J. R., & Josephson, S. G. (Eds.). (1994). *Abductive Inference: Computation, Philosophy, Technology*. New York: Cambridge University press.

Josephson, J. R., Smetters, D., Fox, R., Oblinger, D., Welch, A., & Northrup, G. (1989). *The Integrated Generic Task Toolset, Fafner Release 1.0.* Technical Report, The Ohio State University, Laboratory for Artificial Intelligence Research.

Appendix A: Specifying Hyptypes

Hyptype specifications are put in a single file with extension ".ht"
.

```
<.ht file entry> ::=
  <hyptype name> [ (<param defs>) ]
  <agora name>
  <total considered>, <total true>, <total false>
  <triggers, etc., in any order (order only matters for inheritance>
  -

<param def> ::= <intparam def>

<intparam def> ::= d:<default initial value (an integer)>

<refined from> ::= r:<super hyptype>
<trigger>      ::= t:<triggered hyptype> [ (<derived params>) ],
<temporal offset>
<expectation>  ::= e:<expected hyptype> [ (<derived params>) ],
<temporal offset>,
                           h&e, ~h&e, h&~e, ~h&~e
<implication>  ::= i:<implied hyptype> [ (<derived params>) ],
<temporal offset>,
                           <expectation decision (y or n)>,
                           <resulting hypothesis decision (y or n)>
<accounts for> ::= a:<accounts for hyptype> [ (<derived params>) ],
<temporal offset>

<derived param> ::= d:<intparam init>
  _    (use default value)

<intparaminit> ::=
  <use this param # in the original> /
  <add this offset to it to yield derived param>

<temporal offset> ::=
  '[' '[' <min-start> <max-start> ']' '[' <min-end> <max-end> ']' ']'
```

Appendix B: Example HypTypes files

<these are only an excerpts!>

---

for 3-level phoneme-based machine:

```
M
phon
100,1,99
t:M.AE.S_s,[[0/0 1/0] [2/3 3/3]]
t:M.AX.N.T_s,[[0/0 1/0] [2/3 3/3]]
t:M.EH.N.D_s,[[0/0 1/0] [2/3 3/3]]
t:M.EH.N_s,[[0/0 1/0] [2/2 3/2]]
t:M.EY.D_s,[[0/0 1/0] [2/2 3/2]]
-
M.AE.S_s
syll
100,1,99
a:M,[[0/0 1/0] [2/0 3/0]]
e:M,[[0/0 1/0] [2/0 3/0]]
a:AE,[[0/1 1/1] [2/1 3/1]]
e:AE,[[0/1 1/1] [2/1 3/1]]
a:S,[[0/2 1/2] [2/2 3/2]]
e:S,[[0/2 1/2] [2/2 3/2]]
t:Massachusetts_w,[[0/0 0/0] [2/8 3/10]]
-
Massachusetts_w
word
100,1,99
a:M.AE.S_s,[[0/0 1/0] [1/2 2/2]]
e:M.AE.S_s,[[0/0 1/0] [1/2 2/2]]
a:AX_s,[[0/3 1/3] [1/3 2/3]]
e:AX_s,[[0/3 1/3] [1/3 2/3]]
a:CH.UW_s,[[0/4 1/4] [1/5 2/5]]
e:CH.UW_s,[[0/4 1/4] [1/5 2/5]]
i:CH.UW_s,[[0/4 1/4] [1/5 2/5]],n,n
a:S.IH.T.S_s,[[0/6 1/6] [1/9 2/9]]
e:S.IH.T.S_s,[[0/6 1/6] [1/9 2/9]]
-
```

---

for 3-level feature-based machine:

```
fricative
feature
100,50,50
t:dh_ae_ts,[[0/0.0 1/0.0] [2/100.0 3/650.0]]
t:dh_ae_ts,[[0/-650.0 1/-300.0] [2/0.0 3/0.0]]
t:fUl,[[0/0.0 1/0.0] [2/60.0 3/300.0]]
-
fUl
syllable
100,50,50
e:fricative,[[0/0.0 1/0.0] [0/150.0 1/210.0]],10,5,2,2
e:labial,[[0/0.0 1/0.0] [0/150.0 1/210.0]],10,5,2,2
e:lateral,[[2/-200.0 3/-50.0] [2/0.0 3/0.0]],10,5,2,2
```

```
a:fricative,[[0/0.0 1/0.0] [0/150.0 1/210.0]]
a:labial,[[0/0.0 1/0.0] [0/150.0 1/210.0]]
a:lateral,[[2/-200.0 3/-50.0] [2/0.0 3/0.0]]
t:wonderful,[[0/-600.0 1/-400.0] [2/0.0 3/0.0]]
-
wonderful
word
100,50,50
e:wUn,[[0/0.0 1/0.0] [0/160.0 1/300.0]],10,5,2,2
e:dR,[[0/200.0 1/400.0] [2/-10.0 3/-30.0]],10,5,2,2
e:fUl,[[2/-300.0 3/-200.0] [2/0.0 3/0.0]],10,5,2,2
a:wUn,[[0/0.0 1/0.0] [0/160.0 1/300.0]]
a:dR,[[0/200.0 1/400.0] [2/-10.0 3/-30.0]]
a:fUl,[[2/-300.0 3/-200.0] [2/0.0 3/0.0]]
-
```

Appendix C: Example Input Files

for phoneme-based machine:

```
PAU,[[1 1] [1 1]]
AH,[[2 2] [2 2]]
N,[[3 3] [3 3]]
AY,[[4 4] [4 4]]
N,[[5 5] [5 5]]
T,[[6 6] [6 6]]
IY,[[7 7] [7 7]]
N,[[8 8] [8 8]]
EY,[[9 9] [9 9]]
T,[[10 10] [10 10]]
IY,[[11 11] [11 11]]
N,[[12 12] [12 12]]
S,[[13 13] [13 13]]
T,[[14 14] [14 14]]
EY,[[15 15] [15 15]]
T,[[16 16] [16 16]]
K,[[17 17] [17 17]]
AA,[[18 18] [18 18]]
N,[[19 19] [19 19]]
S,[[20 20] [20 20]]
T,[[21 21] [21 21]]
AX,[[22 22] [22 22]]
T,[[23 23] [23 23]]
UW,[[24 24] [24 24]]
SH,[[25 25] [25 25]]
EN,[[26 26] [26 26]]
EL,[[27 27] [27 27]]
AX,[[28 28] [28 28]]
M,[[29 29] [29 29]]
EH,[[30 30] [30 30]]
N,[[31 31] [31 31]]
M,[[32 32] [32 32]]
AX,[[33 33] [33 33]]
N,[[34 34] [34 34]]
T,[[35 35] [35 35]]
```

"A 1918 state constitutional amendment" BU f2b

---

for feature-based machine:

```
voiced_cons,[[20 30] [70 90]]
interdental,[[20 30] [70 90]]
fricative,[[20 30] [70 90]]
low,[[90 100] [240 250]]
stop,[[250 260] [300 310]]
apical,[[250 260] [300 310]]
fricative,[[305 315] [440 455]]
apical,[[305 315] [440 455]]
rhoticized,[[470 490] [530 540]]
high,[[510 520] [560 570]]
front,[[510 520] [560 570]]
```

```
reduced,[[560 570][625 635]]
lateral,[[630 640] [705 715]]
high,[[700 720] [840 850]]
front,[[700 720] [840 850]]
glide,[[850 860] [940 950]]
labiovelar,[[850 860] [940 950]]
reduced,[[940 960] [990 1010]]
nasal,[[1000 1015] [1100 1110]]
apical,[[1000 1015] [1100 1105]]
stop,[[1110 1115] [1130 1140]]
apical,[[1110 1115] [1130 1140]]
voiced_cons,[[1110 1115] [1130 1140]]
rhoticized,[[1130 1150] [1220 1230]]
fricative,[[1220 1230] [1340 1350]]
labial,[[1220 1230] [1340 1350]]
lateral,[[1370 1380] [1490 1500]]
```

"That's really wonderful."

Appendix D: Example Output (Condensed)
for feature-based machine

```
Abducer syllable: report continued
Accepted hypotheses:

1. dh_ae_ts [[-40 -4][30 31]] [.8 .9 .9]
-    explains 4 5 6 6
2. ri [[47 49][53 68]] [.9 1 1]
-    explains 10 11
3. lone_vowel_syllable [[53 61][57 70]] [1 1 1]
-    explains 12
4. li [[26 99][84 110]] [.9 .9 .9]
-    explains 13 14 15
5. wUn [[85 86][94 125]] [.9 .9 .9]
-    explains 16 176.
6. dR [[104 127][85 202]] [.9 .9 .9]
-    explains 21 22 23 24
7. fUl [[122 123][140 165]] [.8 .8 .8]
-    explains 27

Other possible hypotheses:

8. dh_ae_ts [[2 3][32 68]] [.8 .8 .8]
-    could explain 4 7 8
9. dh_ae_ts [[-63 -27][7 9]] [.8 .8 .9]
-    could explain 3 4
10. wUn [[-5 26][30 31]] [.8 .8 .9]
-    could explain 6
11. lone_vowel_syllable [[-5 30][26 31]] [.8 .8 .8]
12. dh_ae_ts [[12 67][42 133]] [.8 .8 .8]
-    could explain 7 8 20 22
13. fUl [[12 67][38 98]] [.8 .8 .8]
-    could explain 7 13
14. dh_ae_ts [[-345 15][44 455]] [.8 .9 .9]
-    could explain 4 7 8
15. wUn [[5 315][44 455]] [.8 .8 .9]
-    could explain 8
16. dR [[39 49][53 54]] [.8 .8 .8]
-    could explain 9


Abducer word: report

Data to be explained:

1. dh_ae_ts [[-40 -4][30 31]] [.8 .9 .9], explained
2. ri [[47 49][53 68]] [.9 1 1], explained
3. lone_vowel_syllable [[53 61][57 70]] [1 1 1],
     explained
4. li [[26 99][84 110]] [.9 .9 .9], explained
5. wUn [[85 86][94 125]] [.9 .9 .9], explained
6. dR [[104 127][85 202]] [.9 .9 .9], explained
7. fUl [[122 123][140 165]] [.8 .8 .8], explained

Accepted hypotheses:
```

```
1. thats [[-40 -4][30 31]] [1 1 1]
-    explains 1
2. really [[46 50][85 109]] [1 1 1]
-    explains 2 3 4
3. wonderful [[84 87][86 205]] [.9 1 1]
-    explains 5 6 7

Other possible hypotheses:

none
```

Once the result is found, the Apple Speech Manager is used to speak the utterance "That's Really Wonderful."