

CVA/Computational Team

- Fall 2001:
 - CSE 676: preliminary coding of new words
 - Marc Broklawski & Scott Napieralski:
 - Analysis of Ehrlich's noun-definition algorithm
 - English version of the N algorithm

N alg (revised; Eng. Version)

Report (if known):

1. Class membership:

- a) In general, report most specific class of any class hierarchies
 - Exception: always report basic-level class membership
 - Possible exception: always report animal/plant...
maybe report: abstract/physical object
- b) Else report names of indivs that are Ns

(continued...)

N algorithm (in English, revised), cont'd.

(Report:)

2. Properties

- If there are no known class memberships,
then report props of indiv Ns
as “possible props” of Ns

3. Structural, else possible structural, info

- I.e., possessive info
- Should be: part/whole info

(continued...)

N algorithm (in English, revised), cont'd

(Report:)

4. Functions, else possible functions, of Ns
 - I.e., things that Ns are used for
5. Acts, else possible acts, that Ns perform
6.
 - Agents that perform acts on Ns
 - & the acts they perform
7. Ownership info
8. Synonyms

N alg: Comments

- Need to distinguish between:
 - Gathering information
 - Organizing & reporting information
- 2 different kinds of definitions:
 - Human-like:
 - report everything, even redundant info
 - Lexicographic:
 - Constraints on how much info to report

CSE 663 Accomplishments (Spring 2002)

- Marc Broklawski: N algorithm; general resource
- Scott Napieralski: `proximity`
- Matt Sweeney: `estuary`
- Rajeev Sood: V algorithm
- Justin DeVecchio: `proliferate`
- Chris Garver: `taciturn`

Outline:

1. What was your role in the project?
2. What were your accomplishments?
3. What are the immediate next steps?
4. What is the longer-term future work?

Marc Broklawski

- Role: to make N algorithm more efficient;
general resource for other students
- Accomplishments:
 - made N algorithm more efficient
 - case-frame dictionary for N algorithm
 - recreated demos: *cat*, *stender*, *brachet*, *tomato*
 - instructions for how to run system

Marc (continued)

- Next steps:
 - global vars → local vars
 - ‘hackney’ demo (redo from scratch)
 - Add demos to official demo list
 - Solve innet/outnet problem

Marc (continued)

- Future work:
 - SNePSwD? Better: Fran's system
 - explanation/debugging facility
 - flag for 2 different kinds of definitions:
 - complete/human-like
 - lexicographic

Scott Napieralski: `proximity`

1. “Galileo recently flew just 120 miles (200 kilometers) above Europa, a **proximity** which allowed the spacecraft to take the most detailed pictures ever of the mysterious satellite.”
2. “Archeologists who discovered and excavated the small cave in 1983 had assumed that the brittle black material on the skulls was asphalt because of its color and the site's **proximity** to the largest asphalt deposits in Israel.”
3. “Because of its **proximity**, Earth's only natural satellite—the Moon—became the first celestial body to be visited by humans (in 1969).”

Scott (continued)

- Accomplishments:
 - Re-represented 3 passages for Ehrlich's N alg
 - Ran alg to look for improvements
 - Discovered that alg doesn't find everything it could

Scott (continued)

- Next step:
 - Modify N alg to report more info
- Future work:
 - Look for more improvements in N alg
 - Look for common traits in report info
 - Assume N is what X, Y, & Z are;
X, Y, & Z have props P, Q, R;
Therefore, Ns have P, Q, R
 - Need more background knowledge
 - Abstract Ns:
 - No structure, actions, etc., which are heart of N alg
 - Therefore, need other categories for abstract Ns

Matt Sweeney: `estuary'

“An **estuary** is a coastal area where fresh water from rivers and streams mixes with salt water from the ocean. Many bays, sounds, and lagoons are **estuaries**.

Estuaries provide safe spawning grounds and nurseries and are critical for fish, birds, and other wildlife.”

Matt (continued)

- Role:
 - Get “practical” (rather than “good”) representation to work
 - Change code so that code and input would work well together

Matt (continued)

- Accomplishments:
 - Coded `estuary' passage & background knowledge into SNePSUL
 - Modified N algorithm to get
“necessary-condition definition”
 - Discovered necessity of tricking forward inference to work

Matt (continued)

- Next steps:
 - Create “sufficient-condition definition” function
 - Test with SNePS 2.6
 - Develop tracing algorithm
 - For explanation facility
 - Good Lisp exercise
 - Modify N algorithm to report more detail
 - E.g., about agent-act-object case frame
 - Possibly solvable by NLG facility
 - Need def to include info about bays, sounds, lagoons

Matt (continued)

- Future work:
 - Optimal representation, maybe in SNePS 3
 - Continue work with tracing algorithm

Rajeev Sood: Verb algorithm

- Role:
 - Understand V algorithm
 - Translate into English
 - For use by education team
 - Create case-frame dictionary for V algorithm

Rajeev (continued)

- Accomplishments:
 - Met with Justin
 - Analyzed V algorithm
 - Improved V algorithm a bit
 - Created English translation
 - Created case-frame dictionary for V algorithm

English translation of V alg:

1. Categorize the subject and report it.
2. Determine the form of the verb:
 - a. If bitransitive, categorize the direct and indirect objects and report them.
 - b. If transitive, categorize the direct object and report it.
 - c. If reflexive, report that the subject performs that act upon itself.
 - d. If intransitive, go to the next step.
3. Find anything that the verb occurring enables (cause function) and report it.
4. Find anything that occurs that enables the verb to occur (effect function) and report it.

Revised Eng. Transl'n of V alg:

1. Categorize the subject and report it.
2. If it exists, categorize the direct object and report it.
3. If it exists, categorize the indirect object and report it.
4. Find anything that the verb occurring enables (cause function) and report it.
5. Find anything that occurs that enables the verb to occur (effect function) and report it.

Rajeev (continued)

- Next steps:
 - Change setq's to let's
 - Tracing function for V alg
 - Need more info returned
 - How to determine V category from structure

Rajeev (continued)

- Future work:
 - V alg needs **LOTS** of elaboration!!!
 - More than just cause/effect info needed
 - E.g., ‘throw’ = ball leaves hand, goes to other hand, etc.
 - Parser/generator

Justin DelVecchio: 'proliferate'

“A decade ago research on lab animals revealed that stem cells taken from animal embryos are astoundingly versatile. They grow in the lab, **proliferate** like rabbits and turn into specialized cells such as neurons.”

Justin (continued)

- Role
 - A verb!
 - 4 goals:
 - Redo Marc's representation for Ehrlich case frames
 - Work with Rajeev Sood to analyze V algorithm
 - Background knowledge
 - Find more passages

Justin (continued)

- Accomplishments:
 - Studied what Marc had done
 - Met with Rajeev to understand V algorithm
 - Created 'throw' demo to show how V algorithm works
 - Re-represented 'proliferate' passage to match V alg
 - Coded background knowledge
 - Esp. actions common to stem cells & rabbits
 - Tested V algorithm on passage + background kn.
 - Analysis of changes in V algorithm

Justin (continued)

- Next steps:
 - More work on background knowledge:
 - More actions & characteristics of stem cells & rabbits
 - Changes to V algorithm:
 - Better output (maybe in English)
 - Delete labeling of Vs (transitive, etc.);
replace w/ node structure
 - Differentiate results for intransitive & transitive Vs like
'throw'

Justin (continued)

- Future work:
 - Need more Vs!
 - More passages with ‘proliferate’
 - More/better background knowledge
 - Parser/generator!

Chris Garver: `taciturn`

“Unlike his brothers, who were noisy, outgoing, and very talkative, Fred was quite **taciturn.**”

Chris (continued)

- Role:
 - Adjective!
- Accomplishments:
 - Re-represented passage in SNePSUL
 - Represented background knowledge in SNePSUL
 - Looked at other passages & their background knowledge
 - Outline of Adj algorithm
 - Some new case frames

Outline for Adj(/Adv?) Alg:

1. Synonyms & antonyms
 - Problem: How to determine!
2. Class membership
 - E.g., ‘hot’ isa temperature; ‘quick’ isa speed
3. Lists of objs known to be describable & not describable w/ the Adj
4. Lists of actions that can/can’t be performed while possessing the Adj
 - someone is “not communicative” & “talks very little”
 - This example raises issue of “negative” adjs.

Chris (continued)

- Next steps:
 - Fix bugs in implementation
 - Esp. SNeBR problem: unexpected contradiction
 - Need tracing feature!
 - Need “positive” passages

Chris (continued)

- Future work:
 - Analysis of adjs:
 - Categories/possible contexts/domains of adjs
 - Kinds of Ns that certain adjs describe
 - What does adj refer to: “it was heated”
 - Implement Adj algorithm
 - Possibly combine with Adv algorithm
 - Rep & test more passages

Summer Plans:

- Scott: Re-implement N alg
 - with explanation/debugging facility
 - with good documentation!
- Justin: Re-implement V alg
 - Use Levin's theory of V categories?
 - Work on English translation
- Rapaport & Kibby:
 - New grant proposal
 - Orlando conference; Scotland conference(?)
 - Literature-review paper?

Future Work:

- Need parser/generator!
- Implement Fran's automated belief revision system!
- Implement all examples from the CVA literature
- Use OpenCYC to supplement background knowledge?