Explorations in ACT-R Based Language Analysis – Memory Chunk Activation, Retrieval and Verification without Inhibition

Jerry T. Ball (Jerry.Ball@wpafb.af.mil)
Air Force Research Laboratory, 711th Human Performance Wing
2620 Q Street, Bldg 852, WPAFB, OH 45433

Abstract

This paper explores the benefits and challenges of using the ACT-R cognitive architecture in the development of a large-scale, functional, cognitively motivated language analysis model. The paper focuses on ACT-R's declarative memory retrieval mechanism, proposing extensions to support verification of retrieved chunks, multi-level activation spread and carry over activation. The paper argues against the need for inhibition between competing chunks which is necessarily task specific.

Keywords: ACT-R; activation; retrieval; verification; inhibition; language analysis.

Introduction[1]

Our team has been working on the research and development of a language analysis model (Ball, 2011a; Ball, Heiberg & Silber, 2007) within the ACT-R cognitive architecture (Anderson, 2007) since 2002 (Ball, 2003). Currently, the model comprises ~950 productions and over 58,000 declarative memory (DM) chunks. The model is capable of processing a broad range of English language constructions (www.doublertheory.com/ comp-grammer/ comp-grammar.htm; Ball, Heiberg & Silber, 2007) and is a component of a larger synthetic teammate model (Ball et al., 2010). The model accepts written input from single words to entire documents, and processes the input incrementally, one word or multi-word unit at a time. On a 64-bit quad-core Windows machine with 8 Gig RAM, the model incrementally processes ~130 words per minute (wpm) with the full 58,000 chunk mental lexicon and ~320 wpm with a smaller 22,000 chunk mental lexicon. The model processes ~145 wpm in ACT-R cognitive processing time which compares to adult reading rates ranging from 200-300 wpm. We are working on ways to improve the analysis rate of the model—which does not entail full comprehension—to bring it into closer alignment with adult reading rates (Freiman & Ball, 2010).

Our focus is on research and development of a general-purpose, large-scale, functional model that adheres to well established cognitive constraints on human language processing (HLP) (Ball et al., 2010). Two important constraints that we adhere to are *incremental* and *interactive* processing (Just & Carpenter, 1987; Altmann & Steedman, 1988; Tanenhaus et al., 1995; Gibson & Pearlmutter, 1998). Adherence to these constraints precludes the use of computational techniques like algorithmic backtracking and staged analysis (i.e., independent tokenizing, part of speech tagging, syntactic analysis, semantic analysis, and pragmatic analysis) and limits the use of techniques like lookahead, underspecification and parallel propagation of constructed alternatives—all of which are mainstays of many computational linguistic systems.

ACT-R incorporates two architectural constraints, realized as serial bottlenecks, which largely determine incremental processing: 1) a single production can execute at a time, and 2) a single DM chunk can be retrieved at a time. In addition to these serial constraints which are the basis of *incremental* processing, ACT-R provides architectural support for parallel processing in the form of a parallel production selection mechanism based on utility, and a parallel DM retrieval mechanism based on activation. These parallel mechanisms are probabilistic and context dependent. The parallel/probabilistic/context dependent mechanisms provide the basis for *interactive* processing. They guide the processing of the language analysis model in directions that are likely to lead to a successful analysis given the current context and current input. The highly parallel retrieval mechanism is capable of selecting from existing DM chunks, but does not build any structure. The serial integration mechanism is responsible for building new structures, but is constrained to maintaining a small number of constructed representations, in parallel, in working memory which is composed of ACT-R buffers supplemented with specialized language analysis buffers (Ball, 2011b).

Cognitive processing in ACT-R revolves around the selection and execution of a sequence of productions. The production with the highest utility that matches the current context provided by the ACT-R/language analysis buffers, is selected for execution. Production execution can result in a perceptual-motor action (e.g. visual attention shift, mouse movement), a modification to the contents of a buffer, or a DM retrieval. These actions change the context for selection and execution of the next production.

When the executing production invokes a DM retrieval, the parallel spread of activation from chunks in buffers to associated chunks in DM (*soft constraints* or biases) combines with the base level activation—based on prior history of use of the chunk—to determine total chunk activation. The single most highly activated chunk which matches a retrieval template (*hard constraint*) specified by

[1] Thanks to Dan Bothell for pointing out several misconceptions about ACT-R in an earlier version of this paper

the executing production is retrieved. Chunks can either be associated by sharing a slot value or by explicit specification of an association using the **`add-sji`** function. For activation to spread, the activating chunk must be in a buffer (matching slot value) or in a slot in a chunk in a buffer (explicit specification via **`add-sji`**).

The language analysis model makes extensive use of ACT-R's serial and parallel processing mechanisms. The model processes the linguistic input incrementally, one word or multi-word unit at a time, and uses all available information interactively to make the best choice at each choice point. The model also relies on a non-monotonic mechanism of *context accommodation* which is capable of making modest adjustments to the evolving representation when the current input, in combination with the current context, indicates the need for such accommodation. Context accommodation is part of normal processing—in the right context, a production capable of accommodating the input executes. For example, in incrementally processing "the airspeed restriction", when "airspeed" is processed, it is integrated as the head of the noun phrase projected during the processing of "the", but when "restriction" is subsequently processed, the model accommodates "restriction" by shifting "airspeed" into a modifier function and making "restriction" the head. Context accommodation is not capable of handling the kinds of disruptive garden path sentences that are a mainstay of psycholinguistic research (e.g. Bever's (1970) famous "the horse raced past the barn fell"). Such inputs require reanalysis mechanisms which have not yet been implemented. The focus of model development is on handling common English—inputs which humans process with ease, but which, nonetheless, present significant modeling challenges due to ambiguity. The combination of parallel/probabilistic/context dependent processing, and serial processing with context accommodation allows the model to pursue the single best analysis, but to adjust the analysis without backtracking or reanalysis, when needed. The overall result is a *pseudo-deterministic* HLP which presents the appearance and efficiency of deterministic processing, despite the rampant ambiguity which makes truly deterministic processing impossible (Ball, 2011a).

Activation

In ACT-R, all DM chunks have an activation level which depends on the current context (source activation) and prior history of use (base level activation) of the chunk. A key assumption is that the current context is captured in the contents of the ACT-R buffers which are sources of activation. The most basic form of the activation equation (ignoring partial matching which we do not use, and noise) is shown below where A_i = total activation of chunk i; B_i = base level activation of chunk i, and S_i = spreading activation contribution to activation of chunk i:

$$A_i = B_i + S_i$$

The base level activation is a logarithmic function of the number of uses of a chunk over time combined with a negative exponential decay mechanism (assuming the default, optimized base level equation). Spreading activation is a weighted sum of activations from all the sources of activation in buffers which match the slot values of the chunk being activated or for which an explicit association has been specified (via **`add-sji`**). The amount of spreading activation to a chunk from each source decreases with the number of competing chunks which match the source. This proportional spreading activation is known as the *fan effect*. The fan effect does not apply to chunks for which an explicit association has been specified.

The language analysis model makes extensive use of ACT-R's retrieval (activation and selection) mechanism. In the word recognition subcomponent, a perceptual span which encodes the visual contents of the current attention fixation spreads activation to DM and the word or multi-word unit which is most highly activated is retrieved (selected) and compared to the perceptual input. If the comparison is close enough, the retrieved word or multi-word unit is considered a match. Overall, the process involves four steps: 1) perceptual encoding of the input (*encoding*); 2) activation of declarative memory (*activation*); 3) retrieval of the most highly activated DM chunk which matches the hard constraints of the retrieval template (*selection*); and 4) comparison of the retrieved memory element against the perceptual input (*verification*). Completion of all but the third step presents challenges for ACT-R based modeling.

ACT-R's built in perceptual encoding mechanism assumes words are divided into units by spaces and automatically separates punctuation into separate perceptual units. While this typically succeeds in identifying words and punctuation, it often does not. There are words like "etc." and "didn't" which incorporate punctuation and there are words like "a priori" and "none the less" which have spaces. In addition, the model includes multi-word units like "have been", "get out" and "New York" which are encoded in the mental lexicon as lexical items. Higher level knowledge from the mental lexicon is needed to decide what constitutes a word or multi-word unit. To support the integration of higher level knowledge with perceptual processing, we modified ACT-R's perceptual encoding mechanism to incorporate a perceptual span that does not automatically segment the input at spaces and punctuation (Freiman & Ball, 2010). Not only does the perceptual span mechanism support the integration of higher level knowledge, it speeds up processing significantly since words like "don't" are recognized as a single unit instead of three separate units "don", "'" and "t". Likewise, multi-word units like "get out" are also recognized as a unit. Besides speeding up processing, the recognition of multi-word units reduces ambiguity significantly. The word "take" is extremely ambiguous, whereas multi-word units like "take out", "take off" and "take in" are far less ambiguous. The ability to recognize multi-word expressions is an important tool for handling the ambiguity of natural language and for speeding up the model. We view the addition of multi-word expressions as the best way of achieving adult reading rates.

With a mental lexicon near 58,000 lexical items, the computation of activation presents a serious computational challenge. It is not possible to compute the activations of 58,000 lexical items prior to each retrieval, in real-time, on existing hardware. (This is also the reason we are unable to use the partial matching subsystem, since all DM chunks are candidates for retrieval when partial matching is enabled.) As a workaround, we developed a capability to minimize the activation computations in the event of an exact match to the form of the input. If there is a lexical item in DM which is an exact match to the perceptual span, a hard constraint is added to the retrieval template to restrict the number of matching DM elements. When the full perceptual span doesn't match, the match is backed off to the last space in the perceptual span and re-attempted. Prior spaces can also be backed off to. If there is no match (e.g. if the input is "spped")—as a computational compromise—the model attempts a retrieval requiring a hard constraint match on the first letter in the perceptual span. We call this mechanism a *disjunctive retrieval* capability. Except for this last compromise, the disjunctive retrieval capability retrieves the same lexical item as a soft constraint retrieval. Even with this last compromise, computation of activations is slower than real-time in the worst case where only a first letter match is required, since there may be thousands of matching lexical items whose activation must be computed. We are looking for ways to improve processing with minimal compromise compared to the preferred soft constraint retrieval mechanism.

The verification step is also problematic from an ACT-R modeling perspective. ACT-R does not provide the kind of low level perceptual matching capability that is needed to implement this step. Instead, we have incorporated the Levenshtein Distance algorithm to perform this comparison. We view verification as a key element of the word recognition mechanism in accord with the Activation-Verification model of Paap et al. (1987) and in contrast to the Interactive-Activation model of McClelland & Rumelhart (1985) which has no verification stage. Verification is crucial for identifying novel inputs. A novel input is one that is not a close match to any chunk in memory, although exactly what constitutes a "close match" is an open research question.

Multi-Level Activation Spread

In ACT-R, activation spreads from the slots in chunks in buffers to chunks in DM with matching slot values or explicitly set associations (using `add-sji`). For example, we have a context buffer that encodes information about the context that has a slot named "gram-pos-bias" (grammatical part of speech bias). Following the processing of a word like "the" (a determiner), this slot will be set to the chunk `noun`. During the retrieval of a lexical item, the `noun` chunk will spread activation to all lexical items with a matching `noun` chunk (i.e. all nouns). If the word "point" follows "the", this bias will spread activation to the `noun` chunk for "point" as opposed to the `verb` chunk (i.e. "to point"). In this way the grammatical context biases the selection of the part of speech (POS) of a word during retrieval.

There is no mechanism in ACT-R to spread activation from slots in chunks in DM to other chunks in DM with matching slot values or to explicitly associated chunks. Once activation spreads from slots in buffers to DM chunks during a retrieval, activation spread stops and the final activation is computed to determine which DM chunk to retrieve. We refer to this as *single level activation spread*.

Our model assumes that there are DM chunks which encode both the form of a word (e.g. "speed", "speeds") and POS (e.g. "noun", "verb"). Originally, word form and POS information were encoded in distinct `word-form` and `pos` chunks. The model first retrieved a `word-form` chunk given the letters and trigrams in the input, then retrieved a `pos` chunk for the word form. In order to improve the analysis rate of the model (Freiman & Ball, 2010), word form and part of speech information was combined into a single `word-pos` chunk (i.e. word form + part of speech). While we were successful in eliminating a retrieval, the resulting `word-pos` chunks contain a mixture of word form information (e.g. the letters and trigrams in the word) and POS information (e.g. noun, verb, as well as grammatical features like number, animacy and gender for nouns, and tense and aspect for verbs). Note that this mixture of word form and POS information makes it possible to capture the interaction of word form and POS with single level activation spread. For example, retrieval of the POS for "speed" (i.e. noun or verb) given the input "spped" depends on the biasing context (e.g. noun bias following "the", verb bias following "to") as well as the letters and trigrams. However, the `word-pos` chunks do not (yet) contain any representation of phonetic, phonemic, syllabic or morphemic information. With just letter, trigram and POS information, `word-pos` chunks contain many slots. Adding phonetic, phonemic, syllabic and morphemic information will increase the number of slots substantially. Ideally, we would like to represent letter, trigram, POS, phonetic, syllabic etc. information independently of each other in separate chunks—allowing them to interact in retrieving a word (or letter, or POS, or phoneme), but given the single-level activation spreading mechanism in ACT-R, there is no way to capture the interaction without including all the linguistic knowledge in a single chunk or using `add-sji` to establish links between chunks and retrieving and retaining chunks at all levels in buffers to spread activation.

A negative consequence of the integration of word form and POS information is the need to redundantly encode information. If a given word form is associated with multiple POSs, then multiple `word-pos` chunks are needed. For example, the word "speed" can be both a noun and a verb. To represent this, two `word-pos` chunks are needed, one for the noun and one for the verb. In these two `word-pos` chunks, the letters and trigrams in the word form are redundantly encoded. To minimize redundancy, it is desirable to factor out word form and POS knowledge. But to model adult human reading rates, it is important to minimize the number of retrievals. Both can be accomplished with multi-level activation spread. For example, if the goal is to retrieve a POS, the letters and trigrams in the input can spread activation to a **`word-form`** chunk which can spread activation to a **`pos`** chunk. Then the **`pos`** chunk can be retrieved without first retrieving the **`word-form`** chunk.

We are in the process of mapping the linguistic representations that are generated by our language analysis model into a situation model based semantic representation. We are trying to do this in a representationally reasonable way within ACT-R. The problem we face is the many-to-many mapping between words and concepts. Individual words may map to multiple concepts (river "bank" vs. financial "bank"), and individual concepts may map to multiple words ("dog" vs. "canine"). Given this many-to-many mapping, we would like to use **`mapping`** chunks to map from words to concepts. The **`mapping`** chunks would encode a single mapping relationship (e.g. a separate mapping chunk to map from the word "bank" to the financial institution concept; from the word "bank" to the river bank concept; from the concept dog to the word "dog"; from the concept dog to the word "canine"). When processing a word, a key goal is to retrieve the contextually relevant concept. We would like to accomplish this with a minimum number of retrievals since our model is already slower than adult humans even without the mapping into concepts. Since there is no direct link between a word and a concept if **`mapping`** chunks are used (i.e. there is no slot in the **`concept`** chunk that contains the word), the word will not spread activation to the concept. Instead, given the use of **`mapping`** chunks, two retrievals are needed: 1) given a **`word-pos`** chunk, retrieve a **`mapping`** chunk, and 2) given a **`mapping`** chunk, retrieve a **`concept`** chunk. The use of **`mapping`** chunks can be eliminated is we use the **`add-sji`** function to establish direct links between **`word-pos`** chunks and **`concept`** chunks. We are currently pursuing this option to avoid the need to retrieve an intermediate **`mapping`** chunk. Even with explicit links from **`word-pos`** chunks to **`concept`** chunks, a **`word-pos`** chunk must first be retrieved to spread activation to associated **`concept`** chunks. With multi-level activation spread it would be possible to directly retrieve a **`concept`** chunk, eliminating the need to retrieve a **`word-pos`** chunk.

Alternatively, if we were to combine **`concept`** chunks with **`word-pos`** chunks, then a single retrieval could be used to retrieve a **`word-pos-concept`** chunk. However, there may be multiple concepts associated with a **`word-pos`** chunk (e.g. "river *bank*" vs. "financial *bank*"). If we create separate **`word-pos-concept`** chunks for each alternative, the amount of redundancy is increased again. Further, it is questionable whether letter and trigram information should be directly associated with (non-linguistic) concepts.

The main advantage of creating **`word-pos-concept`** chunks is the reduction in the number of retrievals needed to go from the input to a concept. To see how problematic retrievals are for models of reading, consider the E-Z Reader model (Reichle, Warren & McConnell, 2009), a model of eye movements in reading which models lexical processing (not reading). E-Z Reader allows just 25 msec per word beyond lexical access for post-lexical processing to influence lexical processing. According to the authors, 25 msec is "the minimal amount of post-lexical processing that (on average) is necessary to satisfy the language-processing system that comprehension is proceeding without difficulty and that it is not necessary to interrupt lexical processing and/or halt the progression of the eyes" (ibid., p. 6). Since it requires 50 msec to execute a production in ACT-R which attempts a retrieval, plus the retrieval time, there would be insufficient time for a single retrieval in an ACT-R implementation of E-Z Reader to influence lexical processing and eye movements! However, the E-Z Reader model makes the simplifying assumption that words are space delimited and since our model is capable of recognizing multi-word units, the 25 msec limit can be relaxed somewhat. But there is still insufficient time for more than 1 or 2 retrievals (on average) beyond the retrieval needed to support word recognition itself.

Carry Over Activation and Resonance

Activations are computed in ACT-R as part of a retrieval attempt. The activation computation involves combining the base level activation and activation spread from all buffers which are sources of activation. The logarithmic nature of the default, optimized base level activation computation means that the base level of overused DM chunks does not vary much from use to use (i.e. the base level activation has reached asymptote). Words constitute very highly used DM chunks. Using estimates of the number of occurrences of a word over a lifetime results in a base level activation that varies little from use to use and decays very slowly. Since the spread of activation is computed independently on each retrieval, for a word that has been used recently, there is no contextual indication of this prior use (i.e. the base level hasn't significantly changed and any prior spread of activation is not retained). Yet there is clear evidence of priming effects from prior uses of words. According to Dan Bothell (p.c.), this is a limitation of ACT-R's default optimized base level equation. We are exploring use of a hybrid version of the base level equation which does not suffer this limitation (Bothell, 2011, p. 213). Use of the non-optimized base-level equation is not possible given the size of our mental lexicon and the large number of word uses.

Another alternative is to retain the word in a buffer so that activation can continue to spread from the word to corresponding chunks in DM. We have tried this approach in the case of idiom processing. To see the basic challenge, consider the processing of idioms like "kicked the bucket" and verb-particle combinations like "pick…up" as in "pick the ball up". We assume that idioms and verb-particles correspond to distinct chunks (i.e. multi-word units) in DM. These multi-word expressions exceed the size of the perceptual span and cannot be recognized in a single attention fixation. Instead, the model must somehow recognize the idiom "kicked the bucket" when the word "bucket" is processed and the verb-particle combination "pick…up" when the word "up" is processed. If there is no evidence that "kicked" has occurred at the processing of "bucket", then there is no way for the model to retrieve "kicked the bucket" instead of "bucket". Similarly for "pick" when "up" is processed. Since the DM element "bucket" is an exact match to "bucket" and "bucket" has a higher base frequency than "kicked the bucket" (i.e. single words have a higher base frequency than multi-word units containing them), there must be some mechanism for preferring "kicked the bucket" in this context. "Kicked" and "the" could be retained in the context to spread activation to "kicked the bucket" to handle this example, but, in general, this would mean retaining an arbitrary number of words in the context to spread activation. In the case of "pick…up", "pick" would need to be retained in a buffer for an indefinite period of time (e.g. "*pick* the big red ball *up*", "*pick* the ball that is on the table *up*").

Even with the hybrid base-level equation, relying on an increase in base level for "kicked the bucket" will not work in this example. Since the processing of "kicked" is likely to retrieve "kicked" and not "kicked the bucket", the base level activation of "kicked the bucket" will be unaffected at the processing of "kicked" (i.e. the "kicked the bucket" chunk must be retrieved and merged back into DM to constitute a use). Further, any temporary spreading activation from "kicked" to "kicked the bucket" will have been lost at the processing of "bucket".

A possible solution is to introduce a carry-over activation capability. When "kicked" is processed it will spread activation to "kicked the bucket" as well as "kicked". Despite the fact that "kicked the bucket" is not retrieved, some of this activation will carry-over so that when "bucket" is processed, "kicked the bucket" will receive activation from "bucket" as well as carry-over activation from "kicked" and "the". The combination of carry-over activation from "kicked" and "the", plus the activation from "bucket" should allow "kicked the bucket" to be retrieved in this context. In general, this seems like a better solution than trying to retain "kicked" and "the" in the context when "bucket" is processed. In the case of "pick…up", carry over activation should handle cases where the gap between "pick" and "up" is small (e.g. "*pick* the ball *up*"), but cause problems when the gap is large enough that any carry over activation will have decayed. This result might explain the preference for placing the particle before the object when the description of the object is long (e.g. "*pick up* the big red ball on the table" is preferred over "*pick* the big red ball on the table *up*").

There are additional reasons for suggesting the introduction of carry-over activation. Carry-over activation corresponds to a short-term increase in the activation of a DM chunk that extends beyond the execution of a single chunk retrieval. This carry-over activation (i.e. neuron spiking) differs from increases in base level activation which we view as more permanent changes in long-term potentiation. The introduction of carry-over activation combined with multi-level activation spread, could support an ART-like adaptive resonance capability (Grossberg, 1987)—although it is unclear how this could be done in a computationally tractable way. Note that ART uses resonance to distinguish novel from previously experienced inputs—previous inputs lead to resonance with memory whereas novel inputs do not. With carry-over activation, the verification stage of the word recognition subcomponent could be implemented within the architecture via resonance instead of using the Levenshtein Distance metric outside the architecture.

Inhibition

Inhibition is a winner-take-all mechanism that is commonly used in connectionist architectures to allow a network of nodes to settle into a solution (cf. McClelland & Rumelhart, 1981; Kintsch, 1998). Over time, the most active node or co-activating nodes inhibit competing nodes. There is no equivalent in ACT-R—although it is possible to get inhibitory effects by explicitly setting the strength of association between two or more chunks to a negative value. But even here, there is no notion of settling into a solution in ACT-R.

The need for inhibition as a mechanism for settling into a solution is obviated in ACT-R by the retrieval mechanism which results in selection of the single most highly activated chunk matching the retrieval template. This is ACT-R's equivalent of a "winner-take-all" network. The retrieval mechanism picks out the most highly activated chunk which matches the hard constraints of the retrieval template.

ACT-R's spreading activation mechanism doesn't bias a model to any particular task. The same cannot be said of inhibition. For any inhibitory network, it is possible to define conflicting tasks that the inhibitory network cannot perform. For example, if both singular and plural forms of nouns (e.g. "child" and "children") occur in a network, should they inhibit each other? It depends on the task. If the task is a lexical decision task, then we want "child" to inhibit "children" and vice versa, so that they don't interfere (i.e. if "child" is the winner when the input is "children", presumably the lexical decision response would be negative since "child" doesn't match the input). On the other hand, if the task is to generate the singular form of the word in response to the plural word, or the plural in response to the singular, than we need facilitation rather than inhibition. As another example, consider the verbs "go" and "went". For a lexical decision task, these words should inhibit each other. But for a task of generating the past tense of "go", they should facilitate each other.

Not only are inhibitory links task specific, but in a large declarative memory, the number of such links will be explosive. If a word must inhibit all its competitors, with a 58,000 word lexicon, the number of inhibitory links is computationally explosive. Inhibitory links don't scale.

Besides the task specificity of inhibitory networks, most inhibitory networks assume well-defined levels with inhibitory links typically constrained to occurring within a level and excitatory links occurring across levels. The recognition of multi-word expressions like "get up", inflected words like "books" and morphologically complex words like "progressivity" present a challenge for such networks? Are multi-word expressions (e.g. "get up"), inflected words (e.g. "books"), morphologically complex words (e.g. "progressivity"), morphologically simple words (e.g. "cat") and morphemes (e.g. plural "s") represented on the same level, in which case they compete, or on different levels, in which case they co-activate each other? It depends on the task. If "get", "up" and "get up" are all represented on the same *lexical item* level where they inhibit each other, how do we recognize "get up" as a lexical unit (i.e. how does the model settle in to "get up")? Even if multi-word expressions are represented on a different level from words, the words in the multi-word expression will inhibit each other, making it difficult to distinguish the multi-word expression from the word which wins the word level competition, unless the task is specifically to recognize multi-word expressions. Similar questions arise for inflected, morphologically complex, and morphologically simple words, and morphemes. In short, multi-word expressions, inflected words, morphologically complex words and morphemes call into question the typical assumption that there is a well defined word level. In a model restricted to four letter words without inflectional variants where the task is word recognition (e.g. McClelland & Rumelhart, 1981), well-defined levels can be established. When we consider real language, there is no well-defined word level with inhibitory links that is task independent.

In sum, inhibition is not a viable alternative to ACT-R's task general spreading activation mechanism combined with a winner-take-all retrieval mechanism that depends on the current task.

Conclusions

The use of ACT-R for language analysis provides several benefits. ACT-R solves the problems of how to integrate symbolic and probabilistic processing combined with serial and parallel processing in an effective and elegant manner. For the most part, the capabilities provided by ACT-R have proved useful for the development of our language analysis model, and much of the success of our model is attributable to the capabilities and constraints of ACT-R.

However, there is room for improvement of ACT-R. Interestingly, the suggestions presented in this paper are consistent with Anderson's seminal paper on spreading activation (Anderson, 1983a) and the ACT* architecture (Anderson, 1983b). Multi-level activation spread is capable of spreading activation to indirectly related DM chunks, obviating the need to specify indirect links or to add slots to directly model the associations, and keeping the capability to model interactions and minimize retrievals. Carryover activation allows the effects of multi-level spreading activation to be retained until needed. Of course, the computational costs of multi-level activation spread and carry-over activation are potentially explosive and it will be a challenge to figure out how to extend ACT-R's capabilities in the ways suggested in this paper in a computationally tractable manner.

References

Altmann, G., & Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, *30*, 191-238.

Anderson, J. (1983a). A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior*, 22, 261-295.

Anderson, J. (1983b). *The Architecture of Cogntion*. Cambridge, MA: Harvard University Press.

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

Ball, J. (2003). Beginnings of a Language Comprehension Module in ACT-R 5.0. *Proceedings of the Fifth International Conference on Cognitive Modeling*. F. Detje, D. Doerner & H. Schaub (eds). Univ-Verlag Bamberg. ISBN 3-933463-15-7.

Ball, J. (2011a). A Pseudo-Deterministic Model of Human Language Processing. *Proceedings of the 2011 Cognitive Science Society Conference*.

Ball, J (2011b). Explorations in ACT-R Based Cognitive Modeling – Chunks, Inheritance, Production Matching and Memory in Language Analysis. *Proceedings of the AAAI Fall Symposium: Advances in Cognitive Systems*, 10-17. Arlington, VA: AAAI.

Ball, J., Freiman, M., Rodgers, S. & Myers, C. (2010). Toward a Functional Model of Human Language Processing. *Proceedings of the 32nd Annual Meeting of the Cognitive Science Society*.

Ball, J., Heiberg, A. & Silber, R. (2007). Toward a Large-Scale Model of Language Comprehension in ACT-R 6. In R. Lewis, T. Polk & J. Laird (Eds.) *Proceedings of the 8th International Conference on Cognitive Modeling*. 173-179. NY: Psych Press.

Bever, T. (1970). The cognitive basis for linguistic structures. In J. Hayes (ed.), *Cognition and the development of language*, 279-362. NY: Wiley.

Bothell, D. (2011). ACT-R 6.0 Reference Manual. Downloaded from http://act-r.psy.cmu.edu/actr6/reference-manual.pdf

Freiman, M. & Ball, J. (2010). Improving the Reading Rate of Double-R-Language. In D. D. Salvucci & G. Gunzelmann (Eds.), *Proceedings of the 10th International Conference on Cognitive Modeling* (pp. 1-6). Phil, PA: Drexel University.

Gibson, E., & Pearlmutter, N. (1998). Constraints on sentence comprehension. *Trends in Cognitive Sciences*, *2(*7), 262-268.

Grossberg, S. (1987). Competitive Learning: from interactive activation to adaptive resonance. *Cognitive Science*, 22, 23-63.

Just, M. & Carpenter, P. (1987). *The Psychology of Reading and Language Comprehension*. Boston: Allyn and Bacon, Inc.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. NY: Cambridge University Press.

McClelland, J., & Rumelhart, D. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*(5), 375-407.

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.

Reichle, E., Warren, T. & McConnell, K. (2009). Using E-Z Reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review*, 16(1), 1-21.

Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., & Sedivy, J. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science*, *268*,1632-1634.