PICNICS, KITTENS AND WIGS: USING SCENARIOS FOR THE SENTENCE COMPLETION TASK

by
Candace L. Bullwinkle
Massachusetts Institute of Technology
Cambridge, Massachusetts

Abstract

This paper describes a program for filling in blank spaces in sentences which are given to elementary school children to test reading comprehension. The paper discusses the knowledge structure needed to represent what is being described In the sentences. As part of the structure, scenarios of classes of events are described. These scenarios contain general information about how the event occurs, who the likely actors are, and when the subparts of the event occur. The description of the scenario includes the use of free variables and the embedding of one scenario within another. Implementation of these characteristics of the scenario is described in sone detail.

Introduction

This paper addresses some of the questions Involved in performing the sentence completion task by computer. The sentence completion task is the process of finding a word (called a filler) for a blank space left in a sentence so that the sentence with the filler "makes sense" within the surrounding context. Sentence completion tasks are commonly used as part of U.S. reading comprehension tests for all educational grade levels, elementary through graduate school. Sentence completion tests may be as simple as example El below or as complex as E2.

**El) Joe can run very fast. He often wins when the boys <u>A.</u> at school.
A. race swim play jump [1]**

**E2) "I like you very ----." It is not difficult to guess the word which correctly completes the sentence because the structure of the English language sharply <u>A.</u> the number of choices possible. In many instances, however, one word gives the <u>B.</u> more exactly than does any other word.
A. inflates replies permits limits tones
B. result meaning origin grammar spelling [2]**

Basic Components

Consider the sentence completion test example below:

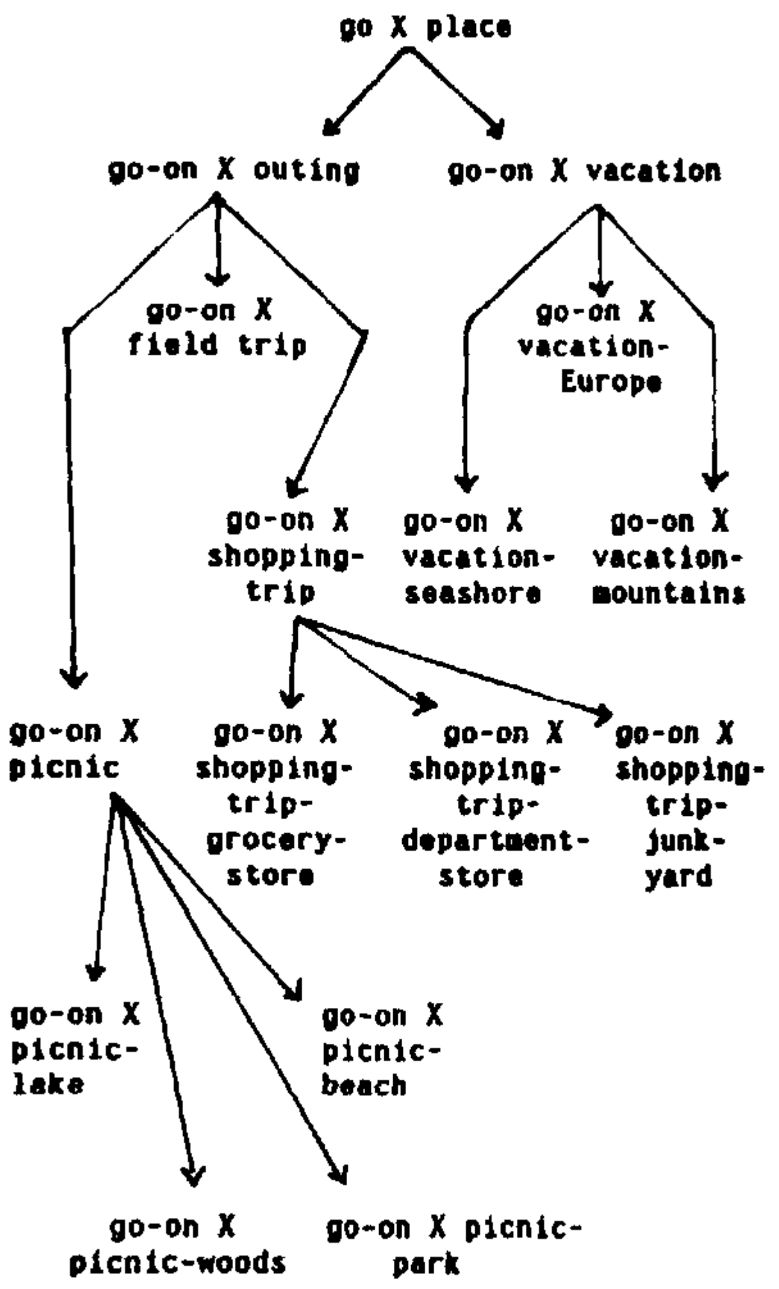
E3) It was a lovely, sunny morning. Bob and Betty were happy as they piled baskets of good things to eat in the car. ______ were going on a ______. [3]

The paragraph does not state in any part that Bob and Betty are going on a picnic. Instead the reader jumps to that conclusion from bits of information that "clue" him/her to what the story is about. Those clues are: *sunny day, happy people, food in baskets, food being put in a car and someone going somewhere.* I claim that the process of concluding that a picnic is about to begin is the process of abduction defined by Pierce [4] below and discussed by Pople [5] and Isner[6]:

The research described here was completed during study in the Department of Computer Science, University of Pittsburgh in partial fulfillment of the requirements for the degree of Hester of Science.

> Surprising fact C is observed. But if A were known to be true, then C would be a matter of course. Hence there is reason to suspect that A is true.

In my work, I call premises scenarios. Scenarios are organized in a hierarchy with each scenario describing the components of a complex action such as going on a picnic, possessing some object, eating food or the like. The scenario contains some general information about the actors of the action and more specific information about how the action Is carried out. Thus the scenario for going on a picnic tells us that the actors are humans, that the event takes place usually on a sunny day, that there are baskets of food which are loaded into a vehicle or carried by the particlpant(s), that the food is eaten at the picnic place, games may be played and the picnic ends when the actors leave. The scenario hierarchy is organized by Is-a-kind~of links between more general scenarios and sub-scenarios. Thus In the tree below, going on a picnic, going on a shopping trip and going on a field trip are all sub-scenarios of the going places scenario. Any actions true of a super-scenario also hold in its sub-scenario.



In addition to this general organization, every scenario has four special characteristics. First, the component clues (here called actions of a scenario) are divided by a time line of initiation,

performance and completion which correspond roughly to before, during and after the event. The time line is necessary because tense can be an important clue to understanding a test paragraph since the order of activities may suggest different scenarios. In example E4 below, the filler could be "grocery shopping trip" rather than "picnic" because the tense of going is past.

£4) It was a lovely, sunny morning. Bob and Betty were happy as they piled bags of good things to eat in the car.________had gone on a________.

Second, scenarios are designed so that other scenarios can be embedded in them. Embedding allows one scenario to function as a clue to another. For example, going on a picnic may have going grocery shopping embedded in It (for the food to be eaten at the picnic). Embedded scenarios imply that the time line of the embedded scenario is also embedded in the time line of the super-scenario and that all the Information about shopping can be used to explain the picnic event.

Third, each action of the scenario has two weights attached to it which measure the degree of association between that action and the scenario. The first weight, called an upward weight or evoking strength, measures the likelihood of an action evoking a scenario. Thus sunny day evokes picnics (as well as baseball games, tennis matches, and walks in the park) but having food in baskets more strongly evokes a picnic and hence it would have a larger numerical weight. The second weight, the downward weight or frequency strength, measures the frequency with which an action occurs when a scenario is known to be happening. For example, the downward weights of having food in baskets and eating that food are closely associated with picnics and will have a high numerical weight. Weights range over values of 0 to 5. This scheme of weights is patterned after the weighting scheme for medical diagnosis of Pople et al [8]. Weights used in sentence completion, however, have conditions attached to them. Thus the weight pattern for an action has the form:

if Conditions' to be true) then <welght value)

The conditionals include constraints such as the semantic class of the actor of the action, the number of actors (especially if it is plural) or the presence of some other action.

Fourth, a scenario is constructed using free variables to denote different classes of objects, places, or actors of the scenario. The free variables take on values from the sentence completion example, and some variables have default values which can be used in filling the blank in the sentence. Every scenario can be considered as a series of production-like rules of the form:

observation ♦ weight <— scenario

The left portion of the production is the action or the scenario and the right portion is the scenario pattern, generally referred to as the scenario when no confusion can result. In the complete scenario below for going on a picnic, each line represents two productions. Thus line LI can be interpreted as rules of the form:

```
(contain (OR baskets bags) food) + al <--
(initiation go-on X picnic)
(contain (OR baskets bags) food) <---
(initiation go-on X picnic) + bl
```

Note that the symbols P, V, and M denote "property", "verb form" and "semantic class of" respectively, and "hdvehicle" denotes objects in the semantic class of heavy duty vehicles (like buses or trucks).

```
                   go-on X picnic
Initiation
weights          observations
a1    b1         (V contain (OR baskets bags) food) L1
a2    b2         (V have X (OR baskets bags))
a3    b3         (P sunny weather)
a4    b4         (P warm weather)
a5    b5         (OR (V pile-in X (OR bags baskets) W)
a6    b6             (V put-on X (OR baskets bags) W)
a7    b7             (V carry X (OR baskets bags)))
a8    b8         (V make X food)
a9    b9         (V go-on X shopping-trip-grocery-stor

Performance
a10  b10         (OR (V go-on X W Z)
a11  b11             (V go-in X W Z)
a12  b12             (V walk-to X Z))
a13  b13         (V eat X food)
a14  b14         (V play-at X games Z)

Completion
a15  b15         (V leave X Z)

upward weights
a1    4.
a2    if (M human X) then 5.
a3    2.
a4    2.
a5    if (AND (M human X)(M vehicle W))
           then 4.
a6    if (AND (M human X)
                 (M hdvehicle W) then 4.
a7    if (AND (M human X)
                 (NOT (M vehicle W)))
      then 4. else if (AND (M human X)
                 (M vehicle W))) then 0.
a8    if (M human X) then 4.
a9    if (M human X) then 4.
a10   if (AND (M human X) (EQ (OR park
                 lake beach woods) Z)
                 (M hdvehicle W)) then 4.
a11   if (AND (M human X) (EQ (OR park
                 lake beach woods) Z)
                 (M vehicle W)) then 4.
a12   if (AND (M human X) (NOT
                 (M vehicle W))) then 4.
      else if (M vehicle W) then 0.
a13   if (M human X) then 4.
a14   if (AND (M human X)(M place Z))
      then 3.
a15   if (AND (M human X)(EQ (OR park
                 lake beach woods) Z))
      then 4.

downward weights
b1    5.
b2    if (M animate X) then 5.
b3    3.
b4    3.
b5    if (AND (M human X)(M vehicle W))
      then 5.
b6    if (AND (M human X)(M hdvehicle W))
      then 5.
b7    if (AND (M human X)
                 (NOT (M vehicle W))) then 5.
b8    if (M human X) then 4.
b9    if (M human X) then 3.
b10   if (M hdvehicle W) then 5.
b11   if (M vehicle W) then 5.
b12   if (NOT (M vehicle W)) then 4.
b13   if (M human X) then 5.
b14   if (AND (M human X)(M place Z))
      then 3.
b15   if (AND (M human X)
                 (EQ (OR park lake beach woods) Z))
      then 5.
```

Finally, one important consideration in the design of a sentence completion program should be mentioned. In examples El and E2, the reader is provided with a set of choices for what could go in each of the blanks. In E3, the reader must generate a filler without a set of possible fillers. In the implementation discussed below, I have designed the program so that it is unaware of any set of pre-selected possible answers. It may have been possible to design the program so that the computer would check each choice given and choose the one that matched the context most completely. Although humans often use the technique of checking the list of answers and reading through the sentence completion question with each choice in place of the blank, they are also capable of performing the task without this "backward search" from answers to questions. In fact exams given before the use of machine scoring did not have a list of choices. I have disregarded the backward search approach chiefly because I think It is subordinate to the general problem of choosing a filler for a blank with a whole data base of words that are "possible" answers.

Implementation

The first step of the implementation is processing the sentences of the test paragraph. This process reduces a sentence from one string of input to several different simpler pieces of information which are listed on an observations list (OUST, for short). The kind of Information produced from this phase is listed below from the sentence "Tom's cat had five kittens":

1. Semantic classes for Tom, cat and kittens.
2. Fact that Tom possesses the cat.
3. Fact that the cat is parent to the kittens,
4. Number of kittens is five.
5. Tense of fact #3 is past.
6. A series of special numbers to denote the uniqueness of Tom, cat and kittens.

In the next stage, each fact on the OLIST is matched to the action portion of a series of scenarios. If a match is found and the conditions on the weight are met, the free variables of the scenario are bound to the nouns in the fact from the OLIST. The resulting instantiation of the scenario pattern is then added to a scenario list. The process is then repeated for every fact of the OLIST.

Adding an Instantiated scenario pattern to the scenario list is done by means of synthesis, a technique of Pople [5]. The scenario pattern is matched to other scenario ps++as on the scenario list. Several differen and occur during the matching routine but will not be discussed here. For discussion see Bullwinkle [7]. Hatching has three purposes:

1. To avoid duplicating scenario patterns on the scenario list. Scenario patterns on the scenario list are considered the same when they have the same pattern and the same bindings for the variables of the scenario pattern.
2. To determine which actions on the OLIST evoked the same scenario pattern several times.
3. To sum together the evoking strengths of those actions of 2) and associate the resulting weight with the scenario pattern.

When all facts of the OLIST have been matched, the scenario with the largest weight is chosen as the probable scenario. It often happens in running this process that more than one scenario will have equal or nearly equal large numerical weights. For purposes of distinguishing between these and also for modelling all the events possible of the test paragraph, a partitioning phase occurs next.

For each scenario with large numerical weight, the rest of the scenarios on the scenario list are partitioned in major and minor lists using the principle of dominance (see Pople [8]) defined here as: scenario A dominates scenario B if the set of sentences which evoked A is a subset of the sentences which evoked B. The major list contains scenarios which either dominate or are dominated by the probable scenario. Scenarios on the major list represent alternatives to the probable scenario for explaining the test paragraph. The minor list can be viewed as a set of scenarios which when combined with the probable scenario explain more than any scenario can separately. Each scenario on the minor list is reviewed to see if it is an embedded scenario of the probable scenario. Those scenarios that are embedded are said to be causally linked to the probable scenario, and their weights are added to the weight of the probable scenario. The major-minor list construction is repeated for each probable scenario and the one with the highest weight Is then chosen as the most probable scenario.

From the most probable scenario, the blank in the sentence can be filled. This is accomplished in one of two ways. When a most probable scenario is chosen, the sentence with the blank is matched against that scenario pattern. If there is a match, a binding for the blank is selected. If there is no match, then the value for the binding of the blank is chosen off a special stack called the blank bindings stack of the scenario. The stack uis filled in the following way. During the process of evoking scenario patterns, a variable, called the blank variable, which represents the sentence blank, can become bound to a scenario variable. The scenario variable may also be bound to a word from the test paragraph, and so the blank variable "takes on" the sentence word as its value. When this happens, a copy of the blank variable and its value are put into the blank bindings stack. The stack is then used as indicated above: to select a filler for the blank. Note that several bindings for a blank variable may be In the stack. Only the top one Is chosen. Finally, the filler for the blank is submitted to a post-processor to check for such syntactic constraints as number of the noun or verb, tense of verbs and article agreement and semantic constraints such as the semantic class of nouns so that certain nonsense choices are thrown out. A small, unsophisticated processor has been implemented to do some of these tasks.

In the above discussion, all phases have been implemented except the process which reduces a sentence from one string to several simpler pieces of information. This portion of the program was hand-simulated In the cases given below and Its implementation Is a subject of further research.

Some real cases

The above implementation, written in LISP, has been run on example E3, a modified version of E3 given below as E5, and on E6 and E7.

E5) It was a lovely, sunny morning. Father came home with bags of groceries. Mother made sandwiches and cookies. Bob and Betty were happy as they piled baskets of good things to eat in the car._________were going on a______.

E6) Have you ever heard of a library without books? There is such a library, owned by a motion picture company, that contains nothing but hair. It has more than 50000 pieces--wigs, mustaches and beards in styles of many years, past and present. The next time you see your favorite brunette actress appear on the screen as a blond, either she may have had her hair bleached, or more likely, she is wearing a ______from this_______. [9]

E7) Tom's cat had five kittens; so Tom decided to take one of them from its _____ and give it to Ann. Tom went to the barn to get the _____ but came out quickly with three long scratches on his arm. The mother cat had let him know that she wanted all her _____. [10]

In E3, and E5 reasonable choices art made. The program chooses "Bob and Betty" and "Bob and Betty, Mother and Father" respectively for blank #B1, and "picnic" for #B2. While the actual filler should be "they", this choice is implementable in a post-processor with heuristics that tall it to use a pronoun of correct number and case if several words are to be put in one blank space. It is important for text explanation, however, that the program know who "they" refers to and hence the choices which were made are significant. In example E3, the blanks are filled from the scenario (init go-on X picnic) with X bound to "Bob and Betty". The minor list for this scenario provides no new information for filling the blanks. In example E5, the same scenario is chosen as the most probable scenario although X in that case is bound to "Bob and Betty, Mother and Father". One scenario on the minor list is of interest: (comp go-on X shopping-trip-grocery-store) with X bound to "Father". The program has determined that Father has been grocery shopping and this information is accounted for in the scenario of picnicking by use of the minor list.

Again in example E6 choices for the blank are the expected ones: "wig" for #B1 and "library" for #BZ. The choices for the blank come from the most probable scenario chosen for E6 which is (perf wear-from X wig Z) with X bound to "actress" and Z to "library." However, it is not until after the minor list of this scenario is checked that the binding for Z is made. On the minor list is the scenario (init borrow-from X Y library) with X unbound and Y bound to "wig". When this scenario becomes causally linked to (perf wear-from X wig Z), Z becomes bound to "library". This example shows the importance of embedded scenarios in binding variables for filling the blank.

Example E7 does not produce the expected results since it fails to choose the scenario which explains the most events and thes it provides incorrect fillers for the blank. Its failure points out the current limitations of the blankfillIng process implementation.

First, the use of weights requires a certain amount of tuning to determine what values for weights should be used and whether these should be summed in a linear or non-linear fashion. An alternative to the use of weights and the problems associated with tuning would be to eliminate the use of weights entirely in favor of a process that relies solely on the number of facts in the paragraph that it can explain.

Second, the present implementation of minor lists limits the causal links between embedded scenarios to a depth of one. This limitation is required to prevent the program from searching infinitely when looking for causal links between scenarios. However, in example E7, the causal link between the fact that Tom is scratched and the scenario of the cat having kittens is embedded too deeply for the current version of the minor list to find.

Third, synthesis on scenario lists at present does not permit scenario patterns with different time designations to establish a link with their common parent node. Thus the scenario patterns (init go-on Tom picnic) and (perf go-on Tom picnic) should become linked to the parent node (go-on Tom picnic) without a time reference. Establishment of this link would evoke the general idea of going on a picnic more strongly than either of two sub-parts of initiation or performance. Since the parent scenario will obtain its weight from a combination of the weights of its sub-parts, evoking the parent scenario could rule out some other scenario as the most probable scenario for the test example.

Conclusion

In this paper I have presented some examples of how scenarios can be used in the sentence completion task. I have illustrated the processes needed to obtain the scenario and fill the blanks as well as pointed out some remaining research problems involved. It is worthwhile to note here that the use of scenarios In a task such as blankfillIng reveals much of the assumed knowledge required of people in performing language related tasks. The scenario represents graphically the Information which must be acquired In order for a person to understand a particular concept and use it in language comprehension. In particular, people not familiar with picnicking customs may not draw the conclusion of "picnic" upon reading example E3 even though their reading comprehension of English is adequate. Additional research with scenarios shows promise of providing us with a better understanding of the "common sense" knowledge often assumed as part of language skills.

References

1. Stanford Achievement Test, primary level II, Harcourt, Brace and World, 1963.
2. Gates Basic Reading Exams, Basic Level of Comprehension L3, Bureau of Publications, Columbia University Press, 1958.
3. Stanford Achievement Test, Intermediate Battery Partial Form L, Harcourt,Brace and World, 1952.
4. Thomas A. Goudge,<u>The Thought of C.S. Peirce</u>. University of Toronto Press, Toronto, Canada, 1950.
5. Harry E. Pople. Jr. "On the Mechanization of AbductIve Logic," <u>1JCA1</u>, 1973.
6. Dale W. Isner, "Understanding 'Understanding* Through Representation and Reasoning," Ph.D. thesis, University or Pittsburgh, 1974.
7. Candace L. BullwInkle, "Computer Performance of the Sentence Completion Task," U.S. thesis, University of Pittsburgh, 1975.
8. Harry E. Pople, Jr., J. flyers, and R. Miller, "The DIALOG Model of Diagnostic Logic and its Use in Internal Medicine," in this volume.
9. Stanford Achievement Test, Intermediate Partial Form D, World Book Co., 1940.
10. Stanford Achievement Test, Revised Primary Form D, World Book Co., 1940.