Real Reading Behavior

Robert Thibadeau, Marcel Just, and Patricia Carpenter
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

The most obvious observable activities that accompany reading are the eye fixations on various parts of the text. Our laboratory has now developed the technology for automatically measuring and recording the sequence and duration of eye fixations that readers make in a fairly natural reading situation. This paper reports on research in progress to use our observations of this real reading behavior to construct computational models of the cognitive processes involved in natural reading.

In the first part of this paper we consider some constraints placed on models of human language comprehension imposed by the eye fixation data. In the second part we propose a particular model whose processing time on each word of the text is proportional to human readers' fixation durations.[1]

Some Observations

The reason that eye fixation data provide a rich base for a theoretical model of language processing is that readers' pauses on various words of a text are distinctly non-uniform. Some words are looked at very briefly, while others are gazed at for one or two seconds. The longer pauses are associated with a need for more computation [2]. The span of apprehension is relatively small, so that at a normal reading distance a reader cannot extract the meaning of words that are in peripheral vision [6]. This means that a person can read only what he looks at, and for scientific texts read normally by college students, this involves looking at almost every word. Furthermore, the longer pauses can occur immediately on the word that triggers the additional computation [4]. Thus it is possible to infer the degree of computational load at each point in the text.

The starting point for the computer model was the analysis of the eye fixations of 14 Carnegie-Mellon undergraduates reading 15 passages (each about 140 words long) taken from the science and technology sections of *Newsweek* and *Time* magazines (see the Appendix for a sample passage). The mean fixation duration on each word (or on larger, clause-like sectors) of the text were analyzed in a multiple regression analysis in which the independent variables were the structural properties of the texts that were believed to affect the fixation durations. The results showed that fixation durations were influenced by several levels of processing, such as the word level (longer, less frequent words take longer to encode and lexically access), and the text level (more important parts of the text, like topics or definitions take longer to process than less important parts). This analysis generated a verbal description of a model of the reading process that is consistent with the observed fixation durations. The details of the data, analysis, and model are reported elsewhere [5].

Some of the most intriguing aspects of the eye-fixation data concern trends that we have failed to find. Trends within noun phrases and verb phrases seem notable by their absence. Most approaches to sentence comprehension suggest that when the head noun of a noun phrase is reached, a great deal of processing is necessary to aggregate the meanings of the various modifiers. But this is not the case. While determiners and some prepositions are looked at more briefly, adjectives, noun-classifiers, and head nouns receive approximately the same gaze durations. (These results assume that word length effects on gaze duration have been covaried out). Verb phrases, with the exception of modals, show a similar flat distribution. It is also notable that verbs are not gazed at longer than nouns, as might be expected. Such results pose an interesting problem for a system which not only recognizes words, but also provides for their interpretation.

Another interesting result is the failure to find any associations with length of sentences (a rough measure of their complexity) or ordinal word position within sentences (a rough measure of amount of processing). That is to say, whether or not word function, character-length or syllables, etc., are controlled, there are no systematic trends associated with ordinal word position or sentence length. There is an added gaze duration associated with punctuation marks. Periods add about 73 milliseconds, and other punctuation (including commas, quotes, etc.) add about 43 milliseconds each above what can be accounted for by character-length or other covariates.

The Framework

The strategy for making sense of these and other similar observations is to develop a computational framework in which they can be understood. That framework must be capable of performing such diverse functions as word recognition, semantic and syntactic analysis, and text analysis. Furthermore, it must permit the ready interaction among processes implied by these functions. The framework we have implemented to accomplish these ambitious goals is a production system fashioned closely after Anderson's ACT system [1]. Such a production system is composed of three parts, a collection of productions comprising knowledge about how to carry out processes, a declarative knowledge base against which those processes are carried out, and an interpreter which provides for the actual behavior of the productions.

[1]This research was supported in part by grants from the Alfred P. Sloan Foundation, the National Institute of Education (G-79-0119) and the National Institute of Mental Health (MH-29617)

A *production* written for such a system is a condition-action pair, conceptually an 'if-then' concept, where the condition is assessed against a dynamically changing declarative knowledge base. If a condition is assessed as true (or *matched*), the action of the production is taken to alter the knowledge base. Altering the knowledge base leads to further potential for a match, so the production system will naturally cycle from match to match until no further productions can be matched. The sense in which processing is cotemporaneous is that all productions in memory are assessed for a match of their conditions before an action is taken, and then all productions whose conditions succeed take action before the match proceeds again. This cycling behavior provides a reference in establishing the basic synchrony of the system. The mapping from the behavior of the model to observed word gaze durations is on the basis of the number of match (or so-called *recognition-act*) cycles which the model requires to process each word.

The physical implementation of the model is equipped at present to handle a dependency analysis of sentences of the sort of complexity we find in our texts (see the Appendix). There is nothing new to this analysis, and so it is not presented here. The implementation also exihibits some elementary word recognition, in that, for a few words, it contains productions recognizing letter configurations and shape parameters. The experience is, however, that the conventions which we have introduced provide a thoroughly 'debugged' initial framework. It is to the details of that framework that we now turn.

Much of our initial effort in formulating such a parallel processing system has been concerned with making each processing cycle as efficient as possible with respect to the processing demands involved in reading to comprehend. To do this we allow that any number of productions can fire on a single cycle, each production contributing to the search for an interpretation of what is seen. Thus, for instance, the system may be actively working on a variety of processing tasks, and some may reach conclusion before others. The importance of concurrent processing is precisely that the reader may develop hypotheses in actively pursuing one processing avenue (such as syntax), and these hypotheses may influence other decisions (such as semantics) even before the former hypotheses are decided. Furthermore, hypotheses may be developed as expectations about words not yet seen, and these too should affect how those words are in fact seen. In effect, much of our initial effort has been in formulating how processes can interact in a collaborative effort to provide an interpretation.

Collaboration in single recognition-act cycles is possible with carefully thought out conventions about the representation of knowledge in the knowledge base. As in ACT, every knowledge base element in our model is assigned a real-number activation level, which in the present system is regard d as a confidence value of sorts. Unlike ACT, the activation levels in our model are permitted to be positive or negative in sign, with the interpretation that a negative sign indicates the element is believed to be untrue.

Coupled with this property of knowledge base elements are *threshold properties* associated with elements in the condition side of the productions. A threshold may be positive or negative, indicating a query about whether something is true or false with some confidence. As the system is used, there is a conventional threshold value above which knowledge is susceptible to being evaluated for inconsistency or contradiction, and below which knowledge is treated as hypothetical. In the examples below, this conventional threshold value is assumed. The condition elements can also include absence tests, so the system is capable of responding on the basis of the absence of an element at a desired confidence. Productions can also pick out knowledge that is only hypothetical using this device. But more importantly confidence in a result represents a manner in which productions can collaborate.

The confidence values on knowledge base elements are manipulated using a special action called <SPEW>. Basically, this action takes the confidence in one knowledge-base element and adds a linearly weighted function of that confidence to other knowledge-base elements. If any such knowledge-base element is not, in fact, in the knowledge base, it will be added. The elements themselves can be regarded as propositions in a propositional network. Thus, one can view the function of productions as maintaining and constructing coherent fields of propositions about the text.

Network representations of knowledge provide a natural indexing scheme, but to be practical on a computer such an indexing scheme needs augmentation. The indexing scheme must do several things at once. It must discriminate among the same objects used in different contexts, and it must also help resolve the difficult problem of two or more productions trying to build, or comment upon, the same knowledge structure concurrently. To give something of the flavor of the indexing scheme we have chosen: where other natural language understanding systems may create a token JOHN24 for a type JOHN, the number 24 in the present system does not simply distinquish this 'John' from others, it also places him within a dimensional space. In the examples to follow the token numbers are generated for the sequential gazes, 1 for the first and so on. An obvious use of such a scheme is that several productions may establish expectations regarding the next word. If some subset of the productions establish the same expectation, then without matching they will create the properly distinguished tokens for that expectation.

Consider one production written for this system:

```
((!WORD :IS !DETERMINER)
-->
(·^PEW> from (WORD :IS DETERMINER)
to (WORD :HAS (<TOK> DETERMINER-TAIL))
(DETERMINER-TAIL :HAS (<TOK> WORD-EXPECTATION))
(WORD-EXPECTATION :IS (<NEXTTOK> WORD)))
```

This production might be paraphrased as "If you see some particular word (say WORD12) is some particular determiner (say THE), then from the confidence you have that that word is that determiner, assign (arithmetic ADD) that much confidence to the ideas that that word a) needs to modify something (has a determiner-tail, DETERMINER-TAIL12), b) the modification itself has a word expectation (say WORD-EXPECTATION12), c) which is to be fulfilled by the next word seen (WORD13). The indexing scheme is manifest in the use of the functions <TOK> and <NEXTTOK>. It is important to be able to *predict* what a token will be, since in a parallel architecture several productions may be collaborating in building this expectation structure.

Type-token and category membership searches are usually carried out within the interpreter itself. The exclamation point prefix on subelements, as in !WORD above, causes the matcher to perform an ISA search for candidate tokens which the decision The matcher is itself dynamically altered with respect to ISA knowledge as new tokens are created, and by explicit ISA knowledge manipulation on the part of specialized productions. This has certain computational advantages in keeping the match process efficient[2]. The use of very many tokens, as implied by the above example, is important if one wants to explore the coordination of different processes in a parallel architecture.

The next production would fire if the word following the determiner were an adjective:

```
((!WORD :HAS !DETERMINER-TAIL)
(DETERMINER-TAIL :HAS !WORD-EXPECTATION)
(WORD-EXPECTATION :IS !1WORD)
(1WORD :IS !ADJECTIVE)
-->
(<SPEW> from (WORD-EXPECTATION :IS 1WORD)
to (WORD-EXPECTATION :IS 1WORD) -1
(WORD-EXPECTATION :IS (<NEXTTOK> WORD)))
```

The number prefixes, as in "1WORD", are tokens local to the production that just serve to indicate different knowledge base tokens are sought not what their knowledge base tokens should be. This production says that if a word has a determiner tail expecting some word and that word has been observed to be an adjective, then bring the confidence at least to 0.0 that the word-expectation is the adjective, and have confidence that the word-expectation is the word following the adjective.

The <SPEW> action of this production makes use of a weighting scheme which serves to alter the control of processing. In this framework any knowledge base element can serve as both a bit of knowledge (a link) and as a control value. The -1 number causes the confidence in the source of the spew to be multiplied by -1 before it is added to the target, (WORD-EXPECTATION :IS 1WORD). If this were the only production requesting this switch of confidence, the effect would be the effective deletion of this bit of knowledge from the knowledge base. If other productions were also switching this confidence, the system would wind up being confident that this word-expectation association is indeed not the case (explicitly false).

[2]The matcher is a slightly altered form of the RETE Matcher written by Forgy for OPS4 [3].

Processes in Sequence

The primary interest in formulating a model is in having as much 'processing' or decision-making as possible in a single recognition-act cycle. The general idea is that an average gaze duration of 250 milliseconds on a word represents few such cycles. The ability of the model to predict gaze duration, then, depends upon the sequential constraints holding among the collection of productions brought to the interpretation process. The 'determiner tail' productions illustrated above represent a processing sequence in most contexts; the second cannot fire until the first has deposited its contribution in the knowledge base. This is not a necessary feature of these two productions, since other productions can collaborate to cause the simultaneous matching of the two productions illustrated (we assume these are easy to imagine). However, one may note that since the 'determiner tail' productions are distributed over several word gazes, they at most contribute one processing cycle to the gaze on any word (besides the determiner). Thus, sequencing over words may not be expensive. Let us consider where it is computationally expensive.

In contrast to rightward looking activities, the presence of strong sequencing constraints among productions is potentially costly in leftward looking activities. To illustrate how such costs might be reduced, consider a production with a fairly low threshold which assigns a need to find an agent for an action-process verb, and another production which says that if one has an animate noun preceding an action-process verb and that animate noun is the only possible candidate, then that animate noun is the agent. These two productions are likely to fire simultaneously if the latter one fires at all. They both create a need to find an agent and satisfy that need at once. They do not set word expectations simply because the look-back at previous text tries to be efficient with regard to sequencing constraints. Had the need not been immediately fulfilled, it would serve as a promotion of other productions which might find other ways of fulfilling it, or of reinterpreting the use of the action-process verb (even questioning the ISA inference). It should be noted that the natural device for keeping these further productions in sequence from firing is having them make the absence test, as in

```
((!WORD :IS !ACTION-PROCESS-VERB)
(WORD :HAS !AGENT)
(<ABSENT> (AGENT :IS !ANYTHING))

-->
...suggest this might be an imperative, passive,
        ellipse, etc.)
```

The interpretation of the production is that "if you know with confidence that you have an action-process-verb and it needs an agent, but you don't know what that agent is, then suggest various reasons why you might not know with appropriately low confidence in them."

Coordination of Mind and Eye

The basic method of coordinating eye and mind in the present model is to make getting the next word contingent upon having completed the processing on the present one. In a production system architecture, this simply means that the match fails to turn up any productions whose conditions match to the knowledge base. Since elements in the knowledge base specify the need-to-know as wel; as what is known, the use of absence tests in the conditions of productions can 'shut off' further processing when it is deemed to be completed, or simply deemed to be unnecessary. It is by this device that the system demonstrates more processing on important information, 'shutting off' extended processing on that which is deemed, for any number of reasons, as less important.

The model must, in addition to various ideas about coordination, be also capable of representing various ideas about dis-coordination. One potential instance of this in the present data is that while virtually every word is fixated upon at least once (recall that several fixations can count toward a single gaze), there are some words, AND, OR, BUT, A, THE, TO, and OF, with some likelihood of not being gazed upon at all (this accounts in some part for the fairly low average gaze duration on these words). This can be considered a dis-coordination of sorts, since to be this selective the reader must have some reasonable strong hypotheses about the words in question (the knowledge sources for these hypotheses are potentially quite numerous, including the possibility of knowledge from peripheral vision). A production to implement this dis-coordination in the present system is:

```
((!WORD :IS !FREQUENT-FUNCTION-WORD)
-->
(<SPEW> ((<OLDTOK> GOAL) :IS INTERPRET-WORD)
((<OLDTOK> GOAL) :IS INTERPRET-WORD) -1
((<OLDTOK> GOAL) :IS GAZE-NEXT-WORD)))
```

This production detects the presence of one of the above function words, and immediately shifts the present goal of interpreting a word (if it happens to be that) to gazing upon the word following the function word. It is important to recognize that the eye need not be on the function word for the system to know with reasonable confidence that the next word is a function word. The indexing scheme permits the system to form hypotheses strong enough to create effective reality (e.g., peripheral information and expectations can add up to the conclusion that the word is a function word). A second important property is that the system does not get confused with such skips, or in the usual case with such brief stays on these words. The reason again is because each word becomes a sort of local demon inheriting demon-like properties from general production. and by interaction with other knowledge base elements through the system of productions.

Summary

This report has provided a brief description on work in progress to capture our observations of reading eye-movements in computational models of the reading process. We have illustrated some of the main properties of reading eye-movements and some of the main issues to arise. We have also illustrated within an implemented system how these issues might be addressed and explored in order to gain insight into more precise queries about real reading behavior.

Appendix

An example text:

Flywheels are one of the oldest mechanical devices known to man. Every internal-combustion engine contains a small flywheel that converts the jerky motion of the piston into the smooth flow of energy that powers the drive shaft. The greater the mass of a flywheel and the faster it spins, the more energy can be stored in it. But its maximum spinning speed is limited by the strength of the material it is made from. If it spins too fast for its mass, any flywheel will fly apart. One type of flywheel consists of round sandwiches of fiberglas and rubber providing the maximum possible storage of energy when the wheel is confined in a small space as in an automobile. Another type, the "superflywheel", consists of a series of rimless spokes. This flywheel stores the maximum energy when space is unlimited.

References

1. Anderson, J. R. *Language, memory, and thought.* Lawrence Erlbaum Associates, 1976.

2. Carpenter, P. A., & Just, M. A. Reading comprehension as the eyes see it. In *Cognitive Processes in Comprehension*, M. A. Just & P. A. Carpenter, Eds., Lawrence Erlbaum Associates, 1977.

3. Forgy, C. L. *OPS4 User's Manual.* Department of Computer Science, Carnegie-Mellon University, 1979.

4. Just, M. A., & Carpenter, P. A. Inference processes during reading: reflections from eye-fixations. In *Eye Movements ɩ 1d the Higher Psychological Functions*, J. W. Senders, D. F. Fisher, and R. A. Monty, Eds., Lawrence Erlbaum Associates, 1978.

5. Just, M. A., & Carpenter, P. A. "A theory of reading: from eye fixations to comprehension." *Psychological Review* (In Press).

6. McConkie, G. W., & Rayner, K. "The span of the effective stimulus during a fixation in reading." *Perception and Psychophysics 17* (1975).