Improving the Reading Rate of Double-R-Language

Mary Freiman[1] and Jerry Ball[2]
L3 Communications[1]
Air Force Research Laboratory[2]
Mary.Freiman@mesa.afmc.af.mil, Jerry.Ball@mesa.afmc.af.mil

Abstract

This paper describes changes to a model of reading comprehension to improve its reading rate and bring it into closer alignment with human reading rates. The broader context of the research is development of language capable synthetic teammates that can be integrated into team training simulations. To use synthetic teammates in team training without detriment, we believe the synthetic teammates must be both functional and cognitively plausible. By functional, we mean that the synthetic teammate operates in real time, performs the task, and handles the range of linguistic inputs that are encountered. By cognitively plausible, we mean that the synthetic teammate adheres to well established cognitive constraints on human language processing—including the incremental and interactive processing of language at human reading rates. Achieving human reading rates in a cognitively plausible and functional model of reading comprehension is a research challenge that has not been met to date.

Keywords: human language processing, reading rate, synthetic teammate, functional, cognitively plausible

Introduction

We are developing a model of reading comprehension called Double-R-Language (Ball, 2007; Ball, Heiberg & Silber, 2007). Double-R stands for Referential and Relational—two key dimensions of meaning that get grammatically encoded in English. The initial application of the reading model is development of a synthetic pilot for use in a three-person UAV simulation. The synthetic pilot flies the simulated UAV from a ground control station and will eventually communicate with a human navigator and photographer in the completion of reconnaissance missions. A prototype system has been developed (Ball, et al., 2009) using the ACT-R Cognitive Architecture (Anderson, 2007). The synthetic pilot prototype communicates with lightweight agent versions of the navigator and photographer developed outside ACT-R.

The prototype communicates with the navigator and photographer using text chat and must be capable of reading and comprehending the messages it receives from them. The reading comprehension model is capable of incrementally processing linguistic inputs and generating linguistic representations of referential and relational meaning. These linguistic representations are interactively mapped into a non-linguistic representation of the objects and situations referred to in the linguistic input. The non-linguistic representation—called the situation model (cf. Zwann & Radvansky, 1998)—drives the task behavior of the synthetic pilot and determines when to communicate with the other teammates to acquire needed information.

A significant challenge for the reading comprehension abilities of the model is input variability. A corpus of text chat communications that was collected in an experiment involving human subjects and the UAV simulation is full of variability in the form of linguistic input (see Table 1). For competent readers, misspelled words activate the intended lexical items because they contain many of the same letters and trigrams (Perea & Lupker, 2003). Hence, key requirements of the reading model include the ability to handle misspellings in input; the ability to separate perceptually conjoined units (e.g. separating punctuation from words as in “He went.”, but not “etc.”; separating words lacking spaces as in “yougo” for “you go”); and the ability to recognize multi-word expressions (e.g. “speed up”) and multi-unit words (e.g. “a priori”, “h-area”).

Table 1. Messages seen during a UAV simulation

MESSAGE:	VARIANT:
i need to be beloe 3000 for f area	i; beloe; f area
effective radiu	radiu
any requirements for altitde/speed?	altitde
can yougo faster yet or is it stll 200	yougo; stll

To satisfy these requirements, the model includes a word recognition subcomponent that uses ACT-R’s spreading activation mechanism to influence lexical item retrieval. The subcomponent maps orthographic input directly into DM representations without recourse to phonetic processing, although a phonetic mapping is not precluded.

The model uses the spreading activation mechanism of ACT-R to retrieve words from the lexicon that are not an exact match to the input. Letters and trigrams in the input spread activation to the words containing those letters and trigrams in the mental lexicon. These processes and encodings are based on the Interactive Activation model of word recognition (McClelland & Rumelhart, 1981), with the addition of trigrams based on “letter triples” (Seidenberg & McClelland, 1989). The subcomponent is embedded in the reading comprehension model as a whole; the effects of context and previous activation levels are taken into consideration when encoding each individual word (Freiman & Ball, 2008). The reading model also includes a verification stage to check the retrieved lexical item against the perceptual input. The verification stage aligns with the Activation-Verification model of Paap et al. (1982). It splits concatenated words in the input (e.g. “yougo”) to match the retrieved word (e.g. "you"), leaving a residual (e.g. "go") for subsequent processing. If the retrieved lexical item is not a sufficiently close match to the input, the model treats the input as an unknown word.

Even without considering the mapping of the linguistic representations into the situation model, the previous version of the reading model was much slower than humans in both cognitive processing time and real time performance. Adult readers read at a rate of 200-300 words per minute (Taylor, 1965; Carver 1973a; Carver 1973b). The average reading rate of the model—prior to the introduction of the changes described in this paper—was 96 words per minute (cognitive processing time), making it impossible to match the model's performance against human performance. Since we are interested in building a model of reading comprehension that is cognitively plausible as well as functional, this presents a real challenge. The prior reading model read slowly for several reasons: 1) it required multiple declarative memory (DM) retrieval requests per word; 2) it lacked the ability to read units of language larger than the word; and, 3) it built complex linguistic representations necessitating the execution of multiple productions. In addition, the model relied on parallel spreading activation to retrieve lexical items, which is computationally expensive for large DMs on serial hardware.

It is important to distinguish between reading rate as measured by the real time functional performance of the model and the rate as measured by the cognitive processing time. ACT-R provides support for measuring cognitive processing time—how long it would take a human to perform some cognitive process. Execution of a single production in ACT-R takes 50ms of cognitive processing time; plus, the time it takes to retrieve a chunk from DM depends on the activation of the chunk and can be measured. Typical ACT-R models with small DMs are capable of executing much faster than real time while measuring cognitive processing time. However, large DMs tax the computational resources of serial hardware and can lead to models that run slower than real time or not at all (cf. Douglass, Ball & Rodgers, 2009). Although it is important to distinguish cognitive processing considerations from real time considerations, these considerations are intertwined. For example, reducing cognitive processing time by eliminating retrievals also reduces the computation of parallel spreading activation, speeding up the real time performance of the model. For each of the shortcomings listed above, one or more remedies is described below and its impact on cognitive and real time processing is considered.

Reducing retrievals

When the model retrieves chunks from DM, the ACT-R Declarative Memory module calculates the activation across all chunks that match the retrieval template, selecting the most highly activated chunk for retrieval. The retrieval template provides hard constraints on memory retrieval—which are difficult to justify from the perspective of cognitive plausibility. Only chunks exactly matching the retrieval template are eligible for retrieval. The spreading activation mechanism provides more cognitively plausible soft constraints on retrieval. Chunks may be activated which are not an exact match to current input or context. For cognitive plausibility, we prefer ACT-R's spreading activation based soft constraint retrieval mechanism, minimizing the use of hard constraints in the retrieval template. For example, we do not want to use a hard constraint exact match to the input which would preclude retrieval of a word which is not an exact match (e.g. "altitde" should retrieve "altitude"). However, use of hard constraints reduces the amount of computation significantly by eliminating non-matching DM elements from the spreading activation computation.

Instead of relying on hard constraint retrievals to reduce the amount of computation, we have pursued the more cognitively plausible alternative of reducing the number of retrievals. An example of this is discussed next.

Combining Word Form and Part of Speech Chunks

In the previous version of the model, there was a word-form chunk for each word that encoded the graphical form of the word, including the letters and trigrams in the word (e.g. speed-wf), and part of speech chunks that encoded the various parts of speech of the word (e.g. speed-noun and speed-verb). The performance of the reading model has been improved significantly by collapsing the word form and part of speech chunks into a single word-form-pos chunk (e.g. speed-wf-noun, speed-wf-verb). Now, a single retrieval is required to determine the part of speech of a linguistic input. Since the production which initiates a retrieval takes 50ms to execute, by combining the word form and part of speech chunks for each lexical item, 50ms plus the retrieval time were saved per word.

From a representational perspective, combining the word form and part of speech chunks is not ideal. The word-form-pos chunks combine two distinct types of information (i.e. graphical vs. grammatical) which are better kept separate. A better solution would retain separate chunks, but support retrieval of part of speech chunks given the linguistic input. This could be achieved via multi-level activation spread if the linguistic input activated a word form chunk which in turn activated related part of speech chunks. Unfortunately, ACT-R does not support multi-level activation spread, although its predecessor ACT* (Anderson, 1983) did. It should be noted that single level parallel spreading activation is already computationally expensive for large DMs. Supporting multi-level spreading activation would add an additional multiple to the computation for each level.

Expanding the Perceptual Span

By default, ACT-R's vision module splits input text into perceptual spans at spaces and punctuation. The module even splits at word internal punctuation, so "*ACT-R"* becomes *"ACT" "-" "R"*, requiring three movements of attention to read. This behavior was changed to a more plausible splitting of the input text, thereby reducing the number of retrievals per input. Words with internal punctuation are no longer split up and retrieved separately.

The width of the perceptual span is now determined dynamically, based on the length of the first word ($word_n$) in the perceptual span. The boundary of $word_n$ is determined by the first space. If $word_n$ is greater than twelve letters in length, it takes up the entire length of the perceptual span. If $word_n$ is fewer than twelve letters in length, up to six letters of the next word ($word_{n+1}$) can also be seen in the perceptual span. No more than twelve letters are contained in the perceptual span.

The size of the revised perceptual span is deliberately conservative, so that even though three very short words (e.g. "out of the") could be perceived at a single attention fixation, the model never retrieves information for more than two words. There is a great deal of evidence that the perceptual span of adult readers is about 14-15 letters to the right of fixation (DenBuurman et al., 1981; McConkie & Rayner, 1975; Rayner, 1986). We implemented a span of up to twelve letters, with the greatest amount of activation spreading from the first few letters of the span and decreasing toward the end of the span. As a result, incorrect letters at the beginning of words are more detrimental to correct retrieval than misplaced letters later in the word. Activation spreads from the letters, trigrams, and length of the first word ($word_n$). If there is more than one word in the perceptual span, $word_{n+1}$ spreads activation from its trigrams. The section of the perceptual span containing $word_n$ is roughly equivalent to the fovea; the perceptual span at $word_{n+1}$ is roughly equivalent to the parafovea.

The revised perceptual span is generally larger than ACT-R's default span. Just as for adult readers, information to the right of fixation is obtained when the next word is predictable from the preceding text (Balota, Pollatsek, & Rayner 1985). Again, we were deliberately conservative in determining how much information could be perceived from $word_{n+1}$. Our intent was not to model in high fidelity the perceptual span in reading, or movements of attention in reading; movement of attention is not our primary focus. We merely wanted to make the vision module more serviceable to our language comprehension model, and more faithful to human perceptual spans in the process.

An example of the reduction in reading time can be seen in the phrase "take us to h-area". Previously, ACT-R's vision module would chop the input into seven parts:

"take" "us" "to" "h" "-" "area"

The model would retrieve each part from DM, integrate it into a linguistic construction, and then move on to the next word. The last three sections of the input would need to go through additional processing for the model to recombine them into a single word. Reading the entire sentence took 2.8 seconds. If ACT-R does not chop up the input at spaces and punctuation, the same phrase takes only 1.74 seconds to read. In the next section, the advantage of the expanded perceptual span for processing multi-word expressions is described.

Multi-Word Expressions

To facilitate reading and word recognition we have modified the ACT-R architecture and the reading model to better interpret multi-word expression (i.e. lexical units containing spaces). By not splitting the perceptual input at all spaces, multi-word expressions and multi-unit words can be retrieved as a single chunk (e.g., "of course" and "a priori"). To accommodate multi-word expressions we modified our lexical chunks in DM to reduce the number of retrievals necessary per word. Multi-word expressions are treated in much the same way as singleton words. Many multi-word expressions are not syntactically alterable units and need not be parsed (Sag et al. 2002), so the model treats them as "words-with-spaces".

An important side effect of the new perceptual span mechanism is that it also increases the reading rate of the model in the process. Since the perceptual span can cross spaces as well as punctuation, multi-word units like "to go", "want to", and "believe in" can be recognized as a single unit and processed in a single attention fixation. This capability is really the key to getting Double R-Language to approach adult human reading speed.

Before the multi-word expression capability was implemented, the phrase "we need to go" took 1.99 seconds for the model to process. After the perceptual span was expanded, the model reads the same phrase in 1.79 seconds. In this phrase "to go" is treated as a single unit, since it is an infinitive verb. There is one fewer retrieval, and the infinitive can be integrated into the phrase as a whole without having to recombine "to" and "go". Whenever there are multi-word units, the model now saves time in retrievals and processing. There is no difference in the time it takes to process other sorts of words. In addition, multi-word expressions are less ambiguous than individual words. "To" in isolation is very ambiguous, whereas "to go" is much less ambiguous.

Linguistic Representations

The reading model incrementally processes the linguistic input and builds a representation of referential and relational meaning that is mapped into the situation model. The building of linguistic representations is driven by the execution of productions which retrieve or construct linguistic elements and integrate them into the evolving representation. It takes more productions and retrievals to build complicated linguistic structures. In an effort to reduce the number of productions and retrievals that are required, we investigated how linguistic representations could be simplified or reduced. Our current approach attempts to build the minimal structure needed to represent the linguistic input, but must support more complex structures when they are needed.

Retrieving object referring expressions

Determiners are words that project definiteness and (sometimes) number information to nominals (Ball, 2010). In the reading model, nominals are called object referring expressions (ORE) to emphasize their referential (referring expression) and relational (object) functions. Determiners include the articles “a”, “an”, and “the”, as well as the negative “no”, demonstrative pronouns “this”, “those”, etc. Linguists have long known that the determiner “the” is the most commonly used word in the English language (cf., Zipf, 1932); other determiners are nearly as common. As the most commonly used words, determiners are likely to be highly proceduralized or simplified in their use (Zipf, 1949). Therefore we concentrated on consolidating the processes associated with determiners.

Previously, the model identified a word as a determiner, then executed a production which projected an ORE. The determiner was integrated as the specifier of the ORE. Given that determiners are used so regularly and frequently, it seems likely that there is an ORE in DM associated with each determiner that can be retrieved without first identifying the part of speech of the word. By retrieving the associated ORE rather than first identifying the word as a determiner, the processing of determiners becomes more proceduralized, faster, and more cognitively plausible. Where separate, general productions were required to retrieve the part of speech, followed by projection of an ORE if it’s a determiner, now a single specialized production projects an ORE directly from determiners. Although we manually created this specialized production, we would prefer that the model learn how to compile such productions automatically.

Reducing structure in nominal heads

Retrieval or projection of an ORE by a determiner establishes the expectation for a head to occur. In the previous version of the model, when a word following the determiner was identified as a noun, a subsequent production projected an object head and integrated the object head as the head of the ORE (Figure 1). Projection of the object head from the noun supported the integration of pre- and post-head modifiers (e.g. the post-head modifier “on the runway” in “the airplane on the runway”). When a post-head modifier occurred, it could be integrated into the object head in the post-head modifier slot. However, in the absence of a post-head modifier, projection of an object head is unnecessary and the noun could be integrated as the head of the ORE. The current version of the model adopts the simpler approach, integrating the noun as the head of the ORE (Figure 2). The tree diagrams below were generated by the previous and current versions of the model and show the contrast between the two approaches for the linguistic input “the restriction” (the pre- and post-head modifier slots in the object head are not displayed):

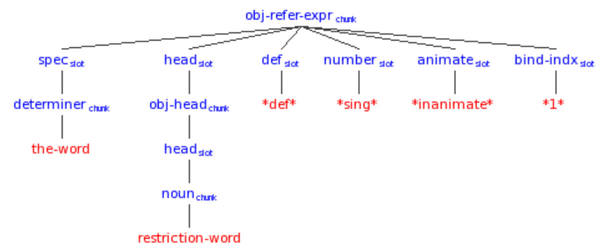

Figure 1. Original nominal structure (including a determiner, projected ORE and object head)

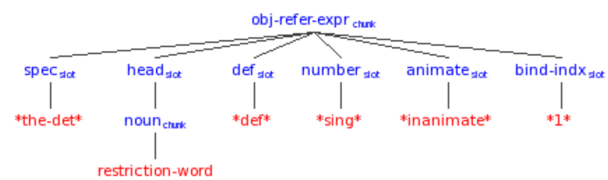

Figure 2. Reduced nominal structure (the retrieved determiner ORE and no object head)

But what happens when a post-head modifier occurs, or when the pre-head modifier slot turns out to be needed? In the processing of the input “the altitude restriction”, when “altitude” is processed it is integrated as the head of the nominal projected from “the”. When “restriction” is subsequently processed there is no expectation for its occurrence. The previous version of the model projected an object head, so “restriction” was accommodated by shifting “altitude” into the pre-head modifier slot so that “restriction” could be integrated as the head. In the current version, we have adopted a similar strategy. In parallel with the integration of “altitude” as the head of the ORE, an object head is constructed in which “altitude” is the head. This object head is available if needed to support subsequent processing. When “restriction” is processed, the object head overrides “altitude” as the head of the ORE and “altitude” is shifted into the pre-head modifier slot so that “restriction” can be integrated as the head (Figure 3). Note that the object head is projected in parallel to facilitate processing. A single production integrates the object head as the head of the ORE, shifts “altitude” to the pre-head modifier slot and integrates “restriction” as the head. It takes no more time to process “restriction” than in the previous version of the model, but it does require parallel projection of the object head.

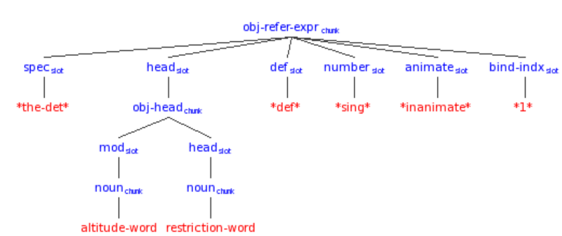

Figure 3. Accommodating “restriction”

Real Time Processing & Spreading Activation

Cognitive time is the time it takes the productions and retrievals in ACT-R to happen, with each production taking a fixed amount of time. When a production fires, 50ms of cognitive time elapses, so having many productions firing for the processing of each word takes up a great deal of cognitive time. Retrievals also take cognitive time—chunks with high activation are retrieved more quickly than chunks with low activation.

Retrievals take *real time* to calculate the activation of all eligible chunks. Real time is the wall clock time that passes while the computer executes the model. When a retrieval request is not very specific, for example, specifying only the chunk-type, then the activation for all chunks of that type must be calculated before the most highly activated chunk can be selected. There are thousands of chunks of type WORD, so when the chunk-type WORD is the only retrieval specification, thousands of activation calculations must be performed before a chunk is retrieved. While this is a parallel process in the brain, it is a serial process for a microprocessor. Since the language model specifies only the chunk type, and relies on spreading activation to retrieve words, thousands of calculations bring the real time reading rate down to 53words per minute (wpm).

Disjunctive Retrieval

One way to retrieve chunks faster in real time is to impose stronger hard constraints on the retrieval. Instead of a weak chunk-type specification that matches thousands of chunks, a strong constraint that matches only a limited set of chunks can be specified. For example, the model could try to retrieve an exact match to input text form, which might only match a single chunk in DM. However, imposing such constraints makes the model less flexible and less cognitively plausible. If the model relies on a hard constraint to match the input form against words in DM, variants cannot be read. Even a hard constraint on just the first letter means that words where the first letter is transposed with the second, or in any other way misplaced, cannot be read by the model.

The model needs the flexibility of a soft-constraint retrieval with the real time speed of a strong hard-constraint retrieval. In order to achieve this affect, we implemented a disjunctive retrieval mechanism. Using an ACT-R function called get-chunk, the model checks DM for the largest constituent of the perceptual span. If it does not find that constituent, it chops the perceptual span at the last punctuation mark or space. If that constituent is not found, it chops at the second to last punctuation mark or space, and so on. If an entire word does not match at any point, a simple soft constraint is attempted.

For example, if the input sentence is “og to h-area”, we want the model to be able to retrieve GO for “og” (see Table 2). The get-chunk function is used to try to find chunks that correspond to smaller and smaller parts of the visual input. If at any point the function finds what it is looking for, the model uses that specification to make the retrieval. Get-chunk is a simple search function into a hash table—it is not computationally expensive, and it functions outside of the cognitive processes of ACT-R, so it does not take any cognitive time.

Table 2. Perceptual span contains “og to h-area”

SEARCH FOR:	RESULT:	RETRIEVAL REQUEST:	RESULT:
og to h-area	NIL	--	--
og to h-	NIL	--	--
og to h	NIL	--	--
og to	NIL	--	--
og	NIL	--	--
--	--	chunk-type WORD	GO-word

Using the disjunctive retrieval, the average reading rate for the model is 249wpm in real time. The cognitive time is unaffected, and the model runs with disjunctive retrieval are identical to the model runs using a pure soft-constraint. The results of retrieval requests are identical. Since the two retrieval methods are equivalent in ACT-R, the disjunctive retrieval is acceptable as a way to make our model fast enough to be functional in real time while we try to catch up in cognitive time.

Conclusions

Although we have not yet succeeded in achieving human reading rates, we have improved the reading rate of the Double-R-Language significantly. The initial version of the model read at a rate of about 96wpm, far from our goal of 200-300wpm, the average reading rate of adults. The model now reads at an average rate of 143wpm in cognitive time, and 249wpm in real time. This rate is the average, achieved while reading a text of just under 2,100 words, without counting punctuation as separate words.

The perceptual span is closer in size to that of human readers than previously. The expanded perceptual span allows for the expansion of the model’s lexicon to include multi-word units, as well as speeding up the reading rate. An additional advantage of multi-word units is that they are less ambiguous than words in compositional phrases.

The model was improved by simplifying various linguistic constructions. Parallel constructions allow for simplified nominal heads, and object referring expressions in declarative memory allow the model to avoid constructing object referring expressions whenever determiners are encountered. We posit that the simplified representations are not only more expedient, but more cognitively plausible as well. Avoiding unnecessary constructions in the model is more likely to track the efficiency of human language use.

Ultimately, we believe that achieving human level reading rates will require a capability to recognize multi-word units that exceed a single perceptual span. Recognition of a linguistic unit as forming a part of a larger linguistic unit across perceptual spans should minimize the amount of higher level processing required to integrate the recognized unit into the evolving representation and speed up the reading rate, allowing the model to approach adult human reading rates.

Although reading rate is important, the language comprehension model is being developed to model the full range of linguistic processes of a competent adult reader, rather than just modeling the reading rate. It is our hope that any improvements we make in the reading rate of our model will be accompanied by improvements in the models accuracy and cognitive plausibility.

References

Anderson, J. R. (1983). *The Architecture of Cognition*. NY: Harvard.

Anderson, J. R. (2007). *How Can the Human Mind occur in the Physical Universe?* NY: Oxford.

Ball, J. (2007a). Construction-driven language processing. *Proceedings of the 2nd European Cognitive Science Conference*, 722-727. NY: LEA

Ball, J. (2007b). A Bi-Polar Theory of Nominal and Clause Structure and Function. *Annual Review of Cognitive Linguistics*, 27-54. Amsterdam: John Benjamins

Ball, J. (2010). Projecting grammatical feature in nominals: cognitive processing theory & computational implementation. *Proceedings of the 19th Behavior Representation in Modeling & Simulation Conference.*

Ball, J., Heiberg, A. & Silber, R. (2007). Toward a large-scale model of language comprehension in ACT-R 6. *Proceedings of the 8th International Conference on Cognitive Modeling*, pp. 163-168.

Ball, J., Myers, C. W., Heiberg, A., Cooke, N. J., Matessa, M., & Freiman, M. (2009). The Synthetic Teammate Project. *Proceedings of the 18th Annual Conference on Behavior Representation in Modeling and Simulation*. Sundance, UT.

Balota, D. A., Pollatsek, A., & Rayner, K. (1985). The interaction of contextual constraints and parafoveal visual information in reading. *Cognitive Psychology*, *17*, 364-390.

Carver, R.P. (1973a). Understanding, information processing and learning from prose materials. *Journal of Educational Psychology*, 64, 76-84.

Carver, R. P. (1973b). Effect of increasing the rate of speech presentation upon comprehension. *Journal of Educational Psychology*, *65*, 118-126.

Cook, V. J, & Newson, M. (1996). *Chomsky's Universal Grammar*. Malden, MA: Blackwell Publishers.

Douglass, S., Ball, J. & Rodgers, S. (2009). Large declarative memories in ACT-R. *Proceedings of the 9th International Conference on Cognitive Modeling*.

DenBuurman, R., Boersma, T., & Gerrissen, J. F. (1981). Eye movements and the perceptual span in reading. *Reading Research Quarterly*, *16*, 227-235.

Freiman, M. & Ball, J. (2008). Computational cognitive modeling of reading comprehension at the word level. *Proceedings of the 38th Western Conference on Linguistics*, 34-45. Davis, CA: University of California, Davis.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception. Part I. An account of basic findings. *Psychological Review*, *88*, 375–407.

McConkie, G. W., & Rayner, K. (1975). The span of the effective stimulus during a fixation in reading. *Perception & Psychophysics, 17*. 578-586.

Paap, K., Newsome, S., McDonald, J., & Schvaneveldt, R. (1982). An activation-verification model of letter and word recognition: The Word-Superiority Effect. *Psychological Review*, *89*, 573-594.

Perea, M., & Lupker, S. J. (2003). Does jugde activate COURT? Transposed-letter similarity effects in masked associative priming. *Memory and Cognition*, *31*, 829–841.

Rayner, K. (1986). Eye movements and the perceptual span in beginning and skilled readers. *Journal of Experimental Child Psychology, 41*, 211-236.

Sag, I., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96*, No. 4, 523-568.

Taylor, S. E. (1965). Eye movements while reading: Facts and fallacies. *American Educational Research Journal*, 2, 187-202.

Zipf, George Kingsley (1932). *Selected Studies of the Principle of Relative Frequency in Language.* Cambridge, MA: Harvard University Press.

Zipf, G. K. (1949)**.** *Human behavior and the principle of least effort: An introduction to human ecology*. Cambridge, MA: Addison-Wesley, 1949..

Zwann, R. and Radvansky, G. (1998). Situation models in language comprehension and memory. *Psychological Bulletin*, *123*(2), 162-185.