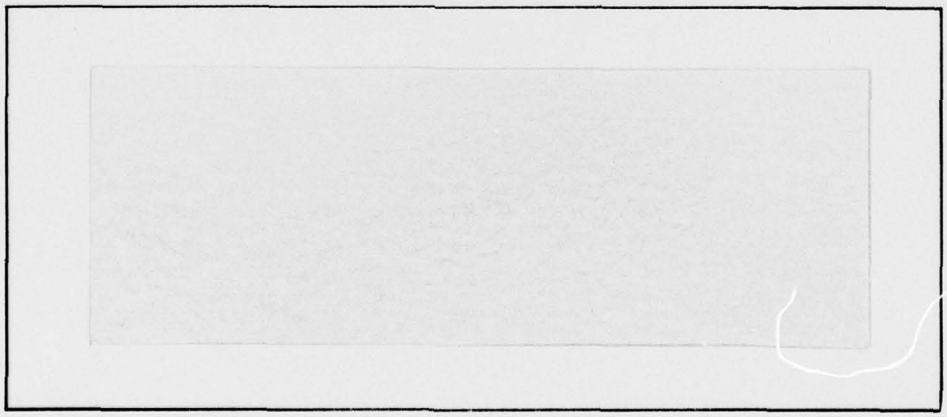
AD-A035 874 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/7
RESEARCH AT YALE IN NATURAL LANGUAGE PROCESSING.(U)
1976 R C SCHANK N00014-75-C-1111
UNCLASSIFIED RR-84 NL

1 OF 1
ADA035874

END

DATE
FILMED
3-77

(12)
B.S.



YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

The research described here was done at the Yale Artificial Intelligence Project in 1976. This project is directed by Professor Roger C. Schank, and is funded in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under Contract N00014-75-C-1111.

RESEARCH AT YALE IN NATURAL LANGUAGE PROCESSING

1976

Research Report #84

Roger C. Schank

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: #84 (14) RR-84
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*: (6) Research At Yale In Natural Language Processing.
5. TYPE OF REPORT & PERIOD COVERED: (9) Technical rept.
6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Roger C. Schank
8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-1111

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Yale University
Department of Computer Science
10 Hillhouse Ave., New Haven, Conn. 06520

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
(12) 32 p.

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

12. REPORT DATE: (11) 1976
13. NUMBER OF PAGES: 29

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
Office of Naval Research
Information Systems Program
Arlington, Virginia 22217

15. SECURITY CLASS. *(of this report)*: Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Parsing
Natural Language Understanding
Artificial Intelligence
Computational Linguistics
Semantics
Conceptual Dependency

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

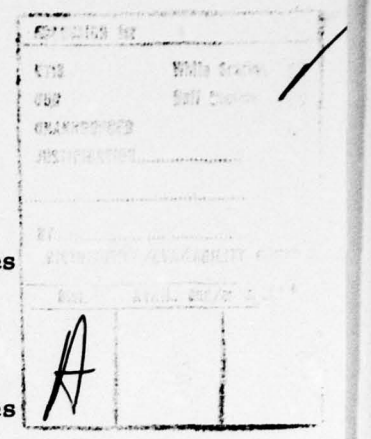
This report describes the state of the computer programs at Yale that do automatic natural language processing as of the end of 1976. The theory behind the programs shown here as well as descriptions of how those programs function, has been described elsewhere. This report is ~~intended as a~~ summary of the capabilities of ~~our~~ computer programs at the present time.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

407 051

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

-- OFFICIAL DISTRIBUTION LIST --



Defense Documentation Center — 12 copies
Cameron Station
Alexandria, Virginia 22314

Office of Naval Research — 2 copies
Information Systems Program
Code 437
Arlington, Virginia 22217

Office of Naval Research — 6 copies
Code 102IP
Arlington, Virginia 22217

Office of Naval Research — 1 copy
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210

Office of Naval Research — 1 copy
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605

Office of Naval Research — 1 copy
Branch Office, Pasadena
1030 East Green Street
Pasadena, California 91106

New York Area Office — 1 copy
715 Broadway - 5th Floor
New York, New York 10003

Naval Research Laboratory — 6 copies
Technical Information Division, Code 2627
Washington, D. C. 20375

Dr. A. L. Slafkosky — 1 copy
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D. C. 20380

Office of Naval Research — 1 copy
Code 455
Arlington, Virginia 22217

Office of Naval Research — 1 copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center — 1 copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E. H. Gleissner — 1 copy
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland

Captain Grace M. Hopper — 1 copy
MAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D. C. 20350

Mr. Kin B. Thompson — 1 copy
Technical Director
Information Systems Division (OP-91T)
Office of Chief of Naval Operations
Washington, D. C. 20350

Advanced Research Projects Agency — 1 copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Professor Omar Wing — 1 copy
Columbia University in the City of New York
Department of Electrical Engineering & Computer Science
New York, New York 10027

1. RESEARCH AT YALE IN NATURAL LANGUAGE PROCESSING

This report describes the state of the computer programs at Yale that do automatic natural language processing as of the end of 1976. The theory behind the programs shown here as well as descriptions of how those programs function, has been described elsewhere. This report is intended solely as a summary of the capabilities of our computer programs at the present time. For each program an introduction to its overall intent is given. After the input/output for that program has been presented, a short discussion of the research underlying that program is given, together with a prognosis for the future development of the program.

The research we present here does not include every natural language processing project at Yale. Rather, we just include the efforts that have produced what we consider to be significant results.

The programs discussed are:

1 - SAM - A script-based story understanding program. SAM is intended to model a human story understander reading a story about which he has detailed knowledge upon which he can rely to help him to understand.

2 - FRUMP - A fast program designed to skim a newspaper looking for events in which it is interested. FRUMP is

based on what we call "sketchy scripts."

3 - PAM - A plan based program designed to understand stories that call upon general knowledge of human goals and relationships rather than the detailed script world of SAM.

4 - TALESPIN - A program intended to make up stories to tell in an interactive mode. TALESPIN uses plans and goals to determine how events relate in the world it is describing.

5 - WEIS/POLITICS - This is a program designed to read newspaper headlines and do two possible things: 1) It codes the sentences into a political coding scheme used by political scientists; 2) It simulates (in POLITICS mode) a person with an ideological belief system being informed of the event in the headlines. The program is then capable of answering questions based on its belief system about appropriate responses of the U.S. to the new events.

2 The Programs

2.1 SAM

SAM (Script Applier Mechanism) is a program running at Yale that was designed to understand stories that rely heavily on scripts. (SAM is described in Schank et al (1975), Schank and Abelson (1977) and Cullingford (1977). The stories processed by SAM tend to be fairly commonplace stories for which a rich knowledge base is available. Story 1 below relies on multiple knowledge bases (called scripts) as well as having a complication arise in one script as a result of an odd occurrence in a previous one. Story 2 is an actual newspaper story that SAM has processed. Stories 3, 4, 5, and 6 are stories that SAM has processed without the parser. That is, the parser is not yet ready to handle them but the rest of SAM can.

SAM understands these stories and others like them. By "understand" we mean SAM can create a linked causal chain of conceptualizations that represent what took place in each story. SAM parses the story into input conceptualizations using Riesbeck's analyzer (Riesbeck, 1975). These are then fed to a program that looks for script applicability (Cullingford, 1976). When a script seems to be applicable, it is used by the script applier to make inferences about events that must have occurred between events specifically

mentioned.

The final internal representation is a gigantic Conceptual Dependency network. We can generate paraphrases that are longer than the original because inferences made by the script applier are retained. We also generate 1) paraphrases that are shorter and closer to the original and 2) summaries that rely on measures of the relative importance of events within a script.

In addition, we have developed a program that can query the obtained representation so as to answer questions about the input story (see Lehnert, 1977).

Since the representation language that we use (Conceptual Dependency) is intended to be interlingual, generation in English is no harder for us than in any other language. We have also written generation programs that translate the stories we understand into Chinese, Russian, Dutch and Spanish. The translation programs work by taking the output from the script applier and using data from the target language in conjunction with Goldman's (1975) program.

SAM works by analyzing each sentence into a Conceptual Dependency representation. If this representation fits into a script, that script is brought into memory. Succeeding inputs are analyzed

and the result is looked for in the script. If the result is found any important events that are known to normally occur between the first input and the second are inferred to have happened. This continues until there are no new inputs or until a new input does not match a part of the current script.

Succeeding inputs may cause other scripts to be brought in. When a new input does not fit into the script-predicted sequence, SAM tries to find a script-based link between the input and what was expected.

Below is Story 1:

Story 1: John went to New York by Bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.

Long paraphrase: JOHN WENT TO A BUS STOP. HE WAITED AT IT A FEW MINUTES. HE ENTERED A BUS. THE DRIVER GOT THE TICKET FROM JOHN. HE WENT TO A SEAT. HE SAT DOWN IN IT. WHILE JOHN WAS ON THE BUS AN OLD LADY AND JOHN TALKED. THE DRIVER TOOK JOHN TO NEW YORK. HE WENT TO

THE DRIVER. WHEN GETTING OFF THE BUS JOHN THANKED THE DRIVER. JOHN GOT OFF IT.

HE ENTERED A STATION. HE PUT A TOKEN INTO THE TURNSTILE. HE WENT TO THE PLATFORM. HE WAITED AT IT A FEW MINUTES. HE ENTERED A SUBWAY CAR. A THIEF WENT TO JOHN. THE THIEF PICKED JOHN'S POCKET. HE WENT. JOHN WENT TO THE SEAT. HE SAT DOWN IN IT. THE DRIVER TOOK JOHN TO LEONE'S. HE LEFT THE SUBWAY CAR. HE LEFT THE STATION.

HE ENTERED LEONE'S. HE LOOKED AROUND INSIDE IT. HE SAW HE COULD GO TO A TABLE. HE WENT TO IT. HE SAT DOWN IN THE SEAT. HE ORDERED SOME LASAGNA. THE WAITER INDICATED TO THE CHEF JOHN WOULD LIKE HIM TO PREPARE SOMETHING. THE CHEF PREPARED THE LASAGNA. THE WAITER GOT IT FROM THE CHEF. THE WAITER WENT TO THE TABLE. HE SERVED THE LASAGNA TO JOHN. HE ATE IT. HE BECAME FULL.

HE ASKED THE WAITER FOR THE CHECK. JOHN GOT IT FROM THE WAITER. JOHN READ THE CHECK. JOHN DISCOVERED HE WAS UNABLE TO PAY THE CHECK. HE INDICATED TO THE WAITER HE WAS UNABLE TO PAY THE CHECK. THE MANAGEMENT TOLD JOHN HE WOULD HAVE HAD TO WASH DISHES. HE ENTERED THE KITCHEN. HE WASHED DISHES. HE LEFT LEONE'S.

HE WENT TO THE BUS STOP. HE WAITED AT IT A FEW MINUTES. HE ENTERED THE BUS. THE DRIVER GOT THE TICKET FROM JOHN. HE WENT TO THE SEAT. HE SAT DOWN IN IT. THE DRIVER TOOK JOHN TO NEW HAVEN. HE GOT OFF THE BUS.

(Paragraphing has been added to the computer output for ease of reading.)

Below are questions asked to SAM (in English), which have been answered by SAM:

Q1: Why did John go to New York?
A1: BECAUSE JOHN WANTED TO GO TO LEONE'S.
Q2: How did John get to Leone's?
A2: JOHN TOOK A BUS TO NEW YORK AND THEN HE TOOK THE SUBWAY TO LEONE'S.
Q3: Did anything unusual happen on the bus?
A3: I DON'T REMEMBER ANYTHING.
Q4: Did anything unusual happen on the subway?
A4: A THIEF PICKED JOHN'S POCKET.
Q5: What did John order?
A5: JOHN ORDERED LASAGNA.
Q6: Did anything unusual happen at the restaurant?
A6: JOHN DISCOVERED THAT HE COULDN'T PAY THE CHECK AND SO HE HAD TO WASH DISHES.
Q7: Why couldn't John pay the check?
A7: BECAUSE JOHN DID NOT HAVE ANY MONEY.
Q8: Why didn't John have any money?
A8: BECAUSE A THIEF PICKED JOHN'S POCKET.
Q9: How did John get back to New Haven?
A9: JOHN WENT FROM LEONE'S BY SUBWAY AND THEN HE WENT TO NEW HAVEN BY BUS.
Q10: How was John able to take the bus to New Haven?
A10: PROBABLY JOHN HAD A TICKET.

Below is the Chinese translation of Story 1 produced by SAM:

John tzuoh charng twu chih che daw Neou Iue chiuh le. Ta tzay charng twu chi che lii. Ta gen i wey lao tay tay tarn i tarn. Ta tsorng charng twu chih che lii chiuh chuh le. Ta shieh shieh sy ji. Ta tzuoh dih shiah huoo che daw Leone's chiuh le. Ta tzay dih shiah tiee luh che lii. I ge par shoou tou le chyan. Ta

tsorng dih shiah tiee luh che chuh chiuh le. Ta jinn le Leone's. Ta chr le i da li tsay. Ta tsorng char farng nadaw i ge janq dhan. Ta fa shyann ta bu kee yii fu chyan. Char farng jiaw ta bih shiu yaw shii bei tz. Ta tsorng fang guar lii chu chiuh le. Ta tzuoh charng tu chi che daw Shin Gang chiuh le.

Story 2 was obtained from the New Haven Register.
Story 2:

A New Jersey man was killed Friday evening when the car in which he was riding swerved off Route 69 and struck a tree. David Hall,27, was pronounced dead at the scene by Dr. Dana Blauchard, medical examiner. Frank Miller, 16, of 593 Foxon Road, operator of the vehicle, was taken by Flanagan Ambulance to Yale-New haven Hospital. He was treated and released. The Branford Fire Department extricated the passenger from the vehicle. No charges were made. Patrolman Robert Onofrio is continuing the investigation.

Question-Answering Output.

Q 1: Was anyone killed?
A 1: YES, DAVID HALL DIED.

Q 2: Was anyone hurt?
A 2: YES, FRANK MILLER WAS SLIGHTLY INJURED.

Q 3: Why was Frank Miller hurt?
A 3: BECAUSE THE AUTOMOBILE HIT A TREE.

Q 4: Did Frank Miller go to the hospital?
A 4: YES, HE WAS IN YALE-NEW HAVEN.

Q 5: How did Frank Miller get to the hospital?
A 5: AN AMBULANCE TOOK HIM TO YALE-NEW HAVEN.
Summary:
AN AUTOMOBILE HIT A TREE NEAR HIGHWAY 69 FOUR DAYS AGO. DAVID HALL, AGE 27, RESIDENCE IN NEW JERSEY, THE PASSENGER, DIED. FRANK MILLER, AGE 16 RESIDENCE AT 593 FOXON ROAD IN NEW HAVEN, CONNECTICUT, THE DRIVER, WAS SLIGHTLY INJURED. THE POLICE DEPARTMENT DID NOT FILE CHARGES.

The remainder of the stories shown here were processed without the parser. That is, they were input in Conceptual Dependency directly. Also, no question-answering run was made. The reason for this is that these programs are all separate entities and

the people concerned with parsing and question answering have not gotten around to looking at the kinds of things of which the script applier program is currently capable.

Story 3

A Pennsylvania man and his wife returning from a Cape Cod vacation were killed in a violent crash on the Connecticut Turnpike Saturday morning. The victims have been identified as John Gavin, 47, and his wife Mary, 41, of Morristown, Pa. Both were pronounced dead at the scene by a medical examiner. Mrs Gavin's 19-year-old son, Grant Butler, miraculously escaped injury in the one-car accident.

Summary:

A VEHICLE HIT SOMETHING NEAR A ROAD. JOHN AND MARY GAVIN, RESIDENCE IN MORRISTOWN, PENNSYLVANIA, DIED. UNEXPECTEDLY GRANT BUTLER, AGE 19, WAS NOT HURT.

Story 4

U Nu, the premier of Burma, and his wife, Mrs U, arrived by plane in Peking on October 8, 1954 at the invitation of the People's Republic of China. Chairman Mao and Mr U discussed economic relations between Burma and China.

Summary:

CHAIRMAN MAO, THE CHINA GOVERNMENT HEAD, AND PREMIER U NU, THE BURMA GOVERNMENT HEAD, DISCUSSED CHINA BURMA ECONOMIC AFFAIRS IN PEKING, CHINA.

Story 5

A passenger train carrying tourists collided with a freight train in the rugged Sierra Madre of northern Mexico. At least 18 persons were killed and 45 were injured, police reported today.

Summary:

A PASSENGER TRAIN HIT A CARGO TRAIN IN THE SIERRA MADRE MOUNTAINS. EIGHTEEN PEOPLE DIED. FORTY-FIVE PEOPLE WERE SLIGHTLY INJURED.

Story 6

Alan L Plucinski, age 17, of Fairfield, died Thursday morning in a motorcycle accident when the cycle he was driving crashed into a utility pole on Dalton Woods Road. Plucinski was pronounced dead on arrival at Milford Hospital, police said.

Summary:

A CYCLE HIT A POLE NEAR A ROAD TWO DAYS AGO. ALAN PLUCINSKI, AGE 17, RESIDENCE IN FAIRFIELD, CONNECTICUT, DIED.

future expectations: Our intention is to supply SAM with more and more scripts until it can read a great variety of stories. We may also develop parsers and generators in other languages so as to have a machine translation capability. We intend shortly to begin work on a Russian parser so as to begin to translate Russian newspaper stories into English.

2.2 FRUMP

FRUMP (Fast Reading and Understanding Memory Program) is a program that is intended to skim a newspaper quickly while looking for things it is interested in. FRUMP uses simpler, less detailed scripts than SAM. Furthermore, its parsing is done directly from the scripts. After a story has been identified to be relevant to a domain of of interest, the particular items that are interesting to FRUMP in

that domain are predicted. Special purpose expectations are set up to look for the concepts around which FRUMP's expectations are organized. Sentences are never completely parsed. When a relevant concept is found, rules of English are used to find the information that FRUMP wants to know.

FRUMP is a very fast program compared to SAM. While SAM often takes a few minutes of CPU time to read a long story, FRUMP rarely takes more than a few seconds to read the same story. SAM is set up to understand a domain fully enough so that it can detect problems or unusual circumstances and understand the kinds of complications that can arise in a story (as it did in story 1). However, for very static newspaper stories describing ordinary events, we have come to believe that the full power of SAM is a hindrance at times. FRUMP can do no more than skim a newspaper story about which it has knowledge, but it is very fast at what it does.

FRUMP also has an update mode in which it can read stories that are follow ups to stories it has already read. In update mode FRUMP only notices pieces of information connected with things it was interested in at the first reading that were left undetermined and standard follow up information. (FRUMP was programmed by Gerald DeJong).

Story 7

INPUT:

A passenger train carrying tourists, including some Americans, collided with a freight train in the rugged Sierra Madre of Northern Mexico, killing at least seventeen persons and injuring 45, the police reported today.

They said that at least five of the injured were Americans, and there were unofficial reports that one of the dead was from New York City.

Some of the passengers were travel agents, most from Mexico City, making the trip as part of a toursim promotion, the police said.

The American Society of Travel Agents had been meeting in Guadalajara, though it was not known whether any of the group were aboard the train.

One observation car on the railroad to the Pacific tumbled into a 45 foot canyon when the passenger train smashed into the freight yesterday afternoon near the village of Pittorreal about 20 miles west of Chihuahua City and 200 miles south of the United States border, the police said.

They said that rescue workers were still trying to pry apart the car's wreckage to reach passengers trapped inside. the rescue squads could not use cutting torches on the wreckage because spilled diesel fuel might ignite, the police reported.

SUMMARY

A TRAIN HIT A TRAIN IN MEXICO. 17 PEOPLE DIED. 45 PEOPLE WERE INJURED.

Story 8:

INPUT:

Officials here said today that the failure of an excursion train engineer to heed a stop signal was responsible for the collision with a freight train on Sunday that killed nearly a score of persons, including two Americans.

The district attorney's office said two other Americans had been injured in the collision near the Barranca Del Cobre in the Sierra Madre.

All the dead were Mexicans except the two Americans and two Britons. Most of those aboard were Mexican travel agents.

The Americans were identified as Mart Mortellaro of New York and Martin Ward whose hometown was still not available. The district attorney's office said the two injured Americans were Paul Joseph Callsen and Mary Callsen, both of New York.

SUMMARY

A TRAIN HIT A TRAIN IN MEXICO. 17 PEOPLE DIED. 45 PEOPLE WERE INJURED. THE ENGINEER WAS BLAMED FOR THE CRASH.

Story 9

INPUT:

At least 12 people were reported killed early today when an express train ran onto a flooded bridge whose rails had been swept away, crashed through it and plunged into a river in Kenya.

The official press agency reported that the death toll was at least 12 and that 70 were injured in what railroad officials called the worst passenger train disaster in East African history.

SUMMARY:

A TRAIN FELL INTO A RIVER IN KENYA. 12 PEOPLE DIED. 70 PEOPLE WERE INJURED.

Story 10

INPUT:

A severe earthquake struck Northeastern Italy last night, collapsing entire sections of towns northeast of Venice near the Yugoslav border, killing at least 95 persons and injuring at least 1000, the Italian interior ministry reported.

In the city of Udine alone, a government spokesman said they feared at least 200 dead under the debris. The city, on the main railroad between Rome and Vienna, has a population of about 90000.

The spokesman for the Caribinieri, the paramilitary national police force, said there had been reports of severe damage from half a dozen towns in the foothills of the Alps, with whole families buried in building collapses. Communications with a number of

points in the area were still out.

The earthquake was recorded at 6.3 on the Richter scale, which measures ground motion. In populated areas, a quake registering 4 on that scale can cause moderate damage, a reading of 6 can be severe and a reading of 7 indicates a major earthquake.

SUMMARY:

THERE WAS AN EARTHQUAKE IN ITALY. 95 PEOPLE DIED. 1000 PEOPLE WERE INJURED. THE EARTHQUAKE REGISTERED 6.3 ON THE RICHTER SCALE.

Story 11

INPUT:

A major earthquake struck a mountainous area of Eastern Turkey today, and the government said the death toll could be at least 3000.

First reports quoted official sources as having said that at least 574 had been killed near the Soviet and Iranian borders, but the defense ministry said it appeared that 3000 could be dead in one township alone.

The earthquake struck at 2 :45 pm (7 :25 New York time) near Mount Ararat in Van province. The Kandilli observatory in Istanbul said the quake had a magnitude of 7.8 on the Richter scale, describing it as the worse to strike turkey since another 7.8 tremor that killed 30000 people in 1939 in the Erzincan area.

Radio and television interrupted regular programming and played classical music in mourning for the victims.

SUMMARY:

THERE WAS AN EARTHQUAKE IN TURKEY. 3000 PEOPLE DIED. THE EARTHQUAKE REGISTERED 7.8 ON THE RICHTER SCALE.

Future expectations: In the near future we expect FRUMP to be tied in to one of the wire services on our computer. FRUMP will eventually be able to read incoming stories looking for items of particular interest to the users logged into the system. When it

finds something that conforms to interest specifications previously set by the users it will summarize the key points and flash that summary on the user's screen. Eventually we would hope that it will be able to answer questions supplied to it by the interested user by running SAM on the same story when that is requested.

2.3 PAM

PAM (Plan Applier Mechanism) is a program running at Yale that is intended to understand plan-based stories. (PAM is described in Schank and Abelson(1977) and in Wilensky (1976)). Stories (12) and (13) below were was processed by PAM. Input and output to PAM is in English.

A plan-based story is one in which the central characters must develop fairly novel plans to deal with problems and interferences that have occurred with respect to goals that they are interested in attaining. Truly interesting stories are plan-based rather than script-based.

Over the past two years we have been developing a theory of how people understand such complex stories (see Schank and Abelson(1977)). PAM was an early test of some of those theories (written by Robert Wilensky). PAM uses knowledge about goals, themes, and planboxes,

to figure out the intentions of all the characters it hears about. The program's job is to keep track of the goals of each of the characters in a story and to interpret their actions as means of achieving those goals. Below is Story (12), followed by the output PAM produces after processing it.

John loved Mary but she didn't want to marry him.
One day, a dragon stole Mary from the castle.
John got on top of his horse and killed the dragon.
Mary agreed to marry him.
They lived happily ever after.

OUTPUT:

Paraphrase

JOHN WAS IN LOVE WITH MARY. SHE DID NOT WANT TO MARRY HIM. A DRAGON TOOK HER FROM A CASTLE. HE LEARNED THAT THAT DRAGON HAD TAKEN HER FROM THE CASTLE. HE MOUNTED A HORSE. IT TOOK HIM TO HER. HE KILLED THE DRAGON. SHE WAS INDEBTED TO HIM. SHE TOLD HIM SHE WAS GOING TO MARRY HIM. HE MARRIED HER. HE AND SHE WERE HAPPY THEREAFTER.

Questions

Why did John get on his horse?
BECAUSE HE WANTED TO BE NEAR MARY.

Why did Mary agree to marry John?
BECAUSE SHE WAS INDEBTED TO HIM.

Why did John kill the dragon?
BECAUSE HE WANTED MARY TO NOT DIE.

Below is Story (13) and the output PAM creates for it:

INPUT
John disliked his job as the bank janitor.
One day, a thief tried to rob the bank.
John jumped on the thief and knocked him out.
The bank president made John the bank guard.
John enjoyed his new job.

OUTPUT

Paraphase

JOHN DISLIKED BEING A BANK JANITOR. A THIEF TRIED ROBBED THE BANK. JOHN LEARNED THE THIEF HAD ROBBED IT. JOHN WANTED THE THIEF TO NOT HAVE THE BANK'S MONEY. JOHN KNOCKED THE THIEF OUT. THE BANK WAS INDEBTED TO JOHN. THE PRESIDENT MADE JOHN A BANK GUARD. HE LIKED BEING IT.

Questions

Why did John knock the thief out?
BECAUSE HE WANTED THE THIEF NOT TO HAVE THE BANK'S MONEY.

Why did the bank president make John the bank guard?
BECAUSE THE BANK WAS INDEBTED TO JOHN.

In order to understand these stories, PAM, like SAM, must make implicit inferences to connect up the sentences of the story it is processing. The following are some of the inferences PAM makes for Story 12:

John wanted to marry Mary.
Mary was endangered by the dragon.
John learned that the dragon had kidnapped Mary.
John wanted to save Mary from the dragon.
John rode his horse to where Mary was.
Mary became grateful to John for rescuing her.
John and Mary got married.

Many of these inferences are dependent on knowledge about how goals can be achieved. For example, to understand why John killed the dragon, we must know that killing a captor is a way to free a captive. In addition, we must know that in order to rescue Mary, John must first be near her, that riding a horse is a way of changing one's location, and that to ride a horse, it is necessary to mount it. Without this knowledge, we could not understand why John got on his horse.

future expectations: We expect that in the next few years the bulk of the theoretical work that we will be doing will revolve around a goal and plan based understander. PAM is just a prototype of such an understander and we have stopped work on it in its present form. In the next year we expect to build a new version of PAM that can handle much longer and more complex stories that the one shown above. Ideally a final version of PAM would include SAM as a subroutine.

2.4 TALESPIN

TALESPIN is a program (written by Jim Meehan and described in Meehan (1976)) which makes up stories by simulating a world, assigning goals to some characters and saying what happens when these goals interact with events in the simulated world. The reader/user gets to supply much of the information about the initial state of the world, such as the choice of characters and the relationships between one character and another.

TALESPIN is, among other things, a program that tests the goal and planning apparatus that we have been developing (Schank and Abelson (1977)). The program tells stories by violating a goal state for one of the characters that it knows about and then creating a plan to achieve that goal.

Below are two stories written by TALESPIN. Story I is in an interactive mode where the user gets to make decisions. Story II is told with no user interaction. Upper case output is from the program, either as part of the story ("JOHN BEAR ATE THE HONEY"), or as part of the question/answer dialogue ("HOW HUNGRY IS IRVING BIRD?"), Our answers to these questions are typed when an asterisk (*) appears.

Story I:

```
 **********  WELCOME TO TALE-SPIN  **********
 CHOOSE ANY OF THE FOLLOWING CHARACTERS FOR THE STORY:
 (BEAR BEE BOY GIRL FOX CROW ANT HEN LION DOG WOLF
MOUSE
 CAT GOAT CANARY)

  *(BEAR BEE BOY CANARY)
  IRVING BEE IS AT THE BEEHIVE.
  SOME HONEY IS AT THE BEEHIVE.
  IRVING BEE HAS THE HONEY.
  SAM ADAMS IS AT A HOUSE.
  WILMA CANARY IS AT A NEST.

 CHOOSE ANY OF THE FOLLOWING PROPS:
 (BREADCRUMBS CHEESE BASEBALL)

*NIL

 CHOOSE ANY OF THE FOLLOWING MISCELLANEOUS ITEMS:
 (BERRIES FLOWER RIVER WORM)
 *(BERRIES WORM)

  SOME BLUEBERRIES ARE AT A BUSH.
  A WORM IS AT A PATCH OF GROUND.

WHO KNOWS ABOUT THE BLUEBERRIES?
 1: WILMA CANARY     2: SAM ADAMS     3: IRVING BEE
 4: JOHN BEAR
  *4

  JOHN BEAR THINKS THAT THE BLUEBERRIES ARE AT THE
BUSH.

HOW HUNGRY IS JOHN BEAR?
 1:VERY  2:SOMEWHAT  3:NOT VERY  4:NOT AT ALL
  *4

  JOHN BEAR IS NOT VERY HUNGRY.
```

WHO KNOWS ABOUT THE WORM?
1: WILMA CANARY 2: SAM ADAMS 3: IRVING BEE
4: JOHN BEAR
*2

SAM ADAMS THINKS THAT THE WORM IS AT THE PATCH OF GROUND.

THIS IS A STORY ABOUT ...
1: WILMA CANARY 2: SAM ADAMS 3: IRVING BEE
4: JOHN BEAR
*2
HIS PROBLEM IS THAT HE IS ...
1: HUNGRY 2: THIRSTY 3: HORNY 4: TIRED
*1

SAM ADAMS IS SOMEWHAT HUNGRY.
SAM ADAMS WANTS TO GET SOME BERRIES.
SAM ADAMS WANTS TO FIND OUT WHERE SOME BERRIES ARE.

DOES SAM ADAMS LIKE WILMA CANARY? 1:A LOT 2:A LITTLE 3:NOT
MUCH 4:NOT AT ALL
*2
SAM ADAMS WANTS WILMA CANARY TO TELL SAM ADAMS WHERE SOME BERRIES ARE.

DOES SAM ADAMS FEEL DECEPTIVE TOWARDS WILMA CANARY?

1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL *4
DOES SAM ADAMS FEEL COMPETITIVE TOWARDS WILMA CANARY?
1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL *2

SAM ADAMS DECIDES THAT WILMA CANARY MIGHT WANT SAM ADAMS TO GIVE WILMA CANARY A WORM. SAM ADAMS WANTS TO ASK WILMA CANARY WHETHER WILMA CANARY WILL TELL SAM ADAMS WHERE SOME BERRIES ARE IF SAM ADAMS GIVES WILMA CANARY A WORM. SAM ADAMS WANTS TO GET NEAR WILMA CANARY. SAM ADAMS WALKS FROM THE HOUSE TO THE GROUND BY THE REDWOOD TREE BY GOING THROUGH A VALLEY THROUGH A MEADOW. SAM ADAMS IS AT THE GROUND BY THE REDWOOD TREE. WILMA CANARY THINKS THAT SAM ADAMS IS AT THE GROUND BY THE REDWOOD TREE. SAM ADAMS ASKS WILMA CANARY WHETHER WILMA CANARY WILL TELL SAM ADAMS WHERE SOME BERRIES ARE IF SAM ADAMS GIVES WILMA CANARY A WORM.

DOES WILMA CANARY FEEL DECEPTIVE TOWARDS SAM ADAMS?
1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL *1

WILMA CANARY TELLS SAM ADAMS THAT WILMA CANARY WILL TELL SAM ADAMS WHERE SOME BERRIES ARE. SAM ADAMS THINKS THAT WILMA CANARY WILL TELL SAM ADAMS WHERE SOME

BERRIES ARE. SAM ADAMS WANTS TO GET A WORM. SAM ADAMS WANTS TO GET NEAR THE WORM.

IN CREATING A VALLEY, WE CAN MAKE UP A NEW ONE
OR USE AN OLD ONE. DO YOU WANT TO USE ANY OF THESE?
1: *VALLEY*0 2: *VALLEY*1
-- DECIDE: *YES
PLEASE TYPE AN INTEGER BETWEEN 1 AND 2
*2

SAM ADAMS WALKS FROM THE GROUND BY THE REDWOOD TREE TO THE PATCH OF GROUND BY GOING THROUGH THE MEADOW THROUGH WILMA CANARY THINKS THAT SAM ADAMS ISN'T AT THE GROUND BY THE REDWOOD TREE. SAM ADAMS TAKES THE WORM. SAM ADAMS WANTS TO GET NEAR WILMA CANARY. SAM ADAMS WALKS FROM THE PATCH OF GROUND TO THE GROUND BY THE REDWOOD TREE BY GOING THROUGH THE MEADOW THROUGH THE VALLEY THROUGH THE MEADOW. WILMA CANARY THINKS THAT SAM ADAMS IS AT THE GROUND BY THE REDWOOD TREE. SAM ADAMS GIVES WILMA CANARY THE WORM. WILMA CANARY THINKS THAT SAM ADAMS GAVE WILMA CANARY THE WORM. WILMA CANARY THINKS THAT WILMA CANARY HAS THE WORM.

HOW HUNGRY IS WILMA CANARY?
1:VERY 2:SOMEWHAT 3:NOT VERY 4:NOT AT ALL
*1

WILMA CANARY EATS THE WORM. THE WORM IS GONE. WILMA CANARY IS NOT VERY HUNGRY. WILMA CANARY TELLS SAM ADAMS THAT SAM ADAMS IS NOT AT ALL SMART. SAM ADAMS LIKES WILMA CANARY NOT AT ALL. SAM ADAMS TRUSTS WILMA CANARY NOT AT ALL. SAM ADAMS DECIDES THAT WILMA CANARY MIGHT WANT SAM ADAMS TO GIVE WILMA CANARY SOME CHEESE. SAM ADAMS ASKS WILMA CANARY WHETHER WILMA CANARY WILL TELL SAM ADAMS WHERE SOME BERRIES ARE IF SAM ADAMS GIVES WILMA CANARY SOME CHEESE. WILMA CANARY TELLS SAM ADAMS THAT WILMA CANARY WILL TELL SAM ADAMS WHERE SOME BERRIES ARE. SAM ADAMS WANTS TO GET SOME CHEESE. SAM ADAMS WANTS TO FIND OUT WHERE SOME CHEESE IS.

DOES SAM ADAMS LIKE IRVING BEE?
1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL *4

DOES SAM ADAMS LIKE JOHN BEAR?
1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL *4

SAM ADAMS DIDN'T FIND OUT WHERE SOME CHEESE IS.
SAM ADAMS DOESN'T GET SOME CHEESE.

DOES SAM ADAMS DOMINATE WILMA CANARY?
1:A LOT 2:A LITTLE 3:NOT MUCH 4:NOT AT ALL
*3

SAM ADAMS STRIKES WILMA CANARY. WILMA CANARY IS NOT AT ALL HEALTHY. WILMA CANARY WON'T TELL SAM ADAMS WHERE SOME BERRIES ARE. SAM ADAMS DIDN'T FIND OUT WHERE SOME BERRIES ARE. SAM ADAMS DOESN'T GET SOME BERRIES. THE END.

Below is Story II. It was generated by TALESPIN in a non-interactive mode. That is, TALESPIN itself made all the decisions.

ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND. THERE WAS A NEST IN AN ASH TREE. WILMA BIRD LIVED IN THE NEST. THERE WAS SOME WATER IN A RIVER. WILMA KNEW THAT THE WATER WAS IN THE RIVER. GEORGE KNEW THAT THE WATER WAS IN THE RIVER. ONE DAY WILMA WAS VERY THIRSTY. WILMA WANTED TO GET NEAR SOME WATER. WILMA FLEW FROM HER NEST ACROSS A MEADOW THROUGH A VALLEY TO THE RIVER. WILMA DRANK THE WATER. WILMA WAS NOT THIRSTY.

GEORGE WAS VERY THIRSTY. GEORGE WANTED TO GET NEAR SOME WATER. GEORGE WALKED FROM HIS PATCH OF GROUND ACROSS THE MEADOW THROUGH THE VALLEY TO A RIVER BANK. GEORGE FELL INTO THE WATER. GEORGE WANTED TO GET NEAR THE VALLEY. GEORGE COULDN'T GET NEAR THE VALLEY. GEORGE WANTED TO GET NEAR THE MEADOW. GEORGE COULDN'T GET NEAR THE MEADOW. WILMA WANTED GEORGE TO GET NEAR THE MEADOW. WILMA WANTED TO GET NEAR GEORGE. WILMA GRABBED GEORGE WITH HER CLAW. WILMA TOOK GEORGE FROM THE RIVER THROUGH THE VALLEY TO THE MEADOW. GEORGE WAS DEVOTED TO WILMA. GEORGE OWED EVERYTHING TO WILMA. WILMA LET GO OF GEORGE. GEORGE FELL TO THE MEADOW. THE END.

Below are two stories that were generated by TALESPIN which were mistakes. We present them here to give an idea of the kind of problems that TALESPIN solves in telling its stories. The solutions are only

very obvious when they are poor solutions as shown below.

Story III

HENRY ANT WAS THIRSTY. HE WALKED OVER TO THE RIVER BANK WHERE HIS GOOD FRIEND BILL BIRD WAS SITTING. HENRY SLIPPED AND FELL IN THE RIVER. HE WAS UNABLE TO CALL FOR HELP. HE DROWNED.

Story IV

HENRY ANT WAS THIRSTY. HE WALKED OVER TO THE RIVER BANK WHERE HIS GOOD FRIEND BILL BIRD WAS SITTING. HENRY SLIPPED AND FELL IN THE RIVER. GRAVITY DROWNED.

2.5 WEIS/POLITICS

One of the projects we are working on involves the coding of newspaper headlines into a coding scheme of use to political scientists. The WEIS program consists of two parts: a Conceptual Dependency parser, and a discrimination net that reads the output Conceptual Dependency parses into the appropriate codings. On occasion a fair amount of inference must be done in order to accomplish this coding after the initial parse.

Below are some example sentences that the WEIS program has parsed and coded (this program was written by Anatole Gershman and Gerald DeJong):

1-Lao forces abandon Ban-Nhil to North Vietnam.
2-Cuba grants asylum to a USA marine.
3-USA Navy task force which has been on patrol on the Indian Ocean for
  a month leaves the area.
4-Austria expelled 4 Chinese in a controversy over their status and
  activities.
5-Algeria protested to Spain the detention of an Algerian diplomat
  in connection with murder of an opposition leader.

The input to this program caused us to wonder about modelling the kinds of decisions that politicians make after receiving inputs of this kind. Some years ago Abelson (1966) constructed a program that was intended to be a model of an ideological cold warrior. It did not work very well largely because of the lack of understanding at the time of how to process natural language. With the advent of scripts as used by SAM, we set out to look at the problem again. Below is the output of the POLITICS program (written by Jaime Carbonell). Actual input and output is in Conceptual Dependency rather than English. (Actual English inputs and outputs should be finished shortly.) The input to POLITICS is a headline sentence of the WEIS type. POLITICS is then asked questions about what it believes underlies the motivations of the actors. The answers presented here are in accord with one possible political philosophy. POLITICS has a number of possible philosophies that can be input to it.

Russia massed troops on the Czech border.
Why did Russia do this ?

BECAUSE RUSSIA THOUGHT THAT IT COULD TAKE CONTROL OF CZECHOSLOVAKIA BY SENDING TROOPS.

What will Russia do next ?

RUSSIA MAY ORDER ITS TROOPS INTO CZECHOSLOVAKIA

What can the United States do ?

THE UNITED STATES CAN DO NOTHING, IT CAN INTERVENE MILITARILY IN CZECHOSLOVAKIA BY SENDING TROOPS, OR IT CAN INTERVENE DIPLOMATICALLY BY TALKING TO RUSSIA ABOUT CZECHOSLOVAKIA.

What should the United States do ?

THE UNITED STATES SHOULD INTERVENE MILITARILY.

Russia sent massive arms shipments to the MPLA in Angola.

Why did Russia do this?

RUSSIA WANTS TO CONTROL ANGOLA THROUGH THE MPLA.

What will the MPLA use the arms shipments for ?

THE MPLA WILL ATTEMPT TO TAKE OVER ANGOLA USING THE ARMS SHIPMENTS.

What will the other factions in Angola do ?

THE OTHER FACTIONS WILL ASK SOME OTHER COUNTRY FOR ARMS.

Which country ?

THE UNITED STATES.

Should the US do anything about it ?

YES, THE US SHOULD SEND ARMS TO THE OTHER FACTIONS IN ANGOLA.

future expectations: The POLITICS program was done largely to demonstrate the progress we have attained in natural language processing by solving a problem that was too difficult ten years ago. We consider the simulation of political and other beliefs to be of

prime importance, however. We have not yet decided whether to further extend this program.

3 OTHER PROJECTS

There are a number of other projects that are going on at Yale in natural language processing that have not been described here. The major ones are involved with the problems of learning and conversation.

There are two different learning projects underway. The first, FOUL-UP, is an adjunct to SAM that helps SAM parse texts that contain vocabulary items with which it is not familiar. FOUL-UP (written by Richard Granger) takes over control from the parser when the parser does not know a particular word. By using expectations already present in the parser and in the script applier, FOUL-UP can predict enough of the meaning of the word in question to allow SAM to continue along.

Examples of the kind of thing FOUL-UP can handle are:

1. The car swerved off the road. -where 'swerve' is unknown
2. The car ran into an elm. -where 'elm' is unknown

In (1), 'swerve' is rendered as PTRANS(for explanation of the primitive actions and other theoretical concepts underlying the representational system used here, see Schank(1975)) using parser and script applier expectations. In (2), 'elm' is rendered as 'obstruction'. The parser thus 'learns' the meaning of these words in the sense that it can predict and understand their use within the present context and ither similar contexts. The fact that 'elm' is a tree is not discernable of course, but neither is it relevant here.

The other learning project in which we are engaged is an attempt to model the rules by which a child learns elementary world knowledge and later learns to talk on the basis of that knowledge. The program we are developing *learns* complex knowledge from simple basic suppositions about what can reasonably be expected to be in the head of a child of age less than one year. The problem is to see how much we can learn from a minimal set of presumably innate knowledge. (Work on this program is being done by Mallory Selfridge.)

The conversational program is just in its beginning stages. Our goal is to create a program that can converse with a user in natural English. Presently, work is being done on establishing a 'syntax

of conversation' that will be able to catalog the kinds of possible responses there are to an input. Work on a 'semantics of conversation' would delimit the possible choices in a given situation.

BIBLIOGRAPHY

Abelson, R. P. (1966) Heuristic processes in the human application of verbal structures in new situations. Proc. XVIII International Congress of Psychology, Symposium 25. Moscow

Cullingford, R. E. (1976) The Uses of World Knowledge in Text Understanding. Proceedings of the International Conference on Computational Linguistics, Ottawa, Canada.

Cullingford, R. E. (1977) Organizing World Knowledge for Story Understanding by Computer. PhD. Thesis, Department of Engineering and Applied Science, Yale University, New Haven, Connecticut.

Goldman, N. (1975) Conceptual Generation. In R. C. Schank (ed), Conceptual Information Processing, North/Holland Publishing Company, Amsterdam.

Lehnert, W. G. (1977) Human and Computational Question-Answering, Cognitive Science, Volume I, Number 1.

Meehan, J. R. (1976) The Metanovel: Writing Stories by Computer. Technical Report #74. Computer Science Department, Yale University, New Haven, Connecticut.

Riesbeck, C. K. (1975) Conceptual Analysis. In R. C. Schank (ed), Conceptual Information Processing, North/Holland Publishing Company, Amsterdam.

Schank, R. C. and Yale A. I. Project (1975) SAM - A Story Understander, Research Report #43, Computer Science Department, Yale University, New Haven, Connecticut

Schank, R. C. (1975) Conceptual Information Processing. North/Holland Publishing Company, Amsterdam

Schank, R. C. and Abelson, R. P. (1977) Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Wilensky, R. (1976) Using Plans to Understand Natural Language. In proceedings of 1976 ACM Conference, Houston, Texas.