Semantic Representations in Working Memory: A Computational Model

Benjamin Kowialiewski[1,2,3], Klaus Oberauer[3]

[1]Psychology & Neuroscience of Cognition Research Unit (PsyNCog), University of Liège, Belgium

[2]Fund for Scientific Research F.R.S.-FNRS, Brussels, Belgium

[3]Department of Psychology, University of Zurich, Switzerland

The research reported in this article was supported by funding from the Swiss National Science Foundation (SNSF, grant number 100014_179002).

Correspondence concerning this article should be addressed to Benjamin Kowialiewski, Psychology & Neuroscience of Cognition Research Unit, 4000 Liège, Belgium. E-mail: bkowialiewski@uliege.be

Open Science statement:

All the data and codes have been made available on the Open Science Framework: https://osf.io/6j7tv/

Abstract

Verbal Working Memory (WM) is supported by semantic knowledge. One manifestation of this is the rich pattern of semantic similarity effects found in immediate serial recall tasks. These effects differ from the effects of similarity on other dimensions (e.g., phonological similarity), which renders them difficult to explain. We propose a comprehensive mechanistic explanation of semantic similarity effects by extending a standard connectionist architecture for modeling immediate serial recall to incorporate semantic representations. Central to our proposal is the selective encoding of categorical features shared among multiple list items. The selective encoding of shared semantic features is made possible via a tagging mechanism that enables the model to encode shared feature retrospectively. Through this mechanism, our model accounts for the majority of semantic similarity effects. Our results imply that working memory represents semantic information in a more restricted way than phonological information.

Keywords: Working Memory; Serial Recall; Semantics; Semantic similarity; Computational modeling

Introduction

Working memory (WM) is supported by semantic knowledge (Patterson et al., 1994; Poirier & Saint-Aubin, 1995; Romani et al., 2008). This influence has been mostly observed in immediate serial recall, requiring people to recall short lists of words in their presentation order. One of the most robust and replicated phenomena showing the influence of semantic knowledge is the *semantic similarity* effect (Poirier & Saint-Aubin, 1995): Performance is higher for lists composed of semantically similar vs. dissimilar words. This semantic similarity effect has been observed in a rich variety of experimental conditions, implying that WM uses semantic knowledge in some way. Therefore, it is important to understand how WM interacts with meaning.

The present study presents a connectionist architecture of WM to account for semantic similarity effects observed across a diverse range of experimental paradigms. The core WM architecture uses generic principles shared by many successful models of serial recall. We assume that memory sets are encoded into WM through bindings between items and positional contexts, a general principle shared by many WM models (Burgess & Hitch, 1999, 2006; Henson, 1998; Lewandowsky & Farrell, 2008; Oberauer et al., 2012; Oberauer & Lewandowsky, 2011). The novelty of our approach is to integrate meaning in this architecture. We start this study by a literature review introducing the way semantic similarity impacts WM performance in different experimental conditions, focusing on well-replicated phenomena that will be used as benchmarks for our simulation work. Next, we describe the core principles of the WM architecture and the way it is implemented. We then present the results of the simulations.

Dissociating Item and Order Recall

In typical WM experiments, participants are required to recall items as well as their presentation order. Throughout this manuscript, we will continuously draw the distinction between people's ability to remember items, and their ability to remember the order in which they appeared, independently of each other. We will refer to these different ways of measuring memory performance as *item recall* and *order recall*. We intend these terms to be descriptive, without any commitment to a theoretical distinction of memory mechanisms underlying item and order recall. Failures to recall the items and failures to recall their presentation order merely reflect different kinds of errors. This distinction is important, as semantic similarity improves item recall, and leaves order recall unaffected (cf. sections below). We therefore need to understand how item and order errors can be measured separately. In immediate serial recall, item recall is assessed using the proportion of list items recalled, regardless of where the items have been recalled. For instance, given the input sequence *ABCDEF* and the output sequence *A*DC*Z* (where the character "*" represents an omission), item recall is equal to 3/6. Order recall is computed by dividing the number of items recalled in their correct position by the total number of items recalled regardless of their output position (Saint-Aubin & Poirier, 1999b). Using the previous example, order recall is equal to 1/3. Proportionalizing by the number of items recalled is important, as it provides an estimate of the conditional probability of committing an order error for an item, given that the item's identity is recalled. As people remember more similar than dissimilar items, they are expected to recall more items in the correct order *and* incorrect order in absolute terms, even if the probability to recall each item in its correct position is equivalent. This way of computing order recall implies that items not recalled at all are treated as missing data as they are not diagnostic for order recall.

Another way to estimate order recall separately from item recall is by using an order reconstruction task, in which the memoranda are made available at retrieval, and participants are asked to reconstruct the original order of the sequence. As all items are available during both encoding and retrieval, item errors are impossible in order reconstruction. As a further way to estimate order recall, researchers often sample items of memory lists from a closed pool of stimuli repeatedly, so that participants soon know all items in the pool perfectly (Neath & Surprenant, 2019; Saint-Aubin & Poirier, 1999a).

Category Membership as a Semantic Similarity Metric

A simple way to manipulate semantic similarity involves the use of taxonomic categories (e.g., fruits, animals, shapes, birds…). The similar lists are constructed by using words from the same category. The dissimilar lists are constructed by using words from different categories. The rationale behind this idea is that similarity between members of the same category is higher than with members of different categories. For instance, “leopard”, “cheetah” and “puma” have in common the features that they are dangerous animals, big wild cats, have a tail, a fur, are carnivores, etc. This way of constructing the material is powerful because each word is used equally often in the similar and dissimilar lists. This implies that all individual items’ linguistic properties, which are known to impact WM performance, are controlled for, such as word length, lexical frequency, or word concreteness (Cowan et al., 1992; Hulme et al., 1991; Walker & Hulme, 1999), among others.

Several lines of evidence have shown that category membership is a valid measure of semantic similarity. First, people generate more shared features for members of the same category than for items from different categories (Binder et al., 2016; Devereux et al., 2014).

Second, functional neuroimaging studies using representational similarity analyses have shown that members of the same semantic category elicit more similar patterns of neural activation than members of different categories (Xu et al., 2018). These observations have been made specifically in core regions of semantic processing, such as the anterior temporal lobe (Lambon-Ralph et al., 2017). A third line of evidence comes from studies on proactive interference (Craik & Birtwistle, 1971; Wickens, 1970). When subjects are tested in a delayed recall paradigm, their memory typically decreases over trials. This proactive interference effect can be released by switching the category of the to-be-remembered items (e.g., from digits to letters), suggesting that the shared features of memoranda from trials N-X interfered with those of trial N. A change of semantic category is known to successfully release proactive interference. If items from the same category weren't similar to each other in memory, release from proactive interference wouldn't be observed upon a change of category. Taken together, these observations imply that members of the same taxonomic category are indeed semantically more similar than members of different taxonomic categories.

Benchmark #1: Semantic Similarity Benefits Item Recall

The main impact of semantic similarity occurs at the item level. As can be seen in **Figure 1**, left panel, item recall increases for similar vs. dissimilar items in immediate serial recall tasks (Goh & Goh, 2006; Guérard & Saint-Aubin, 2012; Kowialiewski, Gorin, et al., 2021; Kowialiewski, Lemaire, & Portrat, 2021; Kowialiewski & Majerus, 2020; Nairne & Kelley, 2004; Neale & Tehan, 2007; Poirier & Saint-Aubin, 1995; Saint-Aubin et al., 2005; Saint-Aubin & Poirier, 1999a; Tse, 2009). This is also observed if semantic similarity is manipulated by using thematic relationships (e.g., "dog", "bones"), or values derived from latent semantic analysis

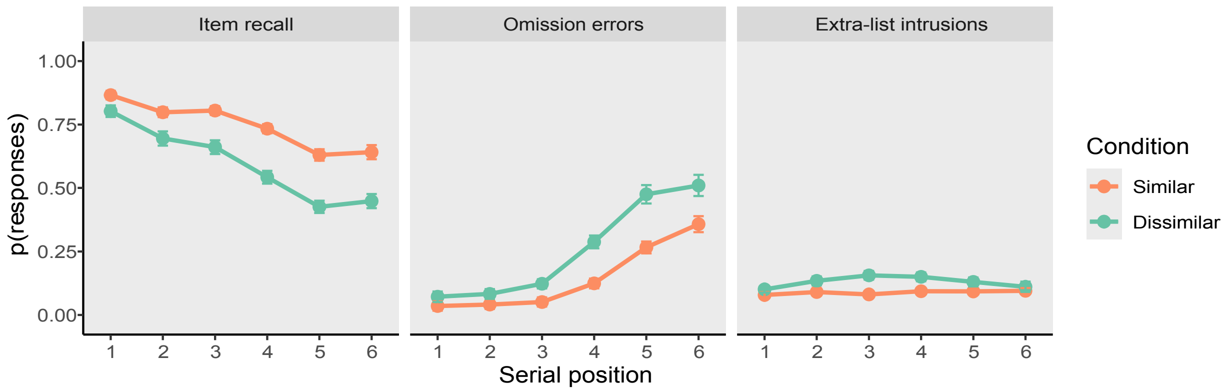
(Landauer & Dumais, 1997), and results are indistinguishable from those observed using taxonomic categories (Rosselet-Jordan et al., 2022; Tse, 2009). The recall advantage for lists composed of semantically similar vs. dissimilar items has been observed in other procedures, such as running span (Kowialiewski & Majerus, 2018), backward recall (Guérard & Saint-Aubin, 2012), Brown-Peterson paradigms (Kowialiewski & Majerus, 2020; Neale & Tehan, 2007), complex span tasks (Rosselet-Jordan et al., 2022), and under concurrent articulatory suppression (Neale & Tehan, 2007; Poirier & Saint-Aubin, 1995; Saint-Aubin et al., 2005; Saint-Aubin & Poirier, 1999a). The beneficial effect of semantic similarity is also observed in children (Monnier & Bonthoux, 2011).

The recall advantage for similar vs. dissimilar lists at the item level can be decomposed into *omission errors* and *extra-list intrusions*. Omission errors refer to participants not recalling any item at all for a given list position.[1] Extra-list intrusions refer to participants recalling an item that was not part of the to-be-remembered list. As can be seen in **Figure 1**, middle panel, semantic similarity mostly impacts performance by reducing the rate of omissions. Although extra-list intrusions are rare, they also slightly decrease for similar vs. dissimilar lists (Poirier & Saint-Aubin, 1995), as can be seen in **Figure 1**, right panel.

Figure 1

The Effect of Semantic Similarity on Item Recall

[1] When not recalling an item at all, participants are usually instructed to say the word “blank”, or to leave an empty answer in case of written recall.



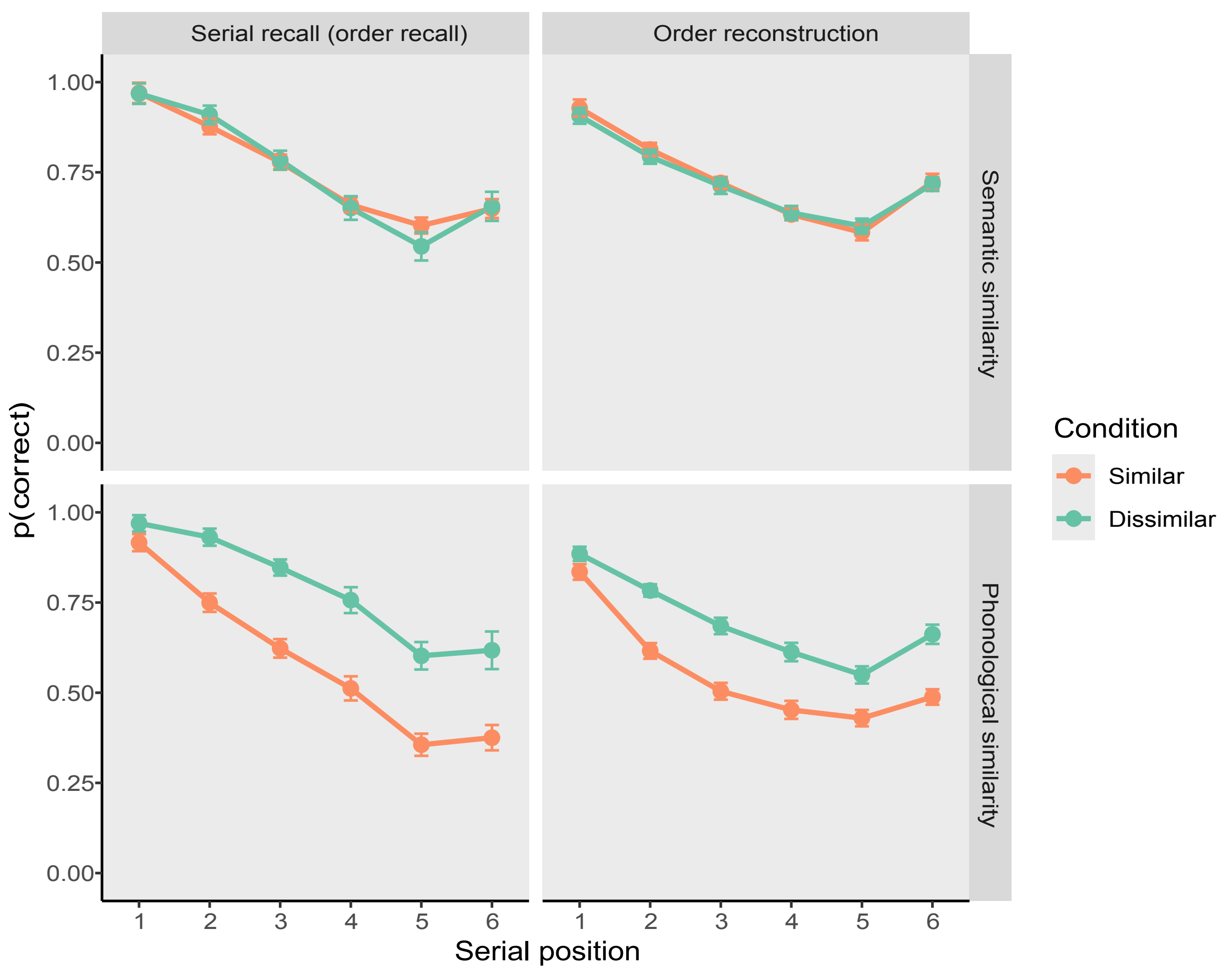
Note. Item recall (left panel) can be decomposed into two broad categories of errors. The failure to recall an item at all is characterized by both omission errors (middle panel) and extra-list intrusions (right panel). Semantic similarity reduces the occurrence of both types of errors. The figure has been adapted from Kowialiewski et al. (2023).

Benchmark #2: Semantic Similarity Does Not Decrease Order Recall

Semantic similarity does not impair order recall, when the scoring procedure used is corrected for the number of items recalled. This null effect is in striking contrast with the phonological similarity effect: Phonological similarity is known to decrease order recall (Baddeley, 1966; Fallon et al., 2005; Gupta et al., 2005; Nimmo & Roodenrys, 2004). Saint-Aubin & Poirier (1999a) were the first authors to consistently show an absence of detrimental effect of semantic similarity on order recall in immediate serial recall. Saint-Aubin and Poirier replicated the null effect on order recall under articulatory suppression with visually presented items. They further replicated it using a small pool of stimuli and an order reconstruction task, thus minimizing the involvement of item recall. Since then, the null effect of semantic similarity on order recall has been replicated several times (Monnier & Bonthoux, 2011; Nairne & Kelley, 2004; Neale & Tehan, 2007; Tehan, 2010), including in recent studies in which this was tested in different manners and using different semantic similarity metrics (Ishiguro & Saito, 2024; Kowialiewski et al., 2023; Neath et al., 2022), with one exception (Guitard et al., 2025) that we will discuss in the General Discussion. The results of our study are displayed in **Figure 2**, upper panel. In our study, we also manipulated phonological similarity for comparison purposes. As shown in **Figure 2**, lower panel, phonological similarity decreased order recall, whereas semantic similarity did not. As we will see, this has important implications regarding how semantic knowledge affects working memory for words.

Figure 2

The Effect of Semantic and Phonological Similarity on Order Recall



Note. Upper panels: Semantic similarity. Lower panels: Phonological similarity, manipulated using rhyming vs. non-rhyming items. Left panels: Order recall, as quantified in an immediate serial recall task. Right panels: Order recall, as quantified in an order reconstruction task. The figure has been adapted from Kowialiewski et al. (2023).

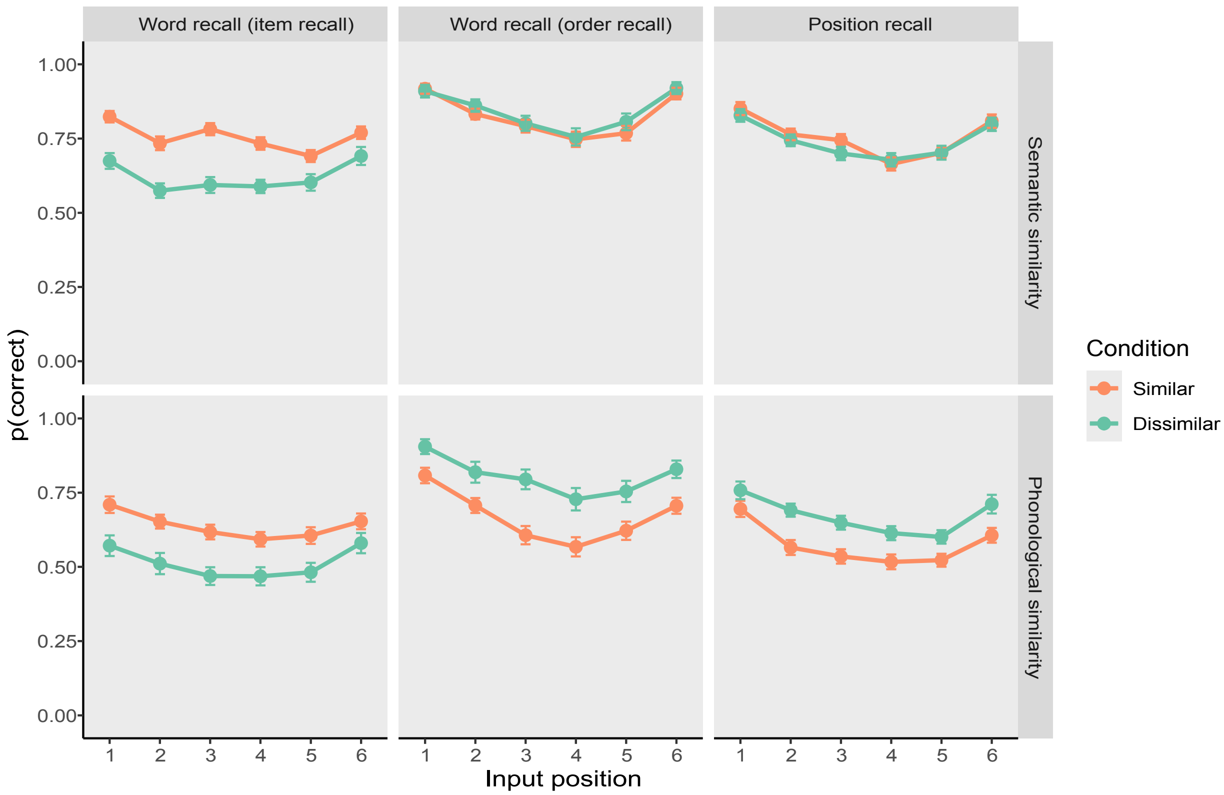
Benchmark #3: Semantically Similar Retrieval Cues do not Lead to Increased Interference

Similarity effects are typically tested in tasks in which the items are the targets of retrieval: The correct item needs to be retrieved for a given list position (see Benchmark #2). A complementary way of testing the effect of inter-item similarity – which has received less attention in the WM literature so far – is to use the items as retrieval cues and ask participants to retrieve their list positions as targets. This reversed direction of retrieval is of interest because it leverages the well-established cue-similarity principle of memory: The more similar two retrieval cues are to each other, the more likely the targets associated to them are confused with each other (Guérard et al., 2009; Mueller & Watkins, 1977; Schneegans & Bays, 2017; Watkins & Watkins, 1976). Hence, to the degree that two lists of words are similar to each other, participants should confuse their list positions when the words are given as cues, and the positions need to be recalled.

Kowialiewski et al. (2023) used this approach to investigate effects of phonological and of semantic similarity. The phonological manipulation involved rhyming vs. non-rhyming lists of items, which is known to have strong and robust detrimental effects on order recall (Roodenrys et al., 2022), and a beneficial effect on item recall (Fallon et al., 2005; Gupta et al., 2005; Lian & Karlsen, 2004; Nimmo & Roodenrys, 2004; Roodenrys et al., 2022). In the study of Kowialiewski et al. (2023), participants studied lists of six words. On half the trials, they were cued with the positions in a random order (e.g., starting with “position 5”, and then “position 2”, and so forth) and were required to retrieve the items associated to them. On the other half of the trials, participants were given the items one by one in a random order, and retrieved the positions associated to them. The results are displayed in **Figure 3**. When the positions were given, and the items were the retrieval targets, semantic (upper panels) and phonological (lower panels) similarity led to increased item recall (left panels), as classically observed. The critical result was to show that only phonological similarity led to increased confusion errors, and this was true for both retrieval directions (middle and right panels). The null effect of semantic similarity in the condition in which the items served as retrieval cues for the positions constitutes a violation of the cue-similarity principle. As the cue-similarity principle has been originally established through variations of the semantic similarity of words in tests of episodic memory (Mueller & Watkins, 1977; Watkins & Watkins, 1976), this cannot mean that the cue-similarity principle does not hold for semantic similarity. Rather, it probably means that word meaning is not represented, or not used, in WM in the same way as in episodic memory.

Figure 3

Semantic and Phonological Similarity Effect on Item Recall, Order Recall, and Memory for Positions



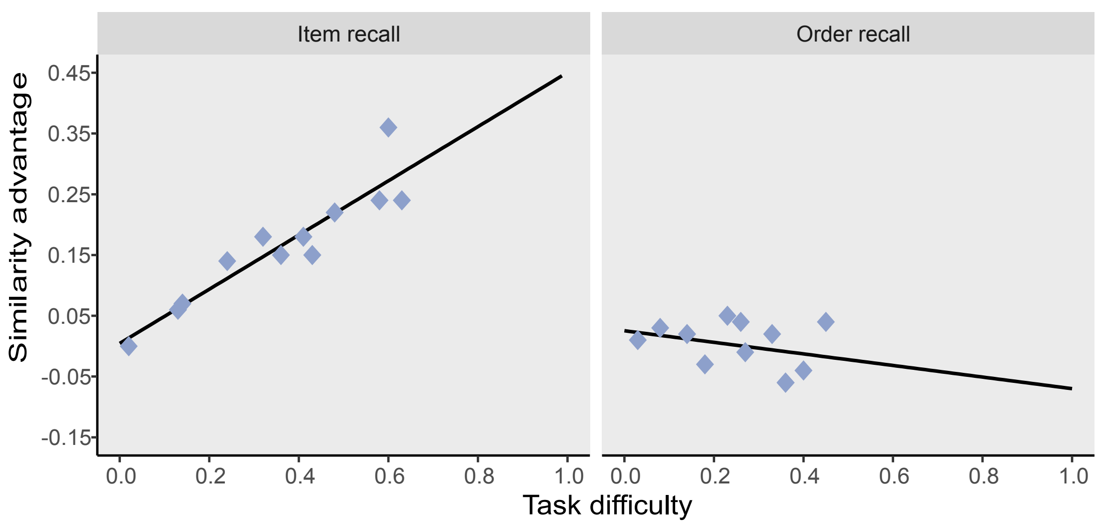
Note. Upper panels: Semantic similarity. Lower panels: Phonological (i.e., rhyming) similarity. Left and middle panels: Participants were cued with a position and were asked to report the item associated to it. Right panels: Participants were cued with a word and were asked to report the position associated to it. The figure has been adapted from Kowialiewski et al. (2023).

Benchmark #4: Semantically Similar Lists are More Resistant to Manipulations of Task Difficulty

When the semantic similarity effect is manipulated under conditions in which WM maintenance gets harder, the difference between similar and dissimilar lists gets larger. This interaction was first observed by Poirier & Saint-Aubin (1995), showing a stronger semantic similarity effect with than without articulatory suppression, which has been subsequently replicated (Saint-Aubin et al., 2005; Saint-Aubin & Poirier, 1999a). Stronger semantic similarity effects have also been observed in a Brown-Peterson than in an immediate-recall paradigm. In the Brown-Peterson task, participants performed a backward-counting task between encoding and recall, which impairs memory (Kowialiewski & Majerus, 2020). Neale & Tehan (2007) offer a comprehensive demonstration of this phenomenon. They parametrically modulated the degradation of WM representations using different interfering tasks and list lengths. **Figure 4** illustrates the magnitude of the semantic similarity effect as a function of task difficulty (measured by the proportion of errors in the dissimilar condition) as observed in Neale and Tehan (2007). The magnitude of the semantic similarity effect linearly increased with task difficulty (**Figure 4**, left panel). The null effect on order recall, in contrast, remained unchanged (**Figure 4**, right panel).

Figure 4

Semantic Similarity as a Function of Task Difficulty



Note. In this study, task difficulty was defined as 1.0 - p(correct) in the dissimilar condition. For instance, an item score of 0.8 was defined as less difficult (i.e., task difficulty equal to 0.2) than a condition leading to an item score of 0.6 (i.e., task difficulty equal to 0.4). The figure has been reproduced using the values reported in Neale and Tehan (2007), Table A1.

Benchmark #5: Semantic Similarity Modulates the Type of Intrusion Errors

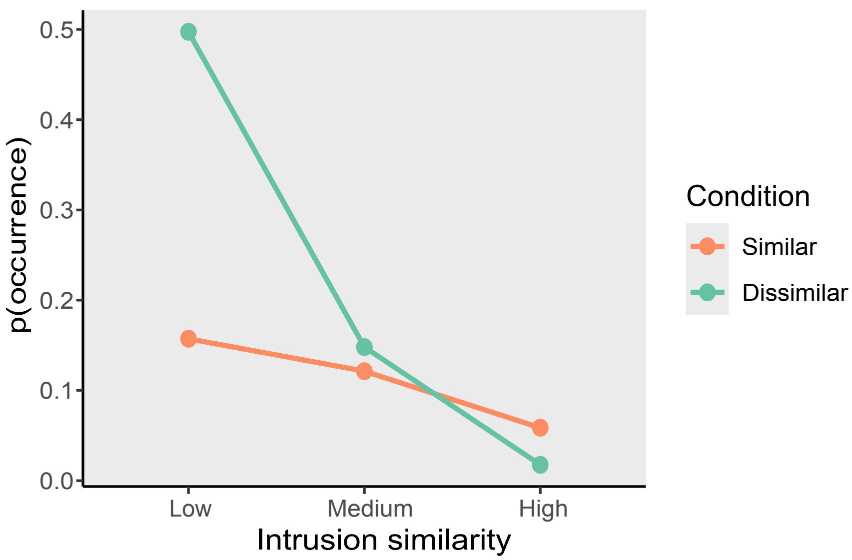
Tehan (2010) compared semantically dissimilar lists to lists of similar items constructed from Deese-Roediger-McDermott lists (McEvoy et al., 1999), in which all words have strong semantic associations to a critical lure that is itself not included. In the long-term memory literature, this manipulation typically induces so-called "false memories" (Deese, 1959; Roediger & McDermott, 1995), in which the critical lure is recalled more often in the similar than the dissimilar list. The same phenomenon was observed in Tehan (2010): The critical lures were recalled more often in semantically similar (6%) than in dissimilar (0%) lists. This effect shows that the composition of a list affects the type of intrusion errors occurring in immediate serial recall.

Another way to look at this phenomenon is via a detailed analysis of extra-list intrusions. We reanalyzed data from six experiments (Kowialiewski et al., 2023; Neath et al., 2022) involving 330 participants who recalled lists of semantically similar and dissimilar items across various experimental conditions (i.e., serial and cued recall). For each trial, we calculated the similarity between each extra-list intrusion and the target item (i.e., the item they were substituted for) using Google news word2vec semantic vectors (https://github.com/mmihaltz/word2vec-GoogleNews-vectors). The extra-list intrusions were then categorized into three bins based on their similarity with their target items: low, medium, and high. The bins were defined by dividing the interval between the minimum and maximum similarity value[2] into three equal ranges. **Figure 5** shows the proportion of intrusion errors for each semantic similarity condition and bin, normalized by the total number of intrusion errors across all conditions. The data reveal that low-similarity intrusions are by far the most common type of error, accounting for approximately 70% of the sample. However, the distribution of intrusion errors varies depending on list composition. In semantically dissimilar lists, low-similarity intrusions dominate, with very few high-similarity intrusions. In contrast, in semantically similar lists, the proportion of low-similarity intrusions decreases substantially, and high-similarity intrusions occur more frequently than in semantically dissimilar lists.

Figure 5

Distribution of extra-list intrusions

[2] The minimum and maximum similarity values were taken from the all the similarity values between every word2vec vector and every target that has been replaced by an intrusion.



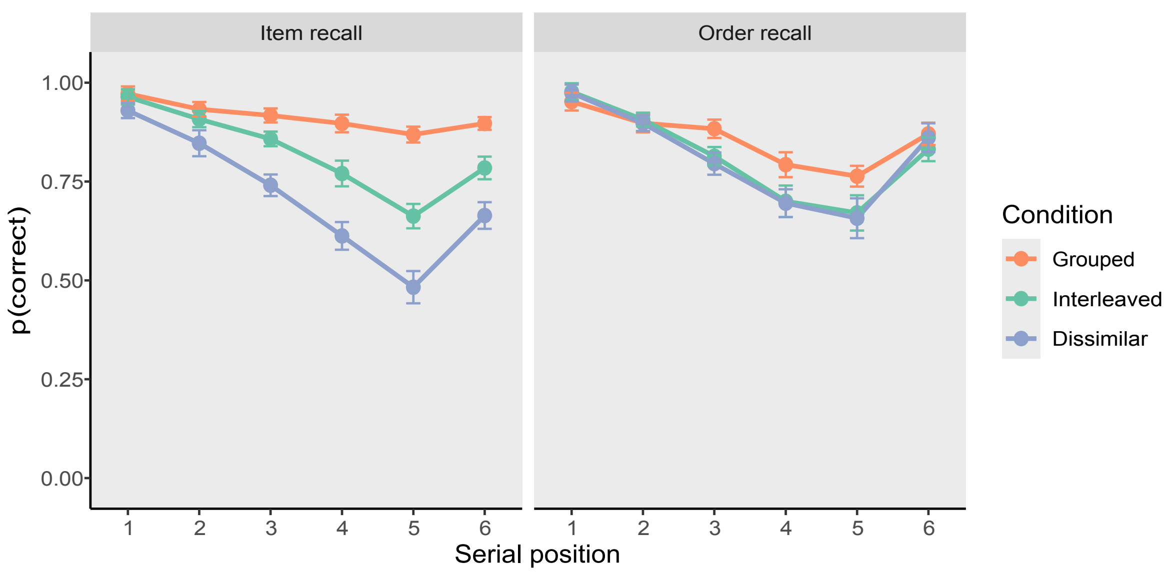
Note. Proportion of intrusion errors as a function of semantic similarity (similar vs. dissimilar). Each extra-list intrusion was categorized into one of three bins based on its similarity to the target item (i.e., the item it replaced): low, medium, and high.

Benchmark #6: The Separation Effect

Saint-Aubin et al. (2014) explored how the positional distance separating two similar items affects serial recall. Their results showed weaker semantic similarity effects as the positional distance between similar items increased. This *separation effect* has recently been replicated by Kowialiewski, Gorin, et al. (2021) using a procedure in which participants were presented with lists composed of three items from each of two categories. In one condition, the similar items were presented at adjacent serial positions, in a grouped manner (i.e., AAABBB, where each letter refers to a semantic category). In another condition, the items were interleaved (i.e., ABABAB). Compared to a dissimilar condition, the grouped condition led to the largest benefit for item recall (i.e., strongest semantic similarity effect). The effect was only half as large in the interleaved condition. Results from a recent replication (Kowialiewski, Majerus, et al., 2023, Experiment 1) are illustrated in **Figure 6**. As can be seen, the separation effect has a beneficial effect on item recall (left panel). There is also a small beneficial effect on order recall (right panel), which is only observed when items are grouped. We recently showed that the separation effect is specifically due to the separation between the similar items at presentation and is not modulated by the order in which they are recalled (Kowialiewski, Krasnoff, et al., 2022), suggesting that the effect originates at encoding. In the same study, we showed that when more than one item separates every two similar items, the semantic similarity effect almost disappears.

Figure 6

The Separation Effect



Note. In the grouped condition, two categories composed each of three similar words were presented in sub-groups at encoding (i.e., AAABBB). In the interleaved condition, the similar words were presented in an interleaved fashion (i.e., ABABAB). In the dissimilar condition, all words were drawn from distinct semantic categories (i.e., ABCDEF).

Benchmark #7: Semantic Similarity Constrains Order Errors

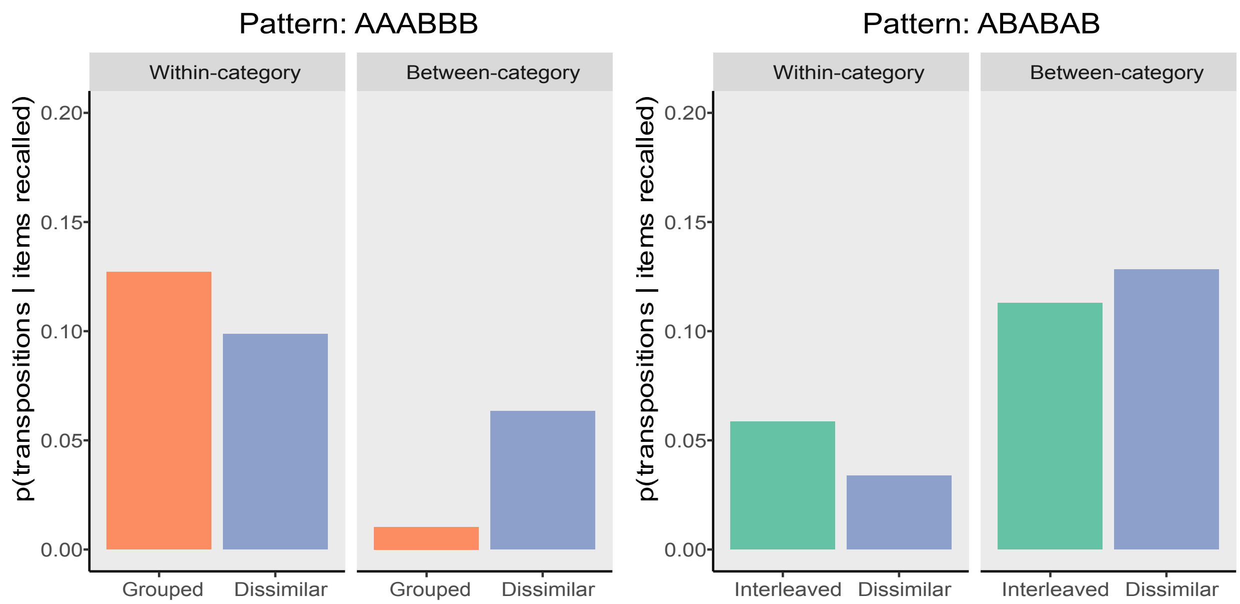
Although semantic similarity does not impair order recall, it can constrain the *pattern* of order errors. Poirier et al. (2015) presented participants with lists in which the three first items were semantically similar. In a control condition, the three last items remained dissimilar (e.g., "officer, badge, siren, music, tourist, yellow"). In an experimental condition, the fifth item was semantically similar to the triplet in the first half of the list (e.g., "officer, badge, siren, fence, police, tractor"). They observed an increase of order errors of the fifth item in the experimental conditions, compared to the fifth item in the control condition. Specifically, participants more often recalled "police" erroneously in the list positions of "officer", "badge" or "siren". For simplicity, we will refer to this phenomenon as an increase of *within-category transpositions*.

Similar results have been found in Kowialiewski, Gorin, et al. (2021), who presented two categories of semantically similar items in a grouped (AAABBB) or interleaved (ABABAB) manner (i.e., see also previous section on the separation effect). Dissimilar lists (ABCDEF) served as a control condition. Semantic similarity changed the pattern of transposition errors: In semantically similar lists, there were more within-category transpositions compared to matched positions in dissimilar lists, and fewer between-category transpositions (i.e., a transposition involving the migration of an item toward the position of a semantically dissimilar item) compared to dissimilar lists. The full pattern of transposition errors is illustrated in **Figure 7**. The figure plots the number of transposition errors out of the total number of items recalled as a function of semantic condition (left panel: grouped vs. dissimilar, right panel: interleaved vs. dissimilar) and transposition type (i.e., within-category, between-category). What this graph shows is that, in a list such as "**leopard** – **puma** – **tiger** – Denmark – Belgium – Switzerland", people rarely produce confusion errors such as "**leopard** – **puma** – Denmark – **tiger** – Belgium – Switzerland". Instead, when any confusion error occurred, it involved mostly items from the same semantic category. In contrast, items in the dissimilar condition moved throughout the list more freely.

Because the effect was larger in the study of Kowialiewski and colleagues (Cohen's d = 1.244) than the one of Poirier et al. (2015), we decided to focus on results from the two-category design of Kowialiewski et al. as benchmark. A recent work of our own (Kowialiewski, Majerus, et al., 2024) showed that the effect replicates regardless of presentation modality (i.e., auditory, written), and test method (i.e., serial recall, order reconstruction). Furthermore, the increase of within-category transpositions is not due to participants developing long-term memory knowledge or expectations about the semantic list structure during the experimental setup, as the effect persists when participants cannot predict the lists' semantic structures, suggesting a non-strategic origin.

Figure 7

Transpositions as a Function of Semantic Similarity Structure and Transposition Type



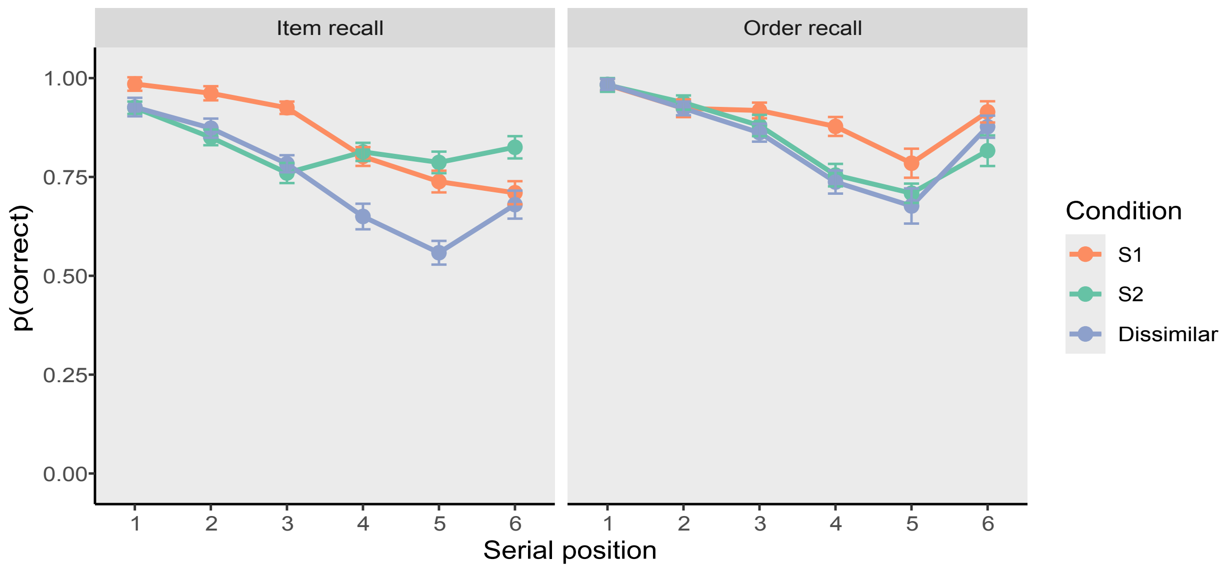
Note. In the study reported by Kowialiewski et al. (2023), participants were presented with lists composed of two semantic categories represented in groups (i.e., pattern AAABBB) or interleaved (i.e., ABABAB). As compared to items presented at identical positions in a dissimilar condition, items in the semantically similar lists, when migrating, tended to be transposed more often toward semantically similar, and less toward other dissimilar items of the list. The y-axis represents the number of transposition errors of a certain type (i.e., within-category or between-category) out of the total number of items recalled in a particular condition. The dissimilar condition represents a control to see what would normally happen in conditions where no semantic structure was given.

Benchmark #8: Semantic Similarity Proactively Impacts Working Memory Performance

The last benchmark focuses on a phenomenon first reported by Brooks and Watkins (1990) and recently replicated by Kowialiewski, Lemaire, et al. (2021). Kowialiewski and colleagues tested serial recall of six-word lists in which three successive words were semantically similar. In one condition (S1), the similar items were presented at the beginning of the list (e.g., flute, guitar, piano, wall, sky, tomato). In another condition (S2), the similar items were presented at the end of the list (e.g., leopard, bike, table, Mars, Jupiter, Venus). As compared to a dissimilar condition, the similar items themselves were better recalled at the item level, thus replicating the general beneficial effect of semantic similarity. The novel finding is that when the similar items were presented at the beginning of the list, the subsequent items were better recalled compared to the same items in the dissimilar condition. Hence, semantic similarity among the early list items had a beneficial *proactive effect* on subsequent, dissimilar items. However, when the similar items were presented at the end of the list, recall performance for the preceding items was not affected. There was therefore no *retroactive effect*. An analogous phenomenon has been reported by Miller & Roodenrys (2012) using high and low frequency items. The results are illustrated in **Figure 8**, showing a proactive benefit for both item and order recall.

Figure 8

Proactive Beneficial Effect of Semantic Similarity



Note. In the S1 condition, items were semantically similar only in the first half of the list (i.e., positions 1, 2 and 3). In the S2 condition, items were semantically similar in the second half of the list (i.e., positions 4, 5 and 6). The remaining items were all semantically dissimilar. As can be seen, the presence of similar items in the S1 condition proactive enhanced memory performance for the subsequent dissimilar items, despite these dissimilar items having the same linguistic status as the items in the dissimilar condition. In contrast, no retroactive impact occurred.

Empirical Section: Summary

To sum up, semantic similarity is characterized by a rich pattern of effects. It mostly impacts WM performance by increasing item recall, with semantically similar items leading to reduced omission errors and extra-list intrusions. Current evidence indicates a lack of semantic similarity effect on order recall when manipulated with pure lists. The absence of a semantic similarity effect on confusion errors occurs regardless of the retrieval direction tested and stands in contrast to the robust effect of phonological similarity on confusion errors. The semantic similarity effect is stronger under difficult maintenance conditions. It is sensitive to the positional distance that separates similar items at encoding. Semantic similarity can constrain serial order errors in mixed lists, and it proactively improves WM performance for dissimilar items. In the next section, we describe a connectionist architecture modeling the interactions occurring between WM and the semantic long-term memory system, which explains most of these phenomena.

Model Description: General Principles

In this section, we first describe the general cognitive principles of the architecture we used to simulate semantic similarity effects. We start by describing the WM architecture itself, followed by the principles specific to the representation of semantic information in WM. We will test the ability of this mechanism to qualitatively capture the benchmarks presented above.

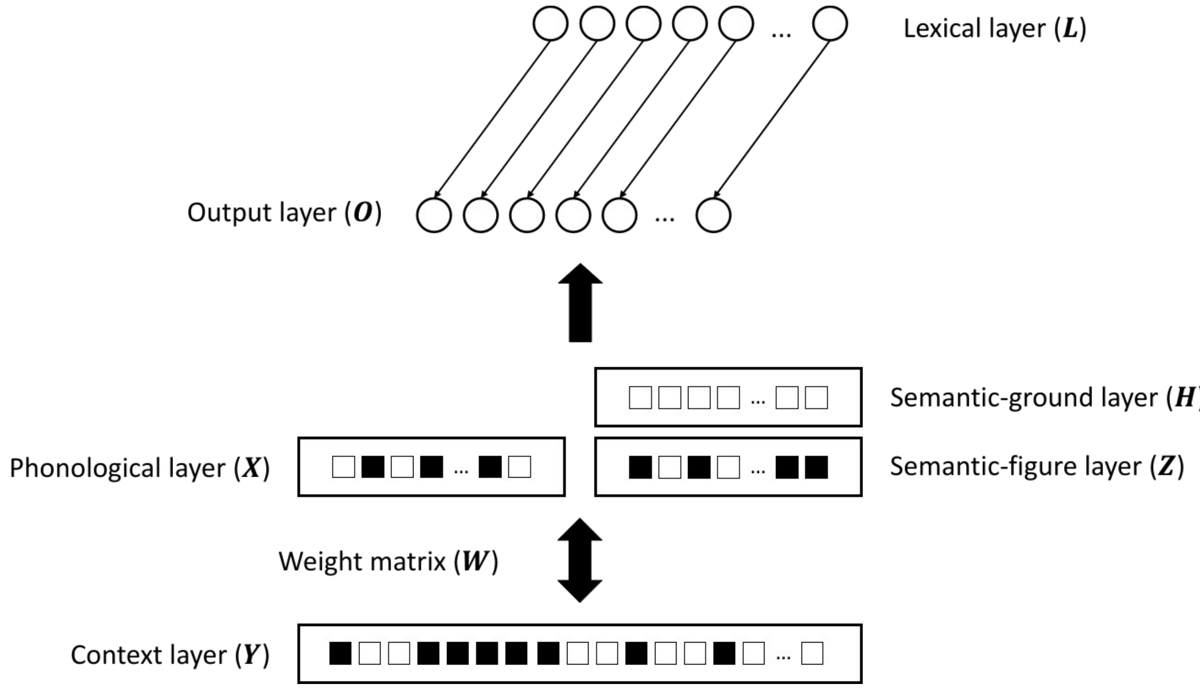
The Working Memory Architecture

In this section, we present a verbal description of the model we used to integrate meaning. The various approaches we will introduce to represent meaning in WM exhibit consistent behavior regardless of the specificity of the architecture. Nonetheless, choosing an architecture involves committing to assumptions. We chose to use a WM architecture using general principles shared by various models (Burgess & Hitch, 1999; Henson, 1998; Lewandowsky & Farrell, 2008; Oberauer et al., 2012; Oberauer & Lewandowsky, 2011).

Encoding into WM is done by associating items to contexts. For instance, when presented with the sequence “flower – pancake – leopard”, the model associates “flower” to position 1, “pancake” to position 2, and so on. The core WM representation is stored in these associations between items (i.e., the words) and contexts (i.e., the positions). In addition to item-context bindings, encoding an item into WM automatically activates its representation in long-term memory, and this activation persists for some time. This is a general idea taken from embedded processes models of WM (Cowan, 1999; Nee & Jonides, 2008; Oberauer, 2002, 2009). Basically, each word has a pre-existing lexical representation (i.e., equivalent to vocabulary) in long-term memory, coded by a localist unit. Every time a word is recognized as such, its lexical unit gets activated. This activation lies outside of the core item-context binding system. The architecture also includes an output layer, which contains a localist unit for each item in the lexicon. Features in the item layer are connected to each localist unit in the output layer through connection weights that represent the person's long-term learning about words. Specifically, the vector of connection weights between the item layer and the localist unit of a word in the output layer is identical to the vector of features by which that word is represented in the item layer. Thus, the more the pattern of activation in the item layer approximates the original feature vector of a word, the more the localist unit representing that word in the output layer is activated. In addition, each node in the output layer has one connection with its corresponding lexical representation. The structure of the architecture is reported in **Figure 9**.

Figure 9

Illustration of the Working Memory Model



Note. The architecture consists of a context layer ($\boldsymbol{Y}$), an item layer, split into phonological and semantic parts ($\boldsymbol{X}$, $\boldsymbol{Z}$, and $\boldsymbol{H}$), an output layer ($\boldsymbol{O}$), and a lexical layer ($\boldsymbol{L}$). Both the items and the contexts are represented in a distributed fashion. Both layers are fully inter-connected through the weight matrix $\boldsymbol{W}$. In addition to the core temporary representations stored in $\boldsymbol{W}$, items have an activation value using localist representations stored in the lexical layer $\boldsymbol{L}$ in long-term memory. The output layer $\boldsymbol{O}$ is used to select items based on their activation levels, and contains items' localist representations. The way the model retrieves items is illustrated in **Figure 10**.

Following Oberauer and Lin (2024), the strength of encoding through item-context binding is modulated in two ways. First, encoding into WM follows a primacy gradient of activation (Page & Norris, 1998), such that encoding strength is maximal for the first presented item, and progressively decreases for each newly encoded item. Second, each newly encoded item decreases the strength of previously encoded representations, a mechanism called *automatic updating*. The rationale behind this idea is that each new event encoded into WM is prioritized, which has the consequence of de-prioritizing previous representations through weakening their item-context associations by a constant proportion (Oberauer & Lin, 2024). This mechanism produces a recency gradient. Jointly, the primacy gradient and automatic updating generate the U-shaped serial position curve that is typically observed in immediate memory for lists (Oberauer et al., 2018).

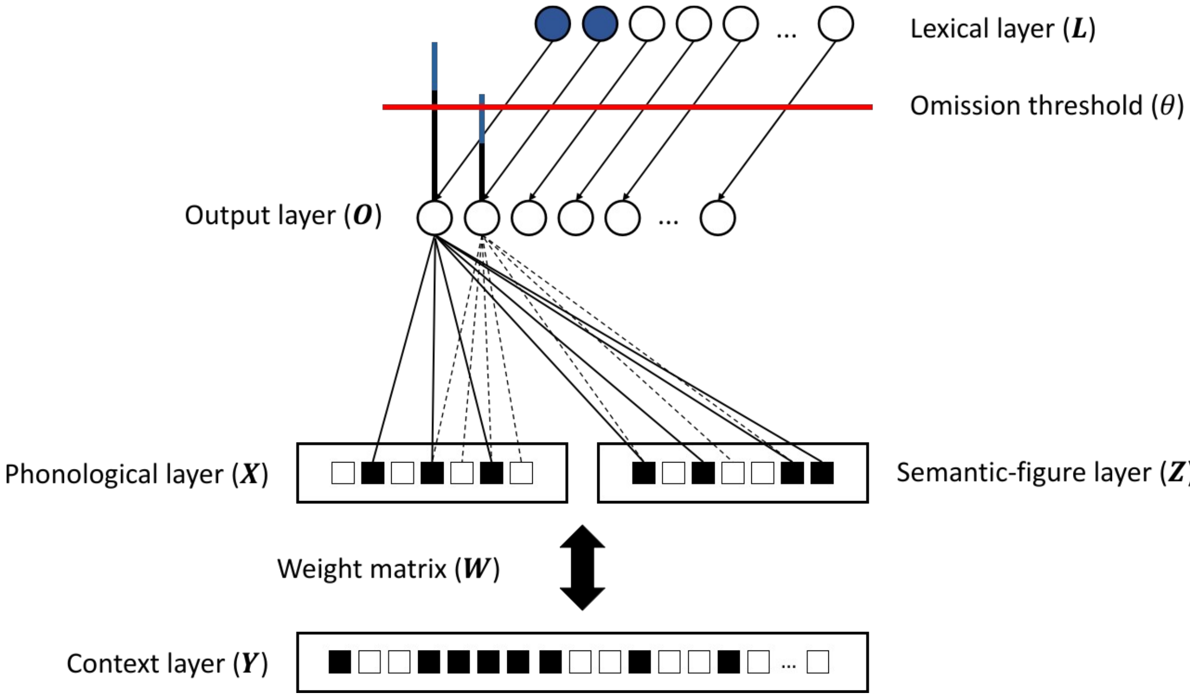
In most WM tests, participants recall each item in a given list position. **Figure 10** illustrates this retrieval process in the model. Recall starts by cueing the to-be-remembered item with its context (e.g., cueing the first item with “position 1”), which leads to the reproduction of a distorted version of the original item in the item layer. This distortion occurs because the context Y_i serving as a cue shares a proportion of features with other contexts. The items associated to those other contexts are therefore also partially retrieved. As a consequence, it is not the original item that is retrieved, but a blend of all items bound to any context Y_k through the weight matrix $\boldsymbol{W}$, weighted by how much context Y_k overlaps with Y_i.

The distorted representation of the to-be-recalled item cannot be produced as response directly. Instead, an item must be produced by selecting among a set of retrieval candidates (Schweickert, 1993). In immediate serial recall of words, the retrieval candidates are the words stored in long-term memory, that is, people’s vocabulary. In order reconstruction tasks where the list items are provided at retrieval, the retrieval candidates are just the given list items. Selection of a candidate in the model is based on the activation of items’ localist units in the output layer. Basically, when a distorted vector is retrieved following the cueing process, its activation is forwarded from the item layer to the localist representations of retrieval candidates in the output layer. In the output layer, the candidates matching the retrieved representation more strongly have a higher activation level, and therefore a higher probability of being selected. For instance, given the retrieved representation “C_T” and the recall candidates “CAT, COT, GEAR, MINE”, it is much more likely to select “CAT” or “COT” as the response than the other items, because they receive stronger activation from the retrieved features corresponding to “C_T”.

At this point, the persistent lexical activation plays a role: The candidate’s persistent activation in the lexical units is added to the degree of match of each candidate to the retrieved representation in the output layer through their one-to-one connections, resulting in the candidate’s activation level with which it enters the competition for being selected as the response. In this way, items with higher persistent activation are more likely to be recalled than weakly activated items. If “CAT” was part of the current memory list in the example above whereas “COT” was not, then “CAT” would be the most activated lexical unit because it combines activation from the comparison to the retrieved representation “C_T” with persistent activation in the lexical layer. Omissions in the model are implemented by a threshold on the activation level of candidates in the output layer. Modeled this way, items receiving stronger persistent activation in the lexical units – as well as items that are bound more strongly to their contexts – are less likely to be omitted, because they will have a higher chance of surpassing the omission threshold.

Figure 10

Working Memory Architecture – Retrieval Phase



Note. Retrieval is performed by cueing the item with its context through the weight matrix $\boldsymbol{W}$, which leads to the retrieval of a distorted version of the original item. This representation is then compared to all *N* items stored in long-term memory, leading to a degree of activation of each item in the output layer. The persistent activation in the lexical layer is added on top of the activation in the output layer.

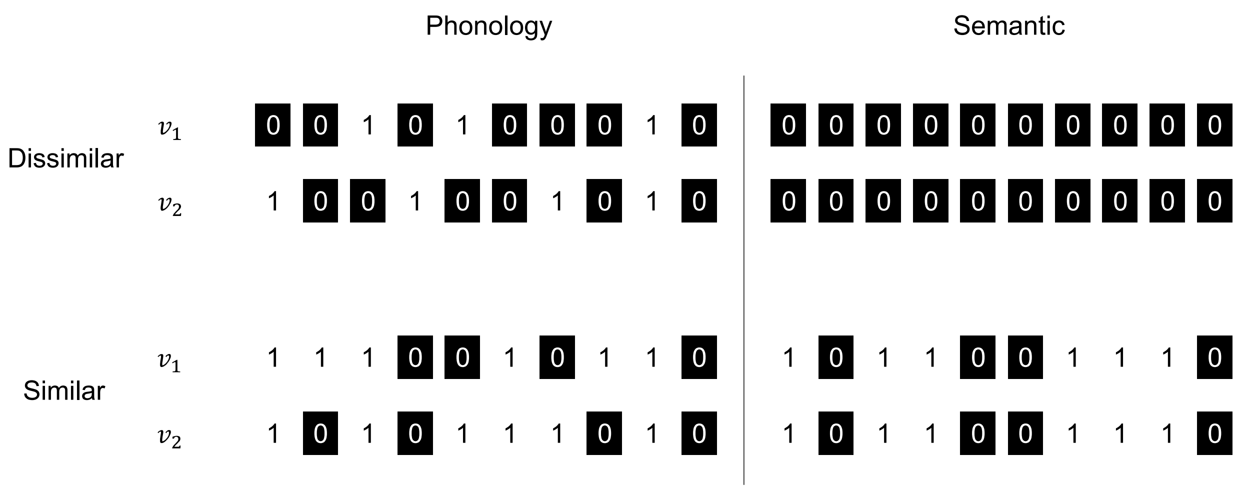
After recalling an item, two processes occur simultaneously. First, automatic updating reduces the strength of the whole WM representation stored in the weight matrix by a constant proportion. Supporting this assumption, studies have found clear evidence that retrieving and/or recalling an item leads to forgetting (Cowan et al., 2002; Oberauer, 2003), a phenomenon called *output interference*. Second, the activation of the just-encoded item is downgraded in the lexical layer, which prevents the model from repeating items it already recalled. Note that if no item has been recalled (i.e., omission error), the items' activation is not downgraded. After each recall attempt, activation in the output layer is reset to zero and a new retrieval attempt occurs. The recall phase finishes after the last recall attempt.

Semantic Encoding Mechanism

Each time an item is encoded, it triggers the activation of the items' associated semantic features in different modality-specific regions across the neo-cortex (Binder et al., 2009; Binder & Desai, 2011; Lambon-Ralph et al., 2017). One important aspect of our implementation is the way semantic features are encoded (i.e., bound to their positional context). We assume that only semantic features shared among several items are encoded in this way. In a list in which items are drawn from the same taxonomic category, only the items' features characteristic for that category will be encoded, which boils down to encoding the category itself. For instance, when encountering a list such as "knife – sword – dagger", people maintain features common to all items, such as "hurts people", "is dangerous", "is a weapon", etc. This mechanism implies that in a list containing semantically dissimilar items, only a negligible number of shared semantic features are retained due to the limited overlap between these items.

Figure 11

Difference Between Phonological and Semantic Representations



Note. This figure illustrates the way phonological and semantic information are represented in the model. White (1) and black (0) values represent active and inactive features, respectively. Left panel: In our model, we assume that all features of a phonological representation are always encoded. When that is the case, it is more difficult to discriminate between two vectors sharing a large proportion of features (lower part) than two vectors sharing little of their features (upper part). Right panel: Different from phonological representations, we assume that the semantic features of an item are encoded only when shared with other memoranda. When that is the case, two vectors will be impossible to discriminate based on their semantic components, regardless whether they are similar (lower part) or dissimilar (upper part).

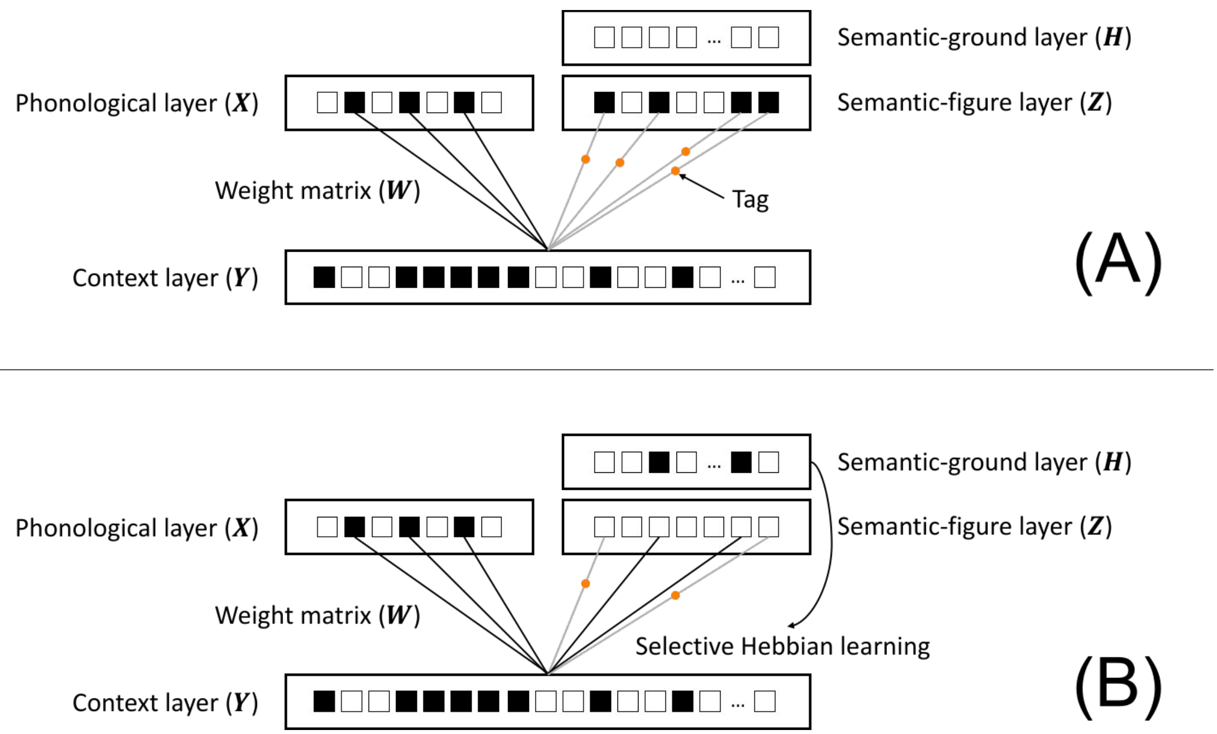
This category-encoding assumption gives unique characteristics to the model's behavior. Suppose two vectors, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, representing the phonological and the semantic features of two items that are bound to their respective positional contexts, as illustrated in **Figure 11**. In a condition in which these two vectors are semantically dissimilar (upper panel), all semantic features of the two vectors will have zero values, because none of their features are shared. In this condition, it is impossible to discriminate the two vectors solely based on their semantics.

Only the phonological part of the vector is relevant to discriminate the items from one another. Now let's consider two semantically similar items (lower panel). Both items are represented by their shared features. In this situation, the semantic part of the vectors is also useless to discriminate between the two items. Hence, as in the dissimilar condition, both items can be discriminated only via the phonological part of the vector. Thanks to this property, items in semantically similar lists will not be confused more often with each other than in lists of dissimilar items, although semantic information – including information about the respective positions of semantic categories within the list – is maintained in working memory. Although the supplementary semantic features encoded with semantically similar items have no effect on order recall, they do improve item recall, because they provide an additional source of activation that helps retrieved representations surpass the omission threshold.

There is one major problem when considering a model which encodes only items' shared features. When first encountering the item "knife", it cannot be known in advance which, if any, semantic features it will share with subsequently presented items. It is only when encountering the words "spoon" and "fork" that it is possible to identify the semantic features they share. At that point in time, these features should be encoded into WM by binding them to the positions in which the three items have been presented. By contrast, when people encode "knife", "squirrel", and "wall", there are few, if any, semantic features shared among them, and hence, the model should not encode any semantic features in WM. We made this possible via the combination of two processes.

Figure 12

Illustration of the Tagging Mechanism



Note. (A) At encoding, new associations are formed between phonological and contextual features. Associations between semantic features and contextual features are not yet formed, but tagged. (B) Based on the tagged associations, selective Hebbian learning may occur later, depending on the relevance of a semantic features in the current trial, based on the features' activation. Note that to keep the figure readable, connections from the item layer are shown only for one unit in the context layer. In the model, item and context layers are fully interconnected such that each unit in the item layer has a connection to each unit in the contextual layer.

First, we implement a learning mechanism based on the creation of synaptic tags (Rombouts et al., 2015). In this mechanism, illustrated in **Figure 12**, associations between semantic features and positional contexts are not immediately encoded, but tagged. These tags are formed through Hebbian learning: Connections between currently active units in the context layer and currently active semantic features in the item layer are tagged. This tagging process works with all semantic features of a word. These features' activation is short-lived; it lasts only for as long as an item remains in the focus of attention.[3] A tag marks a connection between item and context units as one that can be strengthened at a later point in time. A tagged connection is strengthened when the semantic feature that it connects to some context unit is activated in a more persistent manner than the fleeting activation that forms the tags. For that reason, the model represents each semantic feature in the item layer twice; once for the short-lived activation of these features through encoding, which drives tagging, and once for the longer-lived activation of these same features, which represents their longer-lasting relevance in the context of the current processing episode. A processing episode could involve reading a sentence, or encoding a list of items for immediate recall. This distinction between transient and long-lasting activation relates to the figure-ground distinction found in the text comprehension literature (Smith, 2012). The figure is what captures the immediate focus of attention and is directly available to one's mind. The ground is the broader context which supports and enriches the current representation (i.e., the figure) that people process. Hence, we distinguish between a *semantic-figure layer* and a *semantic-ground layer*.

Depending on the relevance of a semantic feature over the course of the trial, a new association may be formed using the tags. This is where the second mechanism comes into play. Shared semantic information is kept active thanks to a threshold mechanism which controls the persistent activation of items' semantic features in the semantic-ground layer. Longer-lasting activation of a semantic feature is initiated only if that feature is encoded multiple times in close

---

[3]This short-lived semantic activation must not be confused with decay in WM: We assume that rapid decay occurs in the linguistic system, which is suggested by studies showing that semantic priming effects are short-lived (McNamara, 1992). This short-lived activation is unrelated to how WM representations are maintained.

succession, as when several words from the same category are encoded closely together in a list. Once a semantic feature achieves longer-lasting activation, all tagged connections of that feature with context units are strengthened. This way of encoding semantics allows the model to selectively bind only those semantic features to their positional contexts that are shared among several items, without knowing ahead of time which features an item will share with subsequently presented items. Details of these mechanisms are presented in the computational implementation section.

Computational Implementation

In this section, we describe the mathematical details of the principles explained above.

The Working Memory Model

Encoding

When a list item *i* is presented, its content is activated as a distributed representation of features in the item layer. This includes phonological features as well as semantic features in the semantic-figure layer and the semantic-ground layer. At the same time, the item's serial position in the list is activated in the context layer as a distributed representation. Mathematically, these distributed representations are vectors of activation values.

The phonological part of the item feature vector is immediately bound to the context vector through rapid Hebbian learning:

$$\Delta W_{xy} = \eta_k X_x Y_y \qquad \textit{Eq. 1}$$

In Eq. 1, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the activation vectors of the current item (i.e., its phonological part) and the current context, respectively; ΔW_{xy} represents the change to the connection weight between unit x in the item layer and unit y in the context layer. The η_k term is the binding strength for the binding of an item to its context k, which depends on a primacy gradient of binding strength, and the automatic updating process:

$$\eta_k = \left(1.0 + \beta \pi^{k-1}\right) \delta^{n-k} \qquad \textit{Eq. 2}$$

Where β and π are free parameters controlling the initial value of the binding strength of the first item, and the progressive decrease in binding strength across serial position k, respectively. The n term corresponds to the number of memoranda. Hence, we assume WM representations as being encoded across serial positions with decreasing strength, generating a primacy gradient (Page & Norris, 1998). This primacy gradient does not directly scale the binding strength but is added as a boost on top of a constant baseline strength (i.e., 1.0). Next, the δ^{n-k} term refers to an automatic WM updating process which weakens already encoded items after a new item enters WM. Every item has its binding strength proportionally reduced $n-k$ times during encoding of the subsequently presented items. By this mechanism, the binding strengths of earlier encoded items progressively decrease as more items are encoded, thereby producing a recency gradient. This recency gradient can help to explain the recency effect observed in many WM paradigms, such as running span procedures involving rapid presentation of long lists of items (Bunting et al., 2006; Hockey, 1973; Hockey & Hamilton, 1977; Pollack et al., 1959).

In addition to binding an item with its context, the presentation of an item automatically activates that item's lexical localist unit, and that activation is sustained over time (Cowan, 1999; Oberauer, 2009):

$$L_i = a \qquad \textit{Eq. 3}$$

where $\boldsymbol{L}$ is a vector of activation values of localist representations of all items, and the subscript *i* indexes the currently encoded item. The parameter a controls the strength of the persistent activation, and is a free parameter.

A core property of our model is the representation of item similarity. The basic WM model represents phonological similarity. The representation of semantic similarity will be explained in a detailed manner in the next section. Phonological similarity is defined in a similarity matrix $\boldsymbol{M}_{phon}$ which stores the similarities between all pairs of items. The similarity value between two items can take any value between 0.0 and 1.0. which encompass the whole lexicon, which for computational reasons is limited to $N=64$ items. Extending item similarity beyond the memoranda allows modeling of extra-list intrusions, because non-list items sharing a proportion of phonological or semantic features with the list items will be partially cued during the retrieval process described below. Phonological similarity between memoranda is determined by the parameter S_1. In simulations in which phonological similarity is not of main concern, the value of this parameter was fixed to 0.1. The phonological similarity between memoranda and all other items is controlled by a free parameter, S_2. As items are maximally similar to themselves, they have a value of 1.0 in the similarity matrix.

Likewise, contexts also have similarity values, stored in the similarity matrix $\boldsymbol{C}$. Similarity values for contexts are determined by a free parameter P. The similarity between any two contexts *j* and *k* decreases exponentially with their positional distance. Thus:

$$C_{jk} = P^{|j-k|} \qquad \textit{Eq. 4}$$

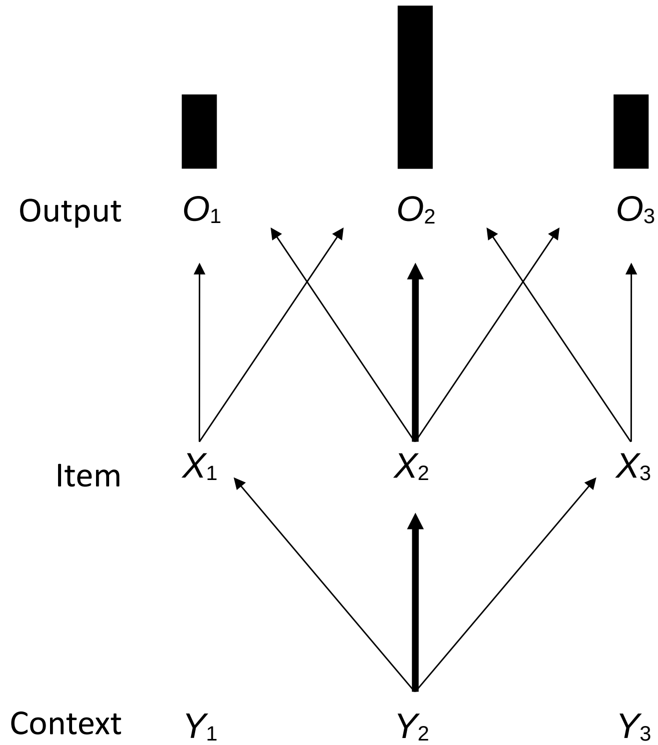
Retrieval

Figure 13 illustrates the flow of activation from one layer to the next in the network during retrieval. At retrieval, the serial position *k* of the to-be-retrieved target item is activated in the context layer. This activation is fed through the connection weight matrix $\boldsymbol{W}$ to the item layer. This generates a pattern of activation in the item layer that is the weighted sum of the activation patterns of all items previously bound to any context *j*, each weighted by the similarity of their positional context to the currently activated context, C_{jk}, and the binding strength of the item to its position, η_j. The distributed item representations are added together in the item layer, but in **Figure 13** we keep them separate for clarity.

The item layer is linked to the output layer through a matrix of fixed connection weights, which associates each distributed representation of an item in the item layer to a corresponding localist representation in the output layer. Hence, each localist unit O_i in the output layer receives activation proportional to the weight with which the feature vector of item *i* is reactivated in the item layer, which is $\eta_j C_{jk}$ for all items bound to any context *j*, and 0 for all others. In addition, the feature vector of each item *i* in the item layer also activates units in the output layer that represents other items to the extent that they are similar to item *i*.

Figure 13

Flow of Activation During Positional Cueing



Note. This figure summarizes the way activation flows from one layer to another, starting from the context layer from which an item is cued with the position it was initially bound to, finishing in the output layer that determines the response.

For instance, consider a memory set of two items *h* and *i* that have been encoded, so that the phonological features of items *h* and *i* are bound to contexts *j* and *k*, respectively. Now memory is cued by position context *k*. This generates a pattern of activation in the item layer consisting of the feature vector of item *i* with weight η_k * 1.0, and the feature vector of item *h* with weight $\eta_j C_{jk}$. Although the feature vectors of the two items are added in the item layer, we can consider their downstream effects separately. The feature vector of each item activates the output unit O_i with a strength corresponding to the similarity between the item's feature vector and the connection weights from the item layer to the output unit. For output unit O_i, the feature vector of item *i* activates it with strength = 1.0 because that feature vector perfectly matches the connection weights leading to O_i. This activation is weighted by the weight of the feature vector in the item layer, which is η_k*1.0. The feature vector of item *h* activates O_i with strength = $Mphon_{hi}$. This is weighted by the weight of feature vector *h* in the item layer, which is C_{jk}. Conversely, output unit O_h is activated by the feature vector of item *i* with strength $Mphon_{hi} * \eta_k$, and by the feature vector of item *h* with strength $\eta_k * C_{jk}$. Generally, Eq. 5 gives the activation accumulated in the output layer for any item *i*, given a context cue *k*:

$$O_i = L_i + \sum_{j=1}^{n} \eta_j Mphon_{ij} C_{jk} \qquad \text{Eq. 5}$$

In this equation, the sum operator runs over the *n* memoranda of the current trial, so that *j* indexes both the serial positions and the items bound to them. The sum collects the contributions of all feature vectors in the item layer to the activation of O_i, each weighted by their similarity to item *i*, and their strength of cueing by context *k*, which is determined by the similarity of *k* to the context *j*, and the strength η_j with which the item is bound to that context. Through this process, retrieval candidates sharing a proportion of features with the retrieved vector will also receive activations in the output layer ***O***. This includes candidates not in the list, and therefore leads to the occurrence of extra-list intrusions. In addition to this activation of output units through cue-based retrieval, the persistent activation of all presented items in the lexical layer, ***L***, is added to the activation of the corresponding units in the output layer.

To implement the omission threshold, we add a further element to the output vector ***O*** representing the strength of activation for an omission:

$$O_{N+1} = \theta \qquad \text{Eq. 6}$$

Here, θ is a free parameter, and the subscript $N+1$ indexes the element of the vector $\boldsymbol{O}$ following the N localist representations. The probability p_i to recall an item is then defined using the exponential version of Luce's choice rule:

$$p_i = \frac{\exp\left(\frac{O_i}{\sigma}\right)}{\sum_{j=1}^{N+1} \exp\left(\frac{O_j}{\sigma}\right)} \qquad \textit{Eq. 7}$$

The σ term represents the standard deviation of noise added to each element in $\boldsymbol{O}$.

After each recall attempt, the connection weights between item and context layers in the weight matrix $\boldsymbol{W}$ are reduced by a proportional factor, which is mathematically equivalent to reducing all binding strengths in the strength vector $\boldsymbol{\eta}$ by the same factor, following this equation:

$$\Delta \boldsymbol{\eta} = -\rho \delta \boldsymbol{\eta} \qquad \textit{Eq. 8}$$

Eq. 8 models output interference (Cowan et al., 1992, 2002; Oberauer, 2003), which occurs via the same automatic updating mechanism as described in Eq. 2. However, the extent to which this updating process occurs is assumed to differ from the encoding stage and is therefore estimated using a free parameter ρ. Finally, items which have already been recalled are discarded from the competition during subsequent retrieval attempts. This is implemented by downgrading this item's activation in the lexical layer:

$$\Delta L_i = -L_i \epsilon \qquad \textit{Eq. 9}$$

The strength of this downgrading mechanisms is controlled by the free parameter ϵ.

Semantic Encoding Mechanisms

To define the similarity between semantically similar items, we extracted items' semantic similarity values using Google news word2vec semantic vectors (https://github.com/mmihaltz/word2vec-GoogleNews-vectors).[4] We did this for each participant and each trial across all experiments. Specifically, we constructed similarity matrices storing the cosine similarity between the vectors of all pair-wise combination of items in each list to be remembered. Hence, for each trial, we have a separate similarity matrix $\boldsymbol{M}_{sem}$ which stores the similarity between item i and all other items j.

Each time an item is encoded, it triggers the activation of its lexical localist unit using Eq. 3. This lexical unit then spreads activation toward the item's semantic features to which it is connected in the semantic portion of the item layer (i.e., the figure and the ground layers). Different from the phonological features, at first, these semantic features are not bound to their context through Hebbian learning. Rather, the connections between active semantic feature units and active context units are tagged. These tags are then subsequently used to guide changes to the connection weights. We describe this tagging mechanism in the next paragraphs.

Learning via Synaptic Tagging

In our implementation, semantic features are not directly encoded via item-context binding when first encountered. Instead, the item-context connections in the weight matrix $\boldsymbol{W}$ are first *tagged* based on the activations in the semantic-figure layer, $\boldsymbol{Z}$, and in the context layer, $\boldsymbol{Y}$.

[4]There is no transparent mapping between dimensions of word2vec vectors and the features we refer to in everyday language. For our model to work, it is not important that the vector dimensions – represented as units in the semantic layers of the neural network – correspond to everyday-language features.

Tagging follows the same Hebbian learning rules as described in Eq. 1, except that instead of changing a connection weight directly, it attaches a tag to it with strength λ_{zy}:

$$\lambda_{zy} = \eta_k Z_z Y_y \qquad \text{Eq. 10}$$

Hence, the strength of a tag depends on the binding strength η_k for the item in list position *k,* but also on the activation in the semantic-figure layer $\boldsymbol{Z}$, determined by the strength of the persistent activation *a* in the lexical layer $\boldsymbol{L}$. The purpose of this tagging mechanism is to indicate to the system the set of potential associations that *could* be formed based on what has just been seen. It is only when sufficient evidence has been accumulated that the tagged associations are transformed into actual item-context associations. This is where the activations in the semantic-ground layer come into play. Whereas the activation pattern in the semantic-figure layer is erased instantly after an item has been encoded, to be replaced by the activation pattern of the next presented item, activation in the semantic-ground layer follows a slower dynamic, gradually accumulating activation across all list items. We describe this dynamic next.

Time-Course of the Activation in the Semantic-Ground Layer

In the semantic-ground layer $\boldsymbol{H}$, a semantic feature *z* for item *i* at serial position *k* receives activation following this equation:

$$\Delta H_{z,ik} = (1.0 - H_{z,ik})(1.0 - T_z) I_z \qquad \text{Eq. 11}$$

This equation ensures that the total activation of the semantic features does not exceed 1.0. The input I_z is filtered by a threshold value T_z such that the input adds to the feature activation to the degree that the threshold falls below 1.0. The input I_z to a semantic feature corresponds to the activation received by the word's phonological representation. Input I_z for item *i* is identical to the input received by the lexical unit L_i, since they get their activation via the same source:

$$I_z = a \qquad \textit{Eq. 12}$$

Each semantic-feature unit *z* in the ground layer has its own threshold T_z. At the beginning of a trial, all threshold values are set to a maximum (i.e., 1.0). From Eq. 11, this means that an item's semantic features in the semantic-ground layer are never activated the first time they receive input. After receiving some input, the values of the thresholds are reduced:

$$\Delta T_z = -T_z I_z \qquad \textit{Eq. 13}$$

This equation reduces the values of the thresholds proportionally to the input received. Due to this, units receiving stronger input are more likely to become activated by input from other semantically similar items during subsequent encoding steps.

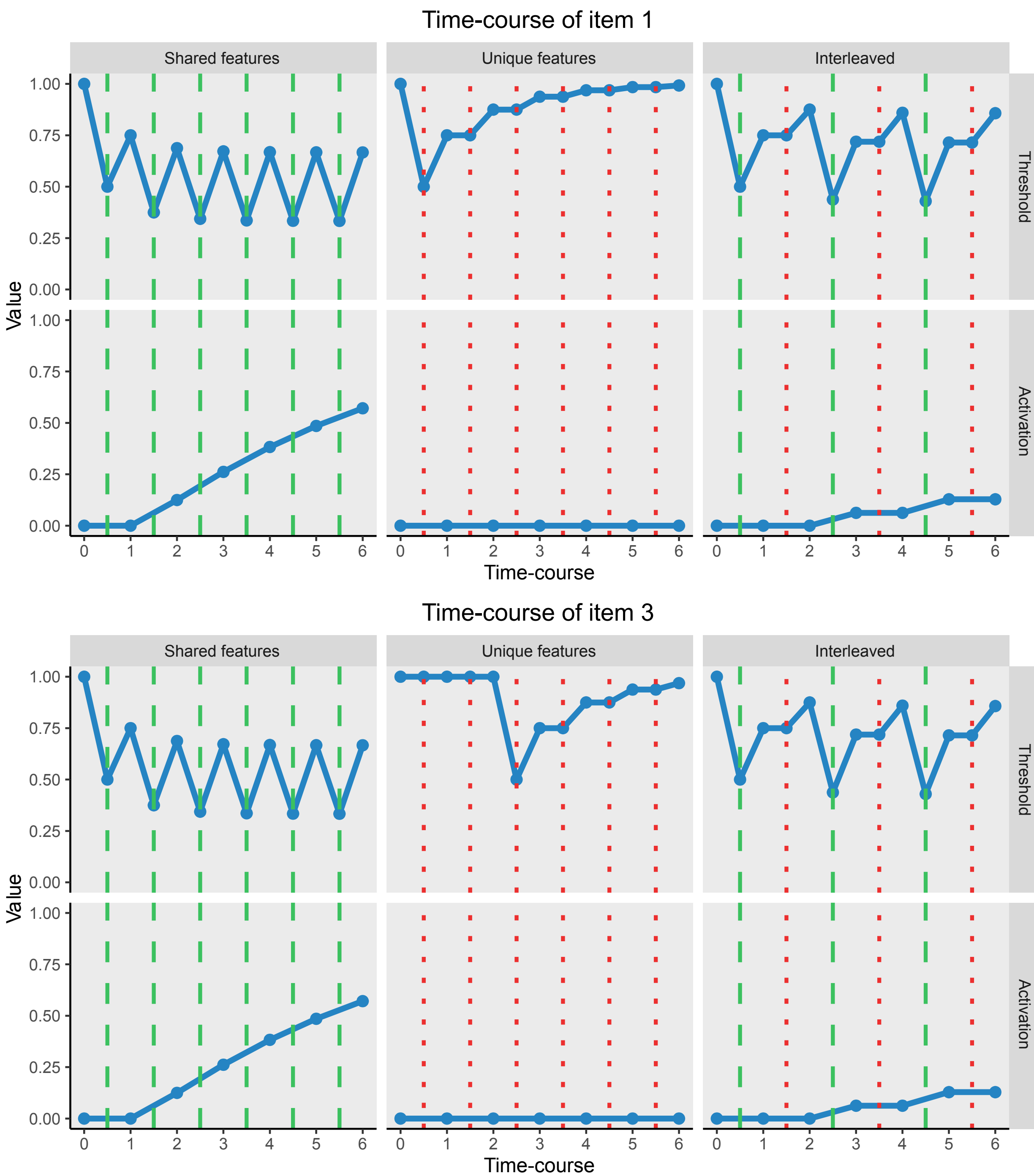
During the presentation of a new item, the thresholds recover half of their value:

$$\Delta T_z = 0.5(1.0 - T_z) \qquad \textit{Eq. 14}$$

This recovery mechanism prevents the items' semantic features from being activated in a context in which they are no longer relevant.

Figure 14

Time-Course of the Dynamic Threshold Mechanism and its Consequences on Items' Activation in the Semantic-Ground layer



Note. The left and middle panels illustrate a list in which all items are semantically similar. We divide the shared (left panels) and unique (middle panels) features, because the dynamics of their activations differ. The right panel illustrates a list in which similar items are interleaved with dissimilar items. In each figure, the top rows indicate values of the thresholds which scale the activation in the semantic-ground layer. The bottom rows indicate activations received by the features in the semantic-ground layer. Green dashed lines indicate moments where shared features between items in the same list receive activation. The red dotted lines indicate moments where (unique) features which are not shared by any other items in the list receive activation. As can be seen, features which are shared by several items in the list receive an increasing amount of activation throughout the trial. This activation does not increase as much in the interleaved condition. In contrast, features which uniquely characterize each item are never activated.

In the mechanism we just described, items' shared semantic features will behave differently from the features uniquely characterizing each item. To visualize this, **Figure 14** illustrates the time-course of the semantic features' thresholds and activations in the semantic-ground layer for the shared (left panels) and unique (middle panels) features of all items in a semantically similar list. The top figure illustrates the time-course for the item encoded first. The bottom figure illustrates the time-course for the item encoded third. In each figure, each panel is divided into two parts: The upper part illustrates the value of the thresholds, and the lower part shows the values of the features' activation. The green dashed lines indicate moments where the shared features receive input. The red dotted lines indicate moments where unique features receive input. Consider what happens to shared and unique features of the third item in a list of semantically similar items, depicted in the left and middle panels, respectively, of the lower figure. In the case of shared features, the presentation of the first item has lowered the third item's threshold values. This occurs because the shared features of items 1 and 3 are, by definition, the same. Therefore, the third item's features already gain activation from other similar items even before its actual presentation. These features continue to receive input throughout the trial from other semantically similar items. Conversely, the unique features never receive any activation (middle panel). Despite the reduction in their threshold values following the item's encoding, subsequent items not sharing these features fail to activate them.

Now, consider the scenario where semantically similar items are interleaved with dissimilar ones, as illustrated in the right panels. For simplicity, we illustrate only the activation values for the shared features (i.e., features shared among the similar items that are presented in every second position), as the unique features are never activated. The presentation of the first item causes the drop of the shared feature's threshold values. When encoding the second item, these features no longer receive input because the second item, belonging to a different category, does not share features with the first. This results in a partial recovery of the threshold values. Consequently, when the third item is encoded, the semantic features that it shares with the first receive reduced activation compared to a situation where the similar items follow each other at successive positions, because of the larger threshold recovery in the interleaved presentation scheme. **Figure 14** also illustrates the time-course of activation of shared and unique features in the semantic-ground layer of item 1 (top figure). As can be seen, in situations where items are semantically similar (left and right panels), the first item still receives some activation through its features shared with other items, although this activation only starts to rise when the subsequent items are encoded.

In sum, unique semantic features are never activated in the semantic-ground layer. The activation of semantic features shared among at least two items is computed using Eq. 11 through 14 after encoding of each new item. Activation in the semantic-ground layer causes changes to the connection weights that have been tagged: Each connection weight W_{zy} is modified according to the strength of its tag, λ_{zy}, multiplied with the activation of the semantic feature unit *z* in the semantic-ground layer $\boldsymbol{H}$:

$$\Delta W_{zy} = H_z \lambda_{zy} \qquad \textit{Eq. 15}$$

With this equation, features having zero activation values in the semantic-ground layer will result in no item-context binding and will therefore not contribute to memory strength. Due to this property, coupled with the threshold mechanism, an item's semantic features which are not shared by any other items in the list will not be encoded into WM via item-context binding, because these features fail to be activated in the semantic-ground layer. In a list in which items are all drawn from the same semantic category, this results in encoding the same semantic features across all items, resulting in the encoding of the category itself. This mechanism predicts better memory performance for semantically similar as compared to dissimilar items, because the additional semantic features encoded in semantically similar lists provide higher activation at retrieval.

Now that we computed the activation value for a particular item given a positional cue for both phonological and semantic information, we can compute the resulting activation of each item in the output layer:

$$O_i = L_i + \sum_{j=1}^{n} \eta_j \left(M_{phon,ij} + \mu_{Hj} M_{sem,ij} \right) C_{jk} \qquad \textit{Eq. 16}$$

with μ_{Hj} for the mean activation level of features of item *j* in the semantic-ground layer $\boldsymbol{H}$. This mean activation depends on the proportion of semantic features that item *j* shares with other items *h*, which is given by $\sum M_{sem,ij}$ and determines which proportion of semantic features of *j* have been activated in $\boldsymbol{H}$ during encoding. It also depends on the number of other items sharing these features, and the positional distance between them, which determines how strongly the shared features of j have been activated during encoding, and hence, the strength with which these features have been bound to the positional context j.

Parameter Estimation

The WM architecture we used has free parameters, which need to be estimated to fit our experimental data. The list of fixed and free parameters is reported in **Table 1**. Model fitting was done for each participant using individual trials, which means that each participant had a different set of parameters. For each recall attempt, we computed the probability p to recall each of the recall candidates using Eq. 7. The log-likelihood was then computed using the recall probability of the observed response o in recall attempt r:

$$logL_r = \log\left(p_{o,r}\right) \quad \text{Eq. 17}$$

We used the deviance as loss function:

$$D = -2.0 \sum logL_r \quad \text{Eq. 18}$$

where the sum operator applies to all trials and retrieval attempts for a given participant.

For instance, suppose the model tries to retrieve the first item in a three-item list. The model might generate a pattern of activation in the output layer ***O*** which looks like this: [1.0 0.3 0.09].[5] Applying Eq. 7 using a noise parameter $\sigma = 0.5$, the probability to retrieve each retrieval candidate is: [0.7099 0.1751 0.115]. If the participant recalled the second item for that particular retrieval attempt, we computed the log-likelihood as $\log(0.1751) = -1.7424$. We then repeated

[5]For this example, we ignore the extra-list retrieval candidates and the omission threshold, which are included as response options in the output layer and therefore also receive a likelihood through Eq. 7.

the process over all retrieval attempts of all trials. To compute the deviance, we summed the log-likelihood over all trials and retrieval attempts for that participant and multiplied this sum by -2.

Parameter estimation was done using the Nelder-Mead algorithm implemented in the Optim package (https://julianlsolvers.github.io/Optim.jl/stable/) of the Julia programming language (https://julialang.org/benchmarks/). The starting points of the gradient descent methods were defined by drawing random values from a uniform distribution, bounded by the minimum and maximum parameter values.

The purpose of the present modeling project is to capture qualitative patterns of results which have been consistently observed in the literature, not to provide a quantitative fit. The fitting procedure used here was applied to get a set of plausible parameter values which enables the model to reproduce important main effects, such as serial position curves, omission errors, extra-list intrusions and order recall performance.

When fitting the data, we use the similarity matrix *Msem* defined for each trial. Each similarity matrix was built by extracting the cosine similarity between vectors representing each item studied for a particular trial using the word2vec GoogleNews corpus. Predictions from the model were obtained by using each matrix for each list participants studied, and then averaging these predictions across trials. Some experiments have been conducted with French-speaking individuals. In these cases, the stimuli were translated from French to English, and similarity values were extracted from this translation.

Our model has a substantial number of free parameters. Most of these parameters are necessary to accommodate the variation of participants' memory performance across the different experimental designs presented in the empirical section.

Table 1. List of Fixed and Free Parameters of the Model.

Symbol	Role	Value
S_1	Phonological similarity between list-items (fixed to 0.1 in lists of phonologically dissimilar items)	0.1
S_2	Phonological similarity between the list-items and all other items	[0.0 – 1.0]
β	Initial value of the encoding strength	[0.0 – 10.0]
π	Progressive decrease in encoding strength	[0.0 – 1.0]
δ	Automatic updating process	[0.0 – 1.0]
a	Base activation of lexical units	[0.0 – 3.0]
P	Positional overlap	[0.0 – 1.0]
θ	Strength of activation for an omission	[0.0 – 10.0]
σ	Standard deviation of noise added at retrieval	[0.0 – 1.0]
ρ	Strength of output interference	[0.0 – 1.0]
ϵ	Strength of downgrading after each retrieval attempt	[0.0 – 10.0]

Note. Fixed parameters are indicated by a single value. Free parameters are indicated by a range.

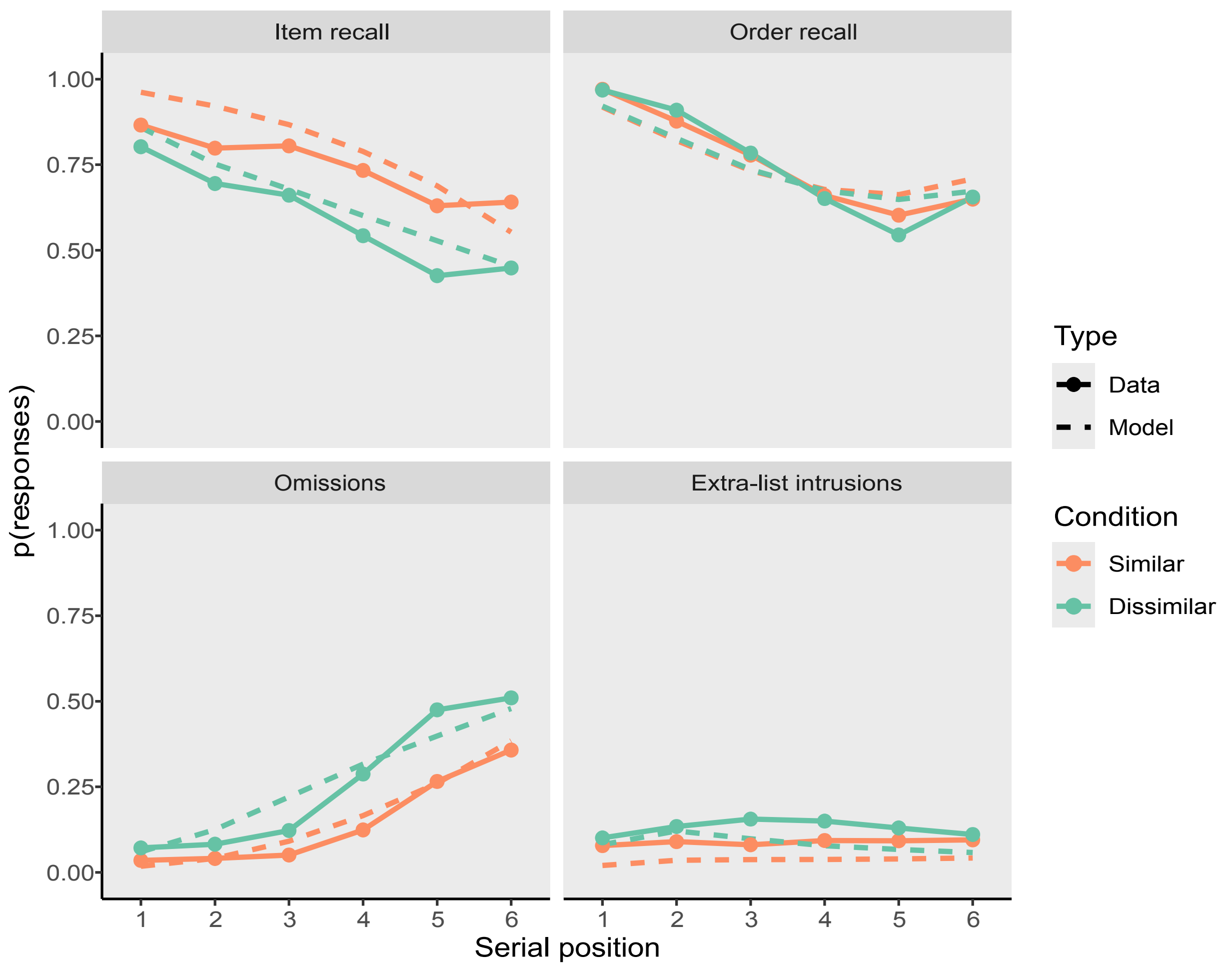
Model Results

Simulation #1 – The General Impact of Semantic Similarity

We first tested the model's ability to simulate the standard semantic similarity effect in immediate serial recall. The model was fitted to the dataset reported in Kowialiewski, Krasnoff, et al. (2023), Experiment 1a. As can be seen in **Figure 15**, the model reproduces the classical pattern of performance characteristic to immediate recall tasks, including serial position curves (primacy and recency) for order errors, omissions, and extra-list intrusions. Importantly, the model predicts the recall advantage for similar vs. dissimilar items (upper left panel). This contribution comes from a reduction of omission errors in the similar condition (lower left panel), but also a reduction of extra-list intrusions (lower right panel). Finally, the model predicts the absence of a semantic similarity effect on order recall (upper right panel).

Figure 15

Semantic Similarity in Immediate Serial Recall – Model Including a Tagging Mechanism

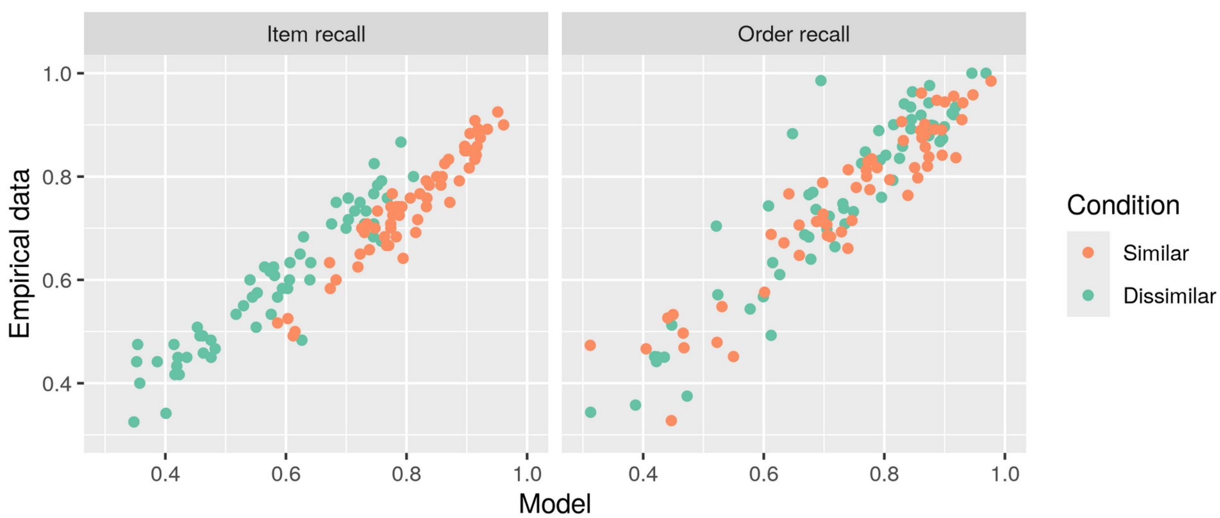


Note. The model reproduces the general semantic similarity effect, with (1) a reduction of omission for lists composed of semantically similar than dissimilar items, (2) a reduction of extra-list intrusions for lists composed similar than dissimilar items, and (3) an absence of impact on order recall. Dashed lines indicate model predictions. The model was fitted to the data reported in Kowialiewski, Krasnoff, et al. (2023), Experiment 1a.

Figure 16 shows the correlation between the model's predictions (x-axis) and the empirical data (y-axis) at the individual level. There is a close match between the model's performance and participants'. This is made possible by fitting the model at the individual level: For participants with low recall performance, the minimization algorithm used for fitting converges toward smaller encoding-strength parameter values, and conversely for participants with high recall performance.

Figure 16

Individual Fits from Simulation #1

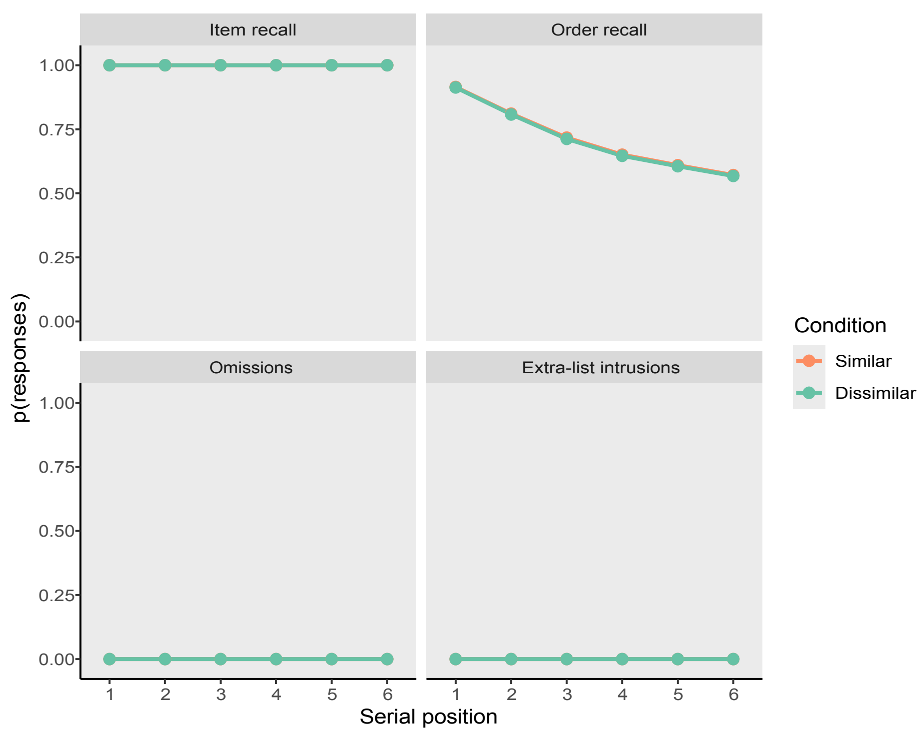


Note. Correlation between the model's performance (x-axis) and the empirical data (y-axis). Left panel: Item recall criterion. Right panel: Order recall criterion.

We extended these simulations by running the model on a reconstruction of order task. In order reconstruction, items are given at retrieval, and participants must reproduce the original sequence's order. To simulate order reconstruction, we (1) restricted the set of recall candidates to the list items and (2) prevented the model from producing omissions (i.e., excluding θ from the set of recall candidates). The same set of parameters as the previous simulation was used. As can be seen in **Figure 17**, the model predicts the absence of a semantic similarity effect on order recall. Because all memoranda are available at retrieval and the model cannot produce omissions, the omission rate is zero. Likewise, as the only available candidates are the list items, the model doesn't produce extra-list intrusions.

Figure 17

Semantic Similarity in Reconstruction of Order – Model Including a Tagging Mechanism

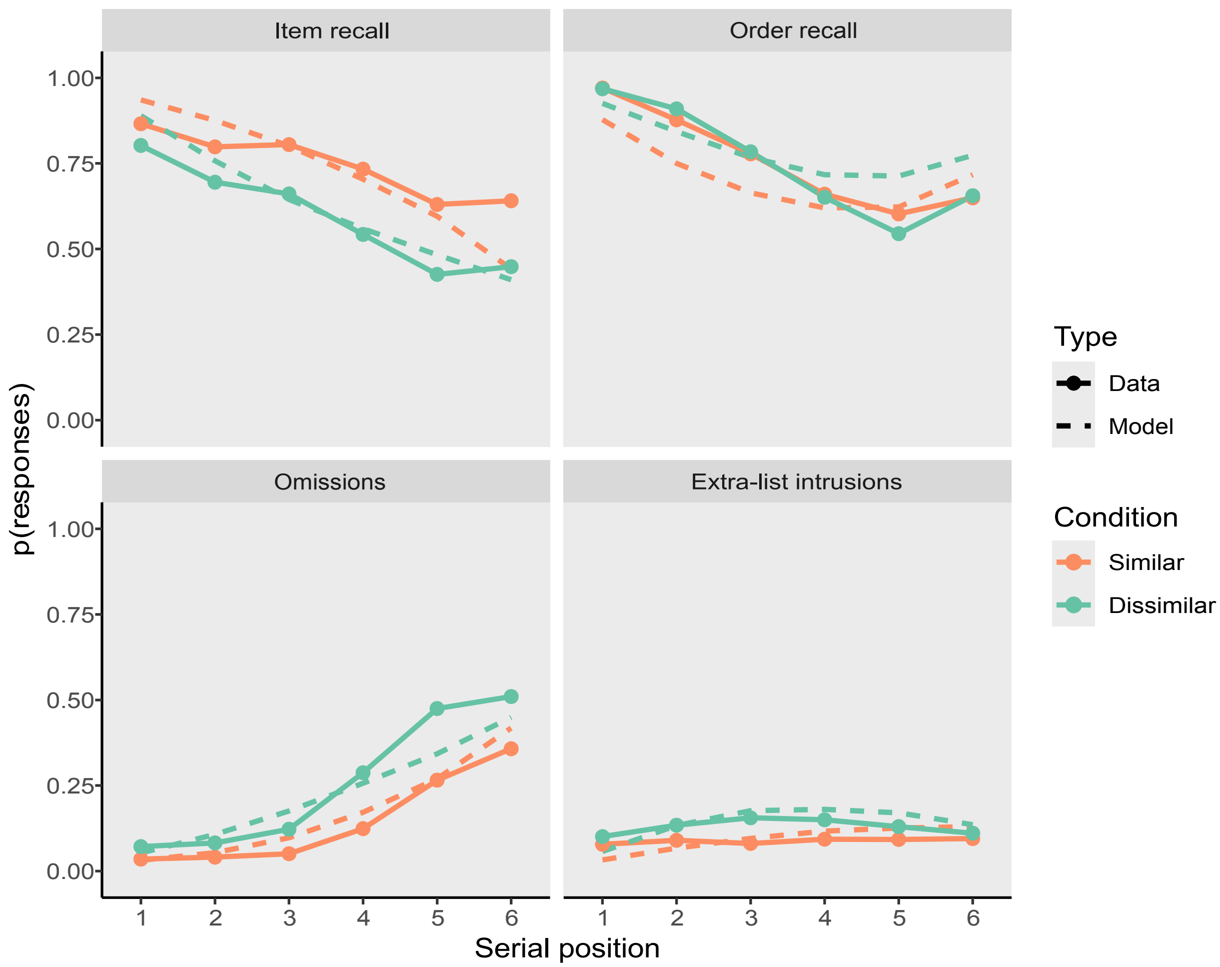


Note. When tested on reconstruction of order, the model successfully reproduces the absence of semantic similarity on order recall. There is no omission and no extra-list intrusion possible in reconstruction of order, because the only items available at retrieval are the memoranda. The model was run using the same parameter values which served to produce results illustrated in **Figure 15**, except that omission errors, repetitions, and extra-list intrusions were not allowed.

We explored whether the model could explain the semantic-similarity effects without the tagging mechanism by deactivating it and then re-fitting the model using the same procedure which served to produce the results in **Figure 15**. As can be seen in **Figure 18**, upper left panel, the model produced a substantial detrimental effect of semantic similarity on order recall. Therefore, it seems that the only way for the model to prevent an increase of order errors following semantic similarity is to include a mechanism which encodes only the semantic features shared by several items, which is precisely what the tagging mechanism does. In the next section, we explain in a more detailed manner the core reasons why such a property is required.

Figure 18

Semantic Similarity in Immediate Serial Recall – Model Without a Tagging Mechanism



Note. Without the inclusion of the tagging mechanism which encodes only features shared between items, the architecture reproduces (1) a reduction of omission for similar lists relative to dissimilar lists, (2) a reduction of extra-list intrusions for similar lists. However, the model fails to account for the absence of a semantic similarity effect on order recall. Dashed lines indicate model predictions. The model was fitted using the dataset reported in Kowialiewski, Krasnoff, et al. (2023), Experiment 1a.

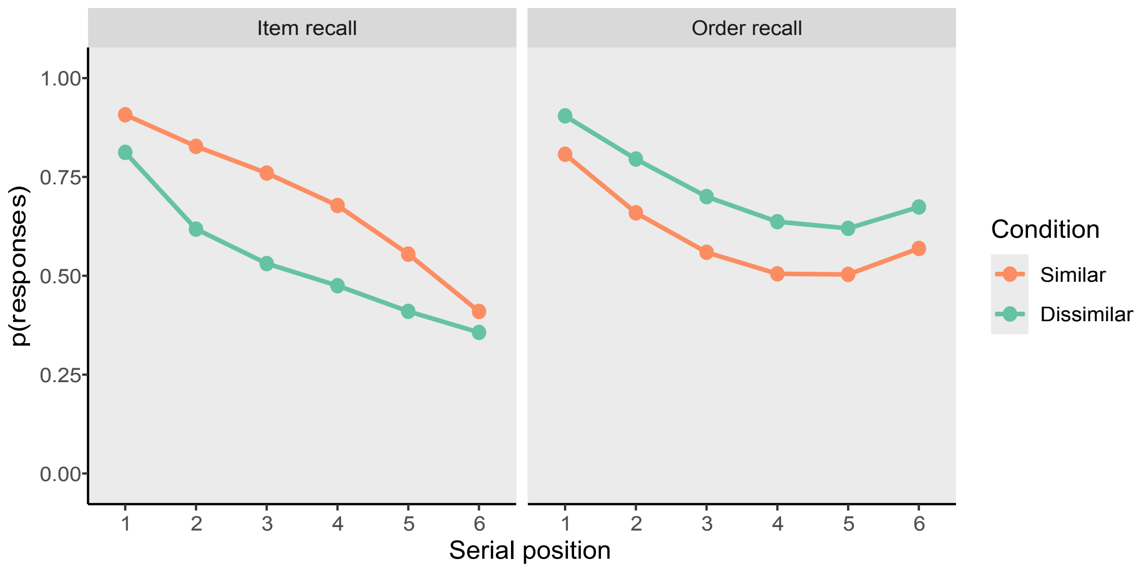
Simulation #2 – Understanding Similarity Effects: Comparison with Rhyming Similarity

The previous simulations show a null effect of semantic similarity on order recall, which means that semantic similarity did not increase confusions errors. To understand why, we need to understand how confusion errors are produced in the model. We illustrate this by manipulating rhyming similarity for comparison. We took the phonological dimension as an example, because this is the best-known phenomenon in the literature. Other dimensions of similarity (for instance, visual similarity; Saito et al., 2008) show qualitatively the same effects on serial recall as phonological similarity.

In this simulation, we varied phonological instead of semantic similarity. To that end we set the value between phonologically similar items to 0.30, while keeping all other parameters from the previous simulations constant. This value is arbitrary and has been set manually. Our purpose was to increase similarity to simulate phonological-similarity effects qualitatively, not to quantitatively reproduce the empirical data. To keep the model’s behavior easy to track, we also deactivated the contribution from the semantic part of the items. As can be seen in **Figure 19**, left panel, the model correctly predicts the recall advantage for rhyming vs. non-rhyming items on item recall, a standard observation (Fallon et al., 2005; Gupta et al., 2005; Neale & Tehan, 2007; Nimmo & Roodenrys, 2004). At the same time, phonological similarity also impairs order recall (right panel).

Figure 19

Rhyming Similarity in Immediate Serial Recall – Model Predictions

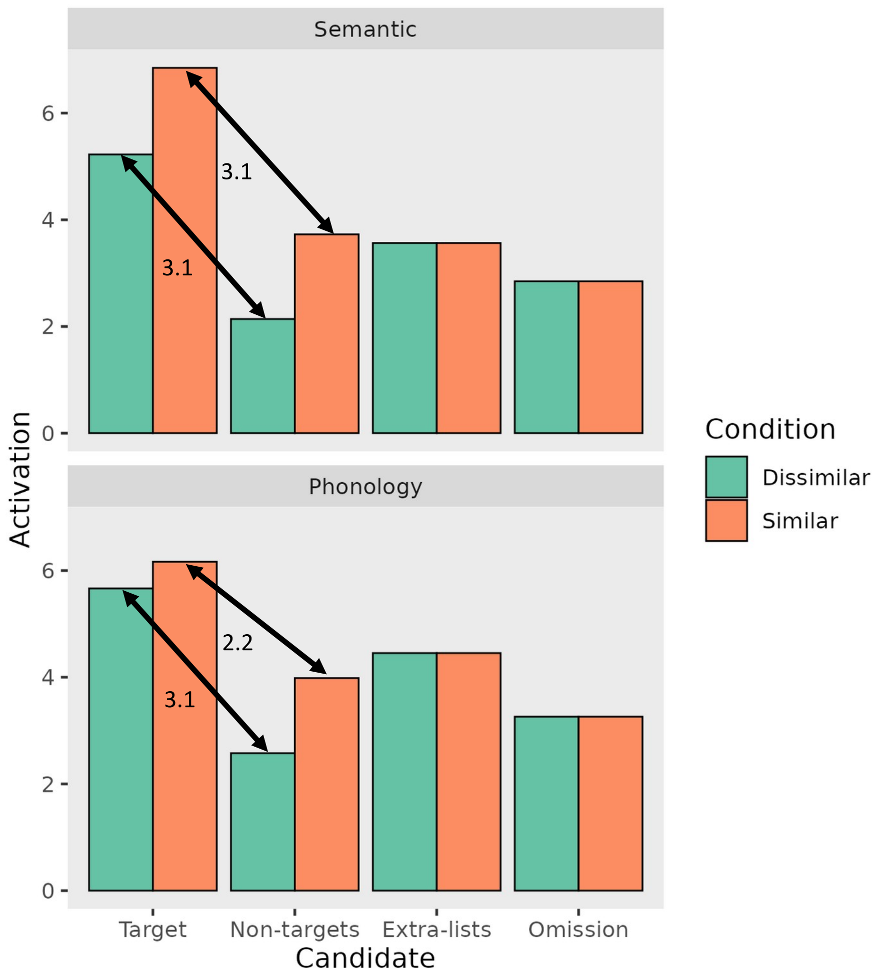


Note. When manipulating rhyming similarity, the model correctly captures the pattern found in humans: an item recall advantage for lists composed of rhyming vs. non-rhyming items, and a detrimental effect on order recall. These results were simulated by increasing the phonological similarity value between items to 0.3 in the similar condition, while keeping all the other parameters from **Simulation #1** constant, and deactivating the contribution from the semantic part of the items.

To understand this discrepancy between semantic and phonological similarity, we computed the items' activation at retrieval. Each time the model attempts to retrieve an item, the activation associated with the target item, non-target items, extra-list items, and the omission threshold were extracted. In the mathematical description presented above, this corresponds to Eq. 16. As can be seen in **Figure 20**, similar items have a higher activation level than dissimilar items. Thanks to this high activation level, similar items are more often selected than non-list items, and more often pass the omission threshold, compared to dissimilar items, thus producing an item recall advantage. This happens both in the semantic and phonological domains, though for different reasons. Let's start with the semantic domain.

Figure 20

Activation Values Produced by the Model at Retrieval



Note. Activation values for the target item, non-target items, extra-list items, and omission threshold. Values are averaged across all retrieval attempts. Upper panel: semantic similarity variation. Lower panel: phonological similarity variation. Due to their higher activation value, similar items can overcome the activation values of extra-list items and the omission threshold more often than dissimilar items, thus producing a net benefit on item recall. Confusion errors between target and non-target items increase as the difference between their activation values decreases. This difference remains identical between semantically similar and dissimilar items, thus preventing an increase in confusion errors between semantically similar (relative to dissimilar) items. When phonological similarity is varied, this difference between target and non-target items is smaller in the similar compared to the dissimilar condition, which increases confusion errors.

When list items are semantically similar, they receive additional activation, which comes from the encoding of a categorical representation that is the same for all items of a similar list. This categorical representation adds a constant amount of activation to all memoranda at retrieval for semantically similar lists. Consequently, the difference in activation between the target and the non-target items remains constant. This can be seen in the upper panel of **Figure 20**, where the relative activation level between target and non-target items is identical in the semantically similar ($M_{diff} = 3.1$) and dissimilar ($M_{diff} = 3.1$) conditions. In the choice rule that we use for response selection (Eq. 7), a constant difference in activation levels translates into a constant proportion of exponentiated activation levels. Let's assume a vector v which contains activation values for n items. The probability to recall each item is given by:

$$p_i = \frac{\exp\left(v_i\right)}{\sum_{j=1}^{n} \exp\left(v_j\right)} \qquad \textit{Eq. 19}$$

Now let's add a constant value c to each index in the vector v:

$$p_i = \frac{\exp\left(v_i + c\right)}{\sum_{j=1}^{n} \exp\left(v_j + c\right)} \qquad \textit{Eq. 20}$$

We can factor out e^c from both the numerator and denominator:

$$p_i = \frac{e^c . \exp\left(v_i\right)}{e^c . \sum_{j=1}^{n} \exp\left(v_j\right)} \qquad \textit{Eq. 21}$$

In Eq. 21, both e^c terms cancel each other, which brings us back to Eq. 19. Therefore, by adding a constant boost of activation to all semantically similar items, our semantic similarity mechanism based on the encoding of categorical information has no effect on the probability of confusing a target item with one of the non-target items in the list. This means that in this model, semantic similarity has no effect on order recall.

Psychologically speaking, this property of the exponential version of Luce's choice rule makes sense: When a constant representation, such as a taxonomic category, is added uniformly to all items in a list (in the present model, this is achieved by keeping only the features shared between list items active in the semantic ground layer), it results in equal discriminability between the items. In this scenario, semantic features are not informative to discriminate between the list items.[6]

[6] As a metaphor, let's consider a situation in which one is asked to identify a criminal among several suspects. To differentiate between the suspects, one can rely on distinguishing cues such as age, facial characteristics, or voice pitch. However, if all the suspects in question happen to be white, recalling that the criminal was white is uninformative. Similarly, knowing that the items were drawn from the category of fruit is uninformative to discriminate between apple, banana, and orange.

The effects of phonological similarity on item and order recall are different from those of semantic similarity, because we assume that phonological features are activated directly by the input they receive at encoding, and therefore, all phonological features of an item are bound to that item's context.

Phonological similarity increases items' activation at retrieval through a different route: When the target item is cued by its positional cue, a pattern of activation is generated in the item layer that is a weighted blend of all list items, each weighted by the similarity of its position to the target position. This vector is then compared to all candidate items in long-term memory. When the item vectors are similar to each other, the target item in long-term memory is similar to some degree to all list items that contribute to the retrieved vector. Therefore, the target item receives higher activation in the output layer at retrieval when list items are phonologically similar. Mathematically, it can be seen from Eq. 16 that when values of *Mphon* get large, the sum of $Mphon_{ij} * C_{jk}$ increases, too.

As an example, let's assume for simplicity a two-item list, with a phonological similarity value of 0.1 and 0.35 for a dissimilar and similar condition, respectively. Retrieval of the first item is done by cueing the WM representation using cue 1, which has a similarity value of 0.5 with cue 2. The current cue is maximally similar to itself (i.e., similarity value of 1.0). In this scenario, item 1 is maximally reactivated by the first cue, resulting in an activation value of 1.0 * 1.0 in the output layer. In addition, item 2 is also cued to some extent (i.e., with a value of 0.5) by virtue of being partially associated to cue 1. Because item 2 is similar to item 1, this results in the partial activation of item 1 through its shared features. Hence, item 1 is reactivated by 0.1 * 0.5 in the dissimilar condition, and 0.35 * 0.5 in the similar condition. The activation value of the first item in the output layer is then the sum of the activation generated by cue 1. In the dissimilar condition, this gives:

$$O_1 = 1.0 * 1.0 + 0.1 * 0.5 = 1.05$$

In the similar condition, this gives:

$$O_1 = 1.0 * 1.0 + 0.35 * 0.5 = 1.175$$

Thus, this way of binding items to context results in higher activation values for lists of similar items. The same effect, however, also applies to non-targets: Their long-term memory representations are also similar to all list items, and therefore they receive activation from the contribution of all list items to the retrieved vector. Critically, the non-targets receive activation from the target item, which enters the retrieved vector with the highest weight, whereas the target receives activation only from the non-targets, which enter with a lower weight. Thereby, non-targets receive a larger boost to their activation in the output layer from phonological similarity than targets do. Therefore, the activation difference between targets and non-targets decreases in phonologically similar lists.

Let's return to our 2-items list example. When the first item is cued using cue 1, item 2 gets re-activated by the current cue proportionally to its similarity value with item 1. This gives 0.1 * 1.0 in the dissimilar list, and 0.35 * 1.0 in the similar list. In addition, item 2 gets activation by virtue of being associated to cue 2. This gives 1.0 * 0.5 in both conditions. Thus, activation value in the output layer for the non-target item in the dissimilar condition becomes:

$$O_2 = 0.1 * 1.0 + 1.0 * 0.5 = 0.60$$

In the similar condition:

$$O_2 = 0.35 * 1.0 + 1.0 * 0.5 = 0.85$$

In our simplistic scenarios, if we take the difference in activation value between the target and non-target items, we have $M_{diff} = 0.45$ in the dissimilar condition, and $M_{diff} = 0.325$ in the similar condition. This is also observed in our simulations: In the phonologically similar condition, the difference in activation level (see **Figure 20**) is smaller ($M_{diff} = 2.2$) than in the phonologically dissimilar condition ($M_{diff} = 3.1$). In this context, the property as described in Eq. 19 through 21 no longer applies. This unequal difference in activation translates into reduced distinctiveness between target and non-target items in the phonologically similar condition. Therefore, the model produces more order errors in phonologically similar than dissimilar lists.

Simulation #3 – Semantically Similar Retrieval Cues do not Lead to Increased Interference

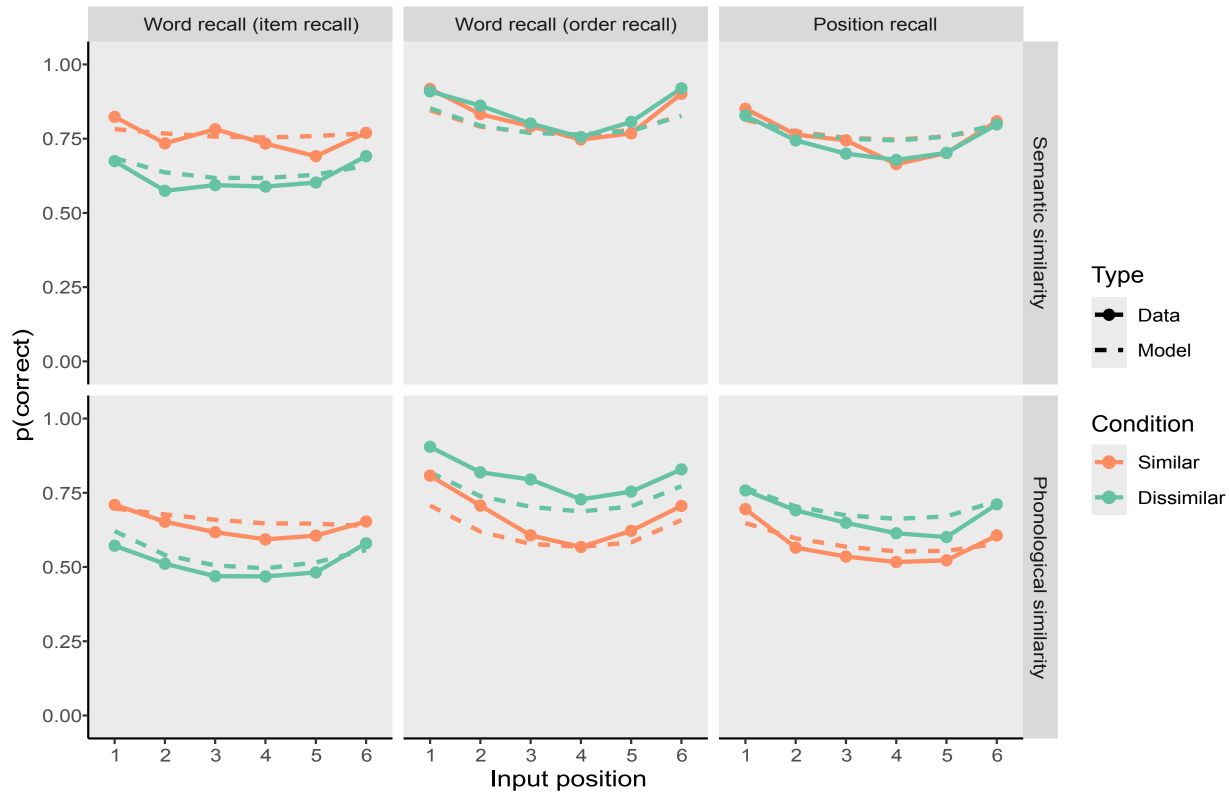
When participants are cued with an item and must retrieve the position associated to it, phonological similarity increases the occurrence of confusion errors, but semantic similarity does not (Kowialiewski et al., 2023). In the study from Kowialiewski, Krasnoff and colleagues, participants retrieved items from their context on half the trials, and retrieved contexts from the items on the other half of the trials. We therefore simulated both retrieval directions. In addition, all items and positions were tested in random order, an aspect of the experimental procedure we also simulated. To retrieve items from their contexts, we used Eq. 16. For the opposite direction the item layer and the context layer switch roles: The given item is activated in the item layer (i.e., the phonological layer $\boldsymbol{X}$ and the semantic figure layer $\boldsymbol{Z}$), which re-activates a distributed representation of the position through the weight matrix $\boldsymbol{W}$. The re-activated position representation is forwarded to an output layer $\boldsymbol{\Omega}$ with localist representations of the positions. The combined phonological and semantic similarity matrices now play the role of similarities between retrieval cues, and the cue-similarity matrix $\boldsymbol{C}$ takes the role of similarities between retrieval candidates. Hence, we adapted Eq. 16 as follow:

$$\Omega_k = \sum_{j=1}^{n} \eta_j C_{jk} \left(M_{phon,ij} + \mu_{Hj} M_{sem,ij} \right) \qquad \textit{Eq. 22}$$

When modeling the retrieval direction from a given item to its position, we restricted the set of retrieval candidates to the 6 positions and set the θ parameter to zero because in this task, what needs to be retrieved (i.e., the positions from 1 to 6) is always known to the participants, thus preventing omissions and extra-list intrusions. This also implies that items' localist semantic units no longer play a role, which is why the L_i term doesn't appear in Eq. 22. We fitted the model's parameters to the data reported by Kowialiewski et al. (2023), Experiment 2a & 2b. Results of these simulations are displayed in **Figure 21**.

Figure 21

Simulation Results from the Cue-Similarity Manipulation



Note. Upper panels: Semantic similarity. Lower panels: Phonological similarity. Left and middle panels: Item and order recall, in the conditions involving the retrieval of items from contexts. Right panels: Positional recall, in the conditions involving the retrieval of positions from items. Dashed lines indicate model predictions. The model was fitted to the data reported by Kowialiewski et al. (2023), Experiment 2a & 2b.

In agreement with the experimental data, the model produces bow-shaped serial position curves instead of the strong primacy effect usually observed in serial recall[7]. The important

[7]In serial recall, input position is fully confounded with output position: The last encoded items are also output last. A significant part of forgetting occurs in WM due to output interference (Cowan et al., 2002). Therefore, in serial recall, the last presented items also suffer most from output interference, which creates a strong primacy effect. In contrast, when items are cued in random order, the effect of output interference is equally spread over all positions. This causes more symmetrical recall performance across serial positions.

results are those related to the similarity manipulations. As can be seen in **Figure 21**, left panels, the model predicts increased item recall for similar vs. dissimilar lists, for the same reasons as explained in simulations #1 and #2. When items are retrieved from contexts, phonological similarity increases confusion errors for the reasons explained in simulation #2. When contexts are retrieved from items, phonological similarity increases confusion errors for a different reason. If the presented cue is similar to other cues, this leads to a stronger activation of the other non-target contexts, increasing the probability to choose another context than the target one.

As can be seen, semantic similarity has no effect on confusion errors for this retrieval direction. In our model, the semantic part of the item that is activated in the item layer functions as a retrieval cue only insofar as it is bound to the item's position in the weight matrix $\boldsymbol{W}$. This is the case only for semantic features that are shared among multiple items. In dissimilar lists, such shared features hardly exist; in similar lists, the shared features are identical for all items, and therefore they are bound equally to all list positions. Hence, they cannot be used to discriminate one position from another. They add a constant amount of activation to all positions, without changing the difference between target and non-target positions, leading to no increase of confusion errors.

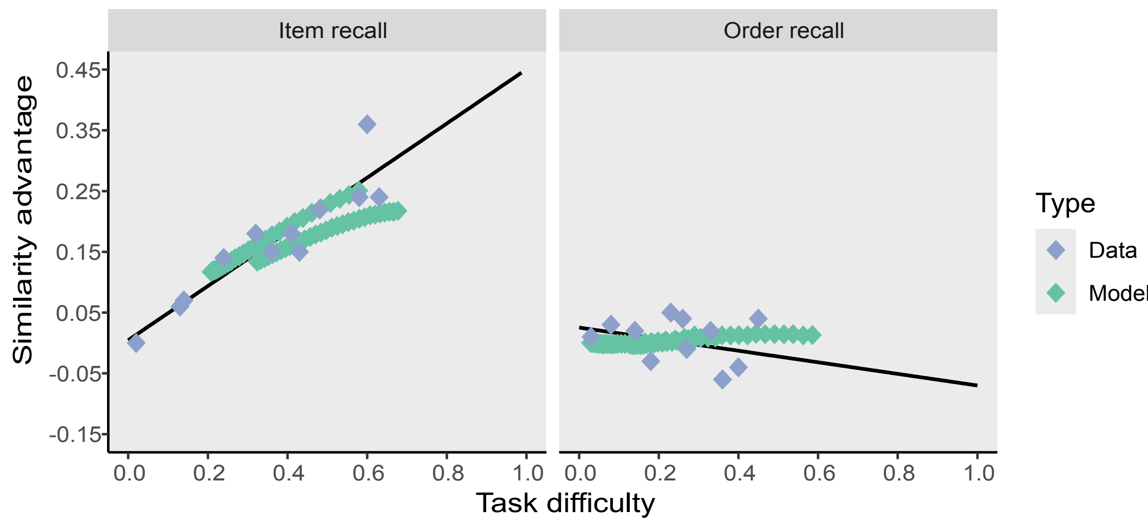
Simulation #4 – Semantic Similarity and Task Difficulty

Serial recall of semantically similar lists is more resistant to task difficulty manipulations than recall of semantically dissimilar lists (Kowialiewski & Majerus, 2020; Neale & Tehan, 2007). We simulated Neale and Tehan’s results, who observed that the magnitude of the semantic similarity effect on item memory gradually increased as memory performance decreased. Interference by a secondary task (e.g., concurrent articulation) was implemented in the model by scaling the encoding strength vector $\boldsymbol{\eta}$ with values ranging from 1.0 to 0.5. This variation of encoding strength affected only the binding of phonological features to their positional contexts. A value of 0.0 means that the content of WM was completely erased, and a value of 1.0 leaves the WM representations unaffected. We reduced the encoding strength specifically for the phonological representation, as the interfering tasks included by Neale and Tehan were phonological in nature (i.e., articulatory suppression, backward counting). Note that Neale and Tehan's experimental setup included two set size conditions: 4 and 6. These conditions were also included in the current simulations. For each set size, we simulated 30 task difficulty conditions. Note that Neale and Tehan (2007)'s raw data were not made available. For this reason, we simulated these results by using the parameter values which served to generate those of simulation #1. As can be seen in **Figure 22**, the model predicts the increased similarity advantage on item recall as task difficulty increases. In contrast, the model doesn't predict any similarity effect on order recall, and this absence is consistent across all task difficulty levels.

The reason why the semantic-similarity benefit increases with higher task difficulty is rather trivial, and most likely not specific to our model: The poorer item recall becomes in the dissimilar condition, the more room there is for improvement through semantic similarity. For order recall this does not happen because semantic similarity has no influence on order recall.

Figure 22

Simulations of the Task Difficulty Effect



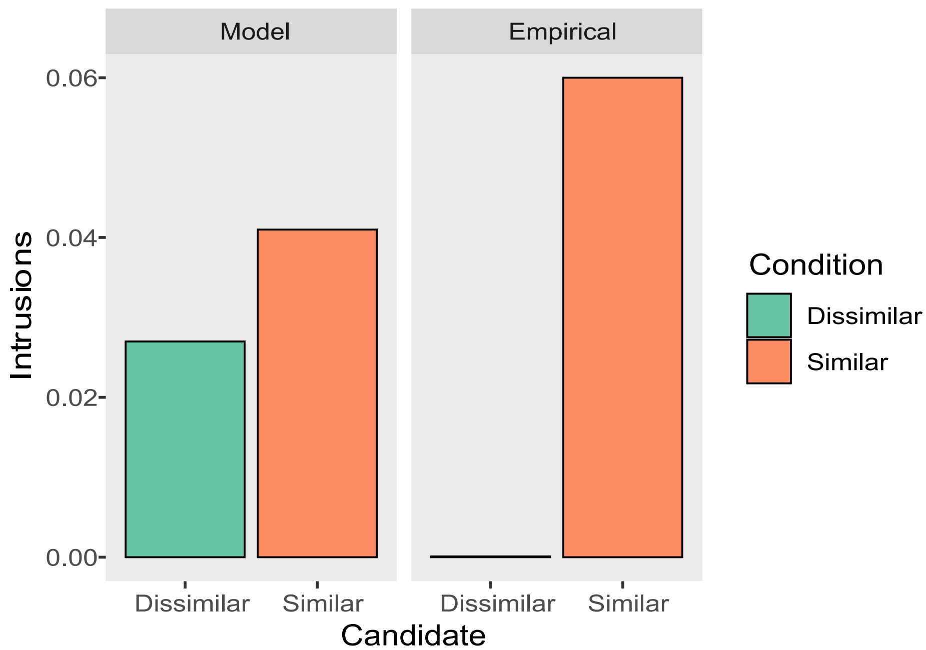
Note. Task difficulty was computed as the decrease in recall performance for a given score (item recall or order recall). Results were simulated by using the same parameter values as in Simulation #1.

Simulation #5 – Semantic Similarity Modulates the Type of Intrusion Errors

In lists composed of semantically similar items, participants frequently recall a critical lure that is highly similar to all list items but is not included in the list. This phenomenon is rarely observed in lists composed of dissimilar items (Tehan, 2010). To simulate this effect, we extracted the word2vec similarity values of 24 lists from the Stadler et al. (1999) norms, which Tehan (2010) based his study on. The 6 strongest associates to the critical lure were chosen. We extracted the similarity values between these list items and the critical lures, and included these values in the similarity matrix *Msem*. We used the same parameter values as Simulation #1. Results from this simulation, reported in **Figure 23**, show that the model accounts reasonably well for this phenomenon: The critical lure is recalled much more often in the similar than the dissimilar list. Whereas in the experiment the critical lure was practically never recalled in the dissimilar condition, the model still produced 2.6% critical-lure intrusions. This is probably due to the fact that we gave the model a much smaller vocabulary than adult human participants have, so that the critical lure has a higher chance of being selected whenever an extra-list intrusion occurred.

Figure 23

Recall of Critical Lures as a function of Semantic Condition



Note. The y-axis shows the proportions of times critical lures were recalled, out of the total number of responses produced. The data were simulated by using the same parameter values as in Simulation #1, and using the word2vec similarity values from the Stadler et al. (1999) norms.

We also simulated the distribution of extra-list intrusions as a function of list composition (semantically similar and dissimilar lists) as shown in **Figure 5**. We started with the assumption that most retrieval candidates in one's vocabulary are only weakly associated with their target items, meaning that the overall probability of producing a semantically dissimilar intrusion is higher than that of producing a semantically similar intrusion. To model this, we first estimated the proportion of words in the language that have low, medium, and high similarity to target items in the memory list. We did this by computing the similarity between each target and the available items in word2vec. These similarity values were used to categorize words in the vocabulary into each bin, and the count in each bin was then divided by the sum of all counts. Given the large number of words available in word2vec (3 million), we drew 10 words at random for each target, and repeated the process for all list items in all experiments. The resulting proportions were then used as a weighting factor in Luce's choice rule (Eq. 7) to simulate the probability of retrieving an intrusion from each similarity bin. Specifically, we decomposed the probability of retrieving a non-target item into non-targets with low, medium, and high similarity to the target, each weighted by the proportion of words in the vocabulary belonging to each of these three bins. Adapting Eq. 7 gives:

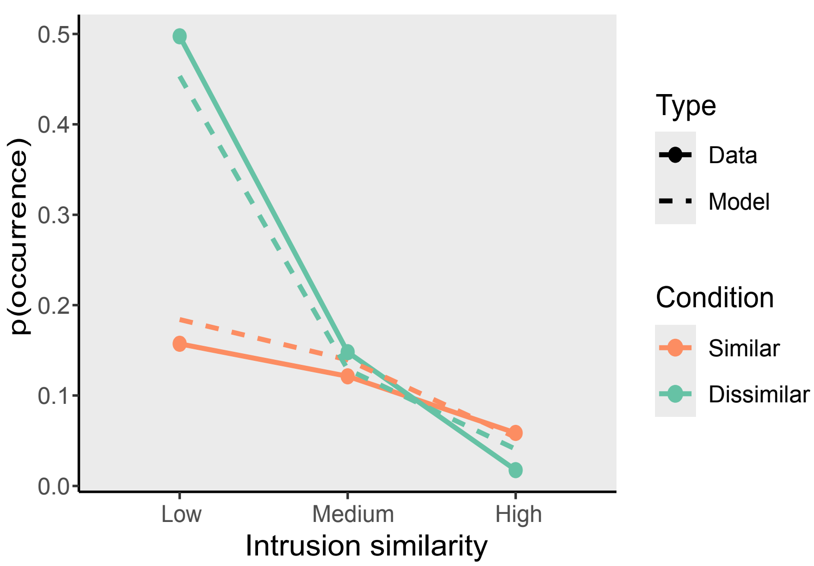
$$p_i = \frac{\exp\left(\frac{O_i}{\sigma}\right) w_i}{\sum_{j=1}^{N+1} \exp\left(\frac{O_j}{\sigma}\right) w_j} \qquad \textit{Eq. 23}$$

Where $\boldsymbol{w}$ is a vector of weights whose values are fixed to 1.0, except for non-list items $n+1:n+3$ for which the $\boldsymbol{w}$ was set to the proportion of words in each bin. For those retrieval candidates, we kept the original phonological similarity value S_2 as used in previous simulations. By this method, the sum of $p_{n+1:n+3}$ gives back the original probability to retrieve a non-list item as found in our previous simulations. Additionally, the model was assigned distinct semantic similarity values between targets and non-target items for each bin. These semantic similarity values were determined using the median semantic similarity values between targets and intrusions observed in the empirical data reported in **Figure 5**. Results of these simulations are reported in **Figure 24**.

The model reproduces the observed decrease in low-similarity intrusions, and the corresponding increase in high-similarity intrusions, in semantically similar lists. The decrease in low-similarity intrusions reflects the overall reduction in extra-list intrusions in semantically similar lists: because list items receive a larger boost of activation in the output layer (***O***) in a semantically similar list, they are more likely to be recalled than non-list items.

Figure 24

Distribution of intrusion errors

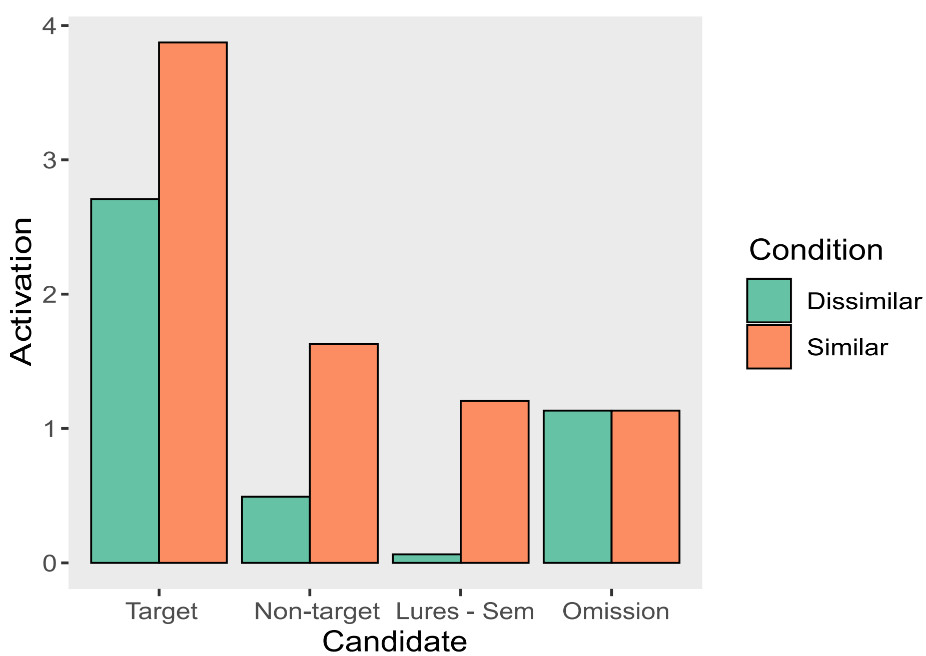


Note. Proportion of intrusion errors as a function of semantic similarity (similar vs. dissimilar).

The reason why highly similar intrusions occur more often in similar than dissimilar lists can be better understood by looking at the activation values extracted at retrieval, which we report in **Figure 25**. For simplicity, the analysis focuses on the simulations from the Tehan (2010) data reported in **Figure 23**, but this theoretical explanation also applies to the simulations reported in **Figure 24**. We divided the activation values in different categories: Target items, non-target items, the critical lures, and the omission threshold. In a semantically dissimilar list (upper panel), the semantic part of the items is not encoded, because few features are shared between list items. This means that the semantic part of the representation has little contribution when it comes to select a recall candidate. Consequently, the lure item will receive very little activation in the output layer ***O*** and will therefore be rarely selected. In contrast, in a semantically similar list the semantic part of the representation contributes strongly to the selection process. When trying to retrieve an item, the semantic features of this item will match those of the lure items, which means that the lure will receive a substantial amount of activation in the output layer O. This leads the model to more often select extra-list items sharing features with the list items.

Figure 25

Activation Values Produced by the Model at Retrieval



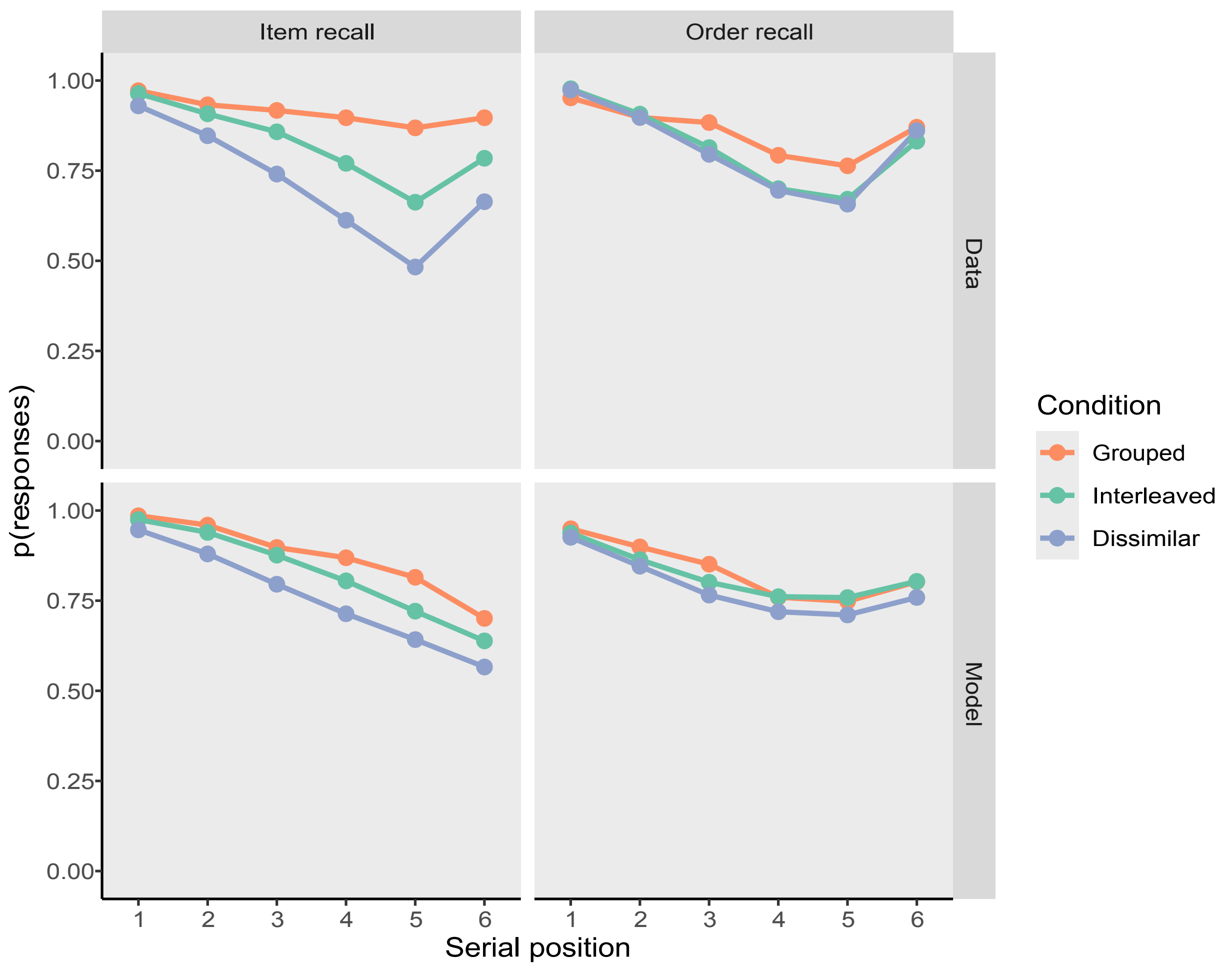
Note. Activation values for the target item, non-target items, critical lures, and omission threshold. Values are averaged across all retrieval attempts. In the dissimilar condition, the semantic part of the items has little contribution during the retrieval stage. As a consequence, critical lures (Lures-Sem) receive little activation and have a low probability to be recalled. In the similar condition, in contrast, critical lures receive high activation values due to their shared semantic features with the list items, which makes them more likely to be erroneously recalled.

Simulation #6 – The Separation Effect

In the separation effect, semantically similar items are better recalled if encoded at close vs. distant serial positions. We fitted the model to the data reported in Kowialiewski, Majerus, et al. (2023), Experiment 1. The experiment involved three conditions: grouped, interleaved, and dissimilar. In the grouped condition, items were presented in two sub-groups of three semantically similar items (i.e., AAABBB). Lists in the interleaved condition also consisted of two sets of three similar items, except that the similar items were presented in an interleaved fashion (ABABAB). The dissimilar condition involved lists composed of items drawn from six different categories (ABCDEF). We fitted the model to these data using the same procedure as described above. The semantic similarity value S_3 was identical in the grouped and interleaved conditions, as the two conditions were constructed using the same semantic categories. As can be seen in **Figure 26**, the model predicts the decreased recall performance in the interleaved compared to the grouped condition. This pattern is made possible via the dynamic activation threshold as computed in Eq. 11 through 14: When several semantically similar items are presented one after the other, the thresholds of their semantic features become lower. Once these thresholds are lowered, the items' semantic features receive more activation from similar items. When similar items are interleaved, the thresholds recover more strongly towards their initial value in between presentations of similar items. Therefore, when encoding the third and fourth semantically similar item, the semantic features receive less activation than if the semantically similar items were presented close to each other. The model also predicts the order recall advantage (see **Figure 26**, right panel) in the grouped vs. dissimilar and interleaved condition. This result is deeply linked to a fundamental property of our model, which we analyze in greater details in the next section.

Figure 26

Simulations of the Separation Effect



Note. Upper panels: Empirical data. Lower panels: Model predictions. The model was fit to the data reported in Kowialiewski, Majerus, et al. (2023), Experiment 1.

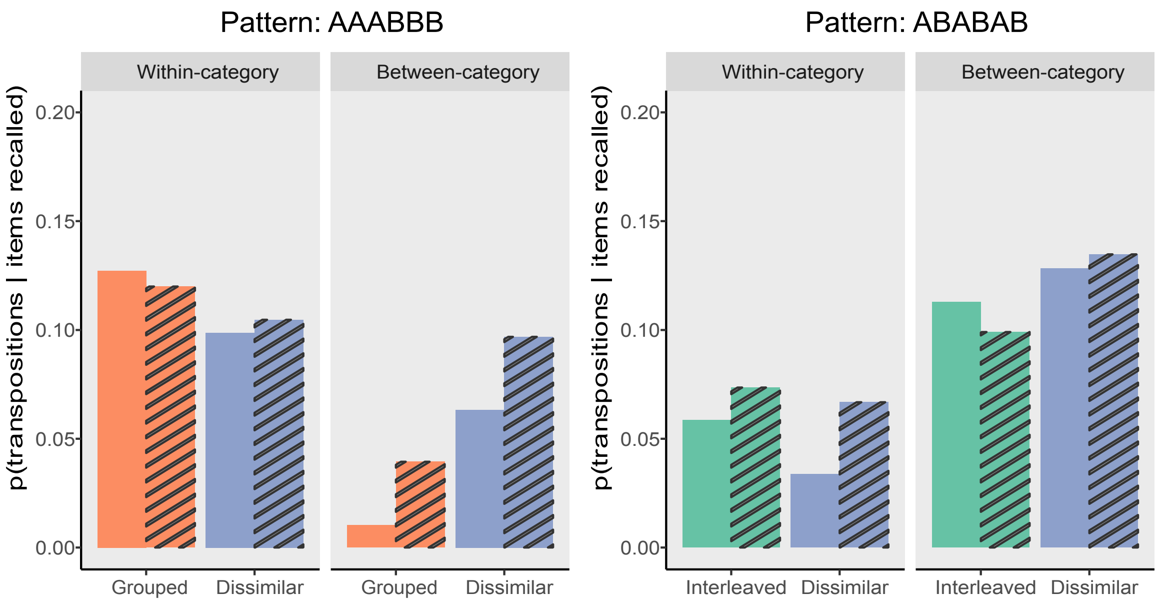
Simulation #7 – Semantic Similarity Constrains Order Errors

When semantically similar items are presented in semantic subgroups (e.g., "piano, flute, guitar, *leopard*, *cheetah*, *lion*"), order errors tend to follow the semantic structure imposed by the experimental setup: People are more likely to transpose two semantically similar items than two semantically dissimilar ones. When the similar items are presented in an interleaved fashion (e.g., "piano, *leopard*, flute, *cheetah*, guitar, *lion*"), the effect is still observed, although only half as large as in the grouped condition. We report the same simulations as those in #5, as they involve the same experimental conditions. As can be seen in **Figure 27**, our model based on the encoding of categorical information captured the pattern of transposition errors induced by the lists' semantic structure.

Our model can capture this benchmark, because it encodes items' semantic features in the grouped and interleaved conditions. The semantic structure prevents transposition errors from one group to another, because the semantic features encoded for items from distinct categories mismatch. For instance, when attempting to retrieve "item 3" in a "AAABBB" list, the semantic features of category "A" of this item are reactivated. When comparing this representation to all representations stored in long-term memory, the semantic representation of "item 3" will strongly match with those of the same category, in this case items 1 through 3, thereby slightly increasing within-category transpositions. In contrast, the representation of this item will mismatch with those of items 4 through 6, which strongly reduces between-category transpositions. This reduction of between-category transpositions explains why order recall is better in the grouped compared to the dissimilar condition, as reported in **Figure 26**.

Figure 27

Transpositions as a Function of Semantic Similarity Structure and Transposition Type



Note. Striped bars indicate model predictions. The model was fit to the data reported in Kowialiewski, Majerus, et al. (2023), Experiment 1.

To better understand what is happening in the model from a mathematical perspective, we can start from a basic example. Suppose a first scenario in which the model tries to retrieve the first item among a list of four dissimilar items to be remembered. Assuming a positional overlap of 0.333, items' activation value for this retrieval step is: [1.0, 0.333, 0.111, 0.037]. Applying the choice rule we reported in Eq. 7 with $\sigma = 0.5$, the probability to recall each item is: [0.634, 0.167, 0.107, 0.092]. For this retrieval attempt, the proportion of within and between-category transposition among all transpositions is 0.801 and 0.199, respectively.

Now suppose a second scenario in which both items 1 and 2 receive a uniform boost of activation of 0.5, which is what typically occurs in the model when two items are semantically similar (i.e., both items receive a constant boost). Items' activation value is now: [1.5, 0.833, 0.111, 0.037], and their probability to be retrieved is: [0.725, 0.191, 0.045, 0.039]. The proportion of within and between-category transposition is now: 0.916 and 0.084, respectively.

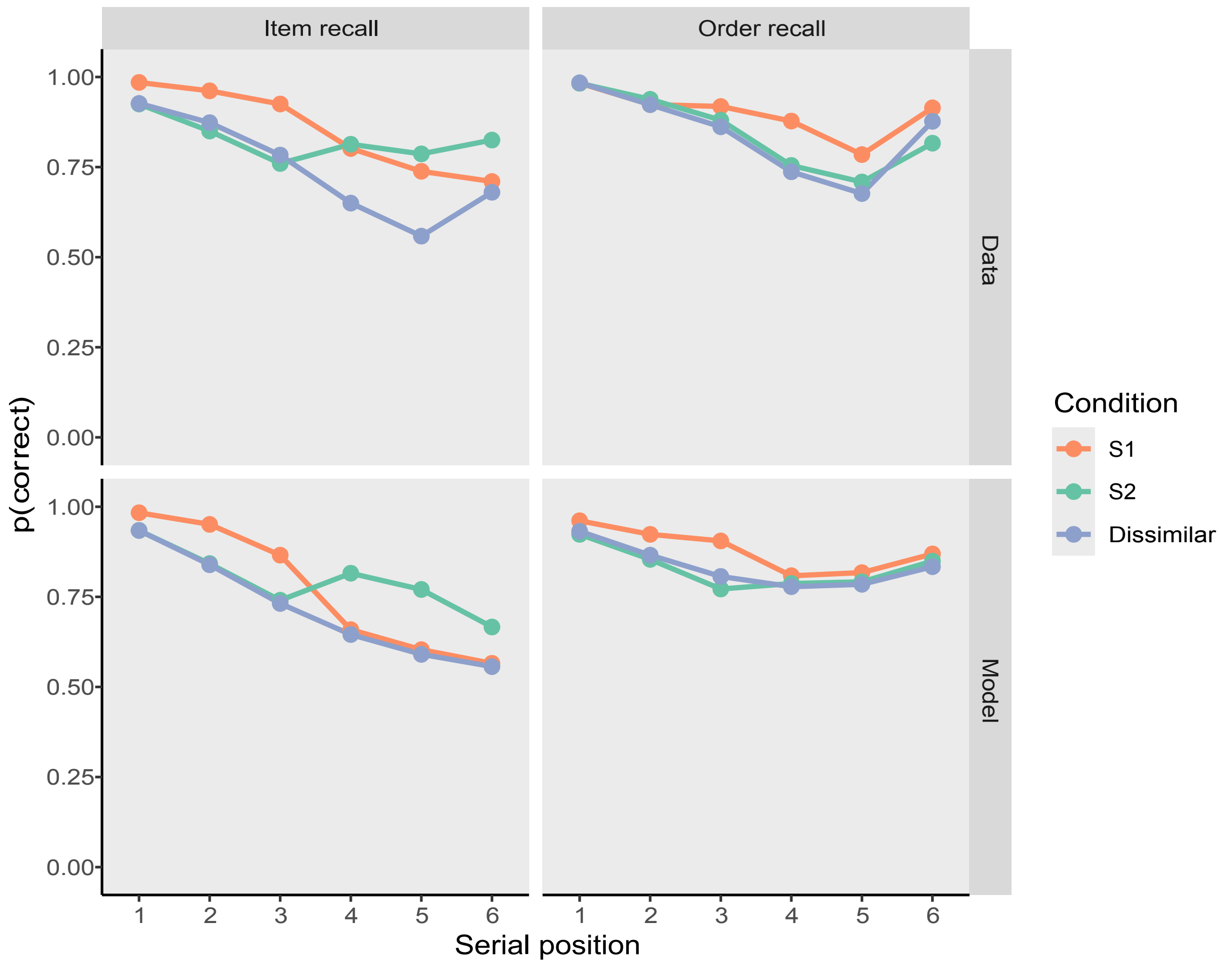
Hence, increasing the activation value of items 1 and 2 increases within-category transpositions, but also decreases the proportion of between-category transpositions. Due to this decrease of between-category transposition, the target item is recalled more often in the correct position, because it migrates less often towards positions 3 and 4. In addition, because of the property described in Eq. 19 through 21, confusion errors do not increase between semantically similar items. If we compute the odds of retrieving item 1 relatively to item 2 in the first scenario, we get: 0.634 / 0.167 = 3.796. In the second scenario, this gives: 0.725 / 0.191 = 3.796. These are the core reasons why the model reproduces the pattern of migration errors observed in the empirical data.

Simulation #8 – Proactive and Retroactive Effects

Semantically similar items, when presented at the beginning of a to-be-remembered list, enhance WM performance for dissimilar items presented later in the list. When the similar items are presented at the end of the list, no such improvement is observed for the dissimilar items earlier in the list (Kowialiewski, Lemaire, & Portrat, 2021). This proactive benefit is observed both at the item and serial order levels. We fitted the model to the data reported by Kowialiewski, Lemaire, & Portrat (2021) using the same procedure as described above. **Figure 28**, lower left panel, shows the recall advantage for similar vs. dissimilar items predicted by the model, as classically observed. We will now provide a more detailed explanation of the results concerning the proactive and retroactive effects.

Figure 28

Proactive and Retroactive Effects



Note. Upper panels: Empirical data. Lower panels: Model predictions. The model reproduces the overall beneficial effect of semantic similarity. However, it falls short at explaining the proactive benefit of semantic similarity (i.e., recall advantage for items 4, 5 and 6 in the "S1" condition). S1 = Semantically similar in the first half of the list. S2 = Semantically similar in the second half of the list. Dissimilar = All items are drawn from a different semantic category. The model was fit to the data reported by Kowialiewski, Lemaire, & Portrat (2021).

The proactive benefit. The model does not predict the proactive benefit on item recall, because it includes no additional WM mechanism to enhance recall performance for the subsequent items. Our implementation increases item recall for semantically similar items, but this influence is specific to the similar items themselves. As there is nothing in the model to make the effect more global, the model is incapable of predicting the proactive benefit.

The absence of retroactive impact. As can be seen in **Figure 28**, lower left panel, the model predicts no retroactive impact on item recall. This occurs for the same reason the model does not predict a proactive benefit: The semantic similarity effect is specific to the semantically similar items themselves. Note however that the model predicts a small, but noticeable retroactive impact in serial position 3 when order recall is considered (see **Figure 28**, right panel). This occurs because when the model tries to retrieve the third item, the semantic features of item 4 will be re-activated to some extent due to the similarity between positions 3 and 4 as retrieval cues. Item 4 will therefore have a slightly higher activation level and therefore a small advantage during the competition for retrieval, resulting in increased anticipation errors.

Simulations: Summary

Results of the simulations for each benchmark are summarized in **Table 2**. As can be seen, the model can explain nearly all the benchmarks presented in the introduction. The only exception is the proactive benefit of semantic similarity.

Table 2. Summary of the Simulations

Benchmarks	Description	Qualitative fit
#1a: Item benefit and omissions	Semantic similarity reduces the production of omission errors	+
#1a: Item benefit and extra-list	Semantic similarity reduces the	+

intrusions (item benefit)	production of extra-list intrusions	
#2: Order recall	Semantic similarity has a null effect on order recall	+
#3: Cue similarity	When positions are tested using items as cue, semantic similarity does not increase confusion errors	+
#4: Task difficulty	Semantic similarity increases with task difficulty	+
#5: Intrusion errors	Semantic similarity modulates the type of intrusion errors	+
#6a: Separation effect on item recall	Similar items presented adjacent to each other are better recalled than when presented at more distant serial positions	+
#6b: Separation effect on order recall	Similar items presented in groups (AAABBB) lead to increased order recall than when presented in an interleaved fashion (ABABAB) or in dissimilar lists (ABCDEF)	+
#7: List structure on transposition errors	The semantic structure of a list constrains the way items are transposed	+
#8a: Proactive benefit	Semantic similarity increases memory performance for subsequent, dissimilar items in the same list	-
#8b: Absence of retroactive impact	Semantic similarity does not retroactively impact memory performance for dissimilar items in the same list	(+)

Note. The different symbols indicate the qualitative fit of the models to the data. +: Predicted correctly; (+): Predicted largely correctly -: Not predicted correctly.

General Discussion

We propose a computational model that offers a comprehensive explanation for the effects of semantic similarity in WM for lists of words. We used a WM architecture in which items are encoded into WM by binding them to their contexts, that is, to their serial positions in the list. Whereas phonological features are bound to their positions instantly and nonselectively, semantic features are encoded in a more limited way: Only those features shared by several memoranda are bound to their contexts. This is made possible by a dynamic threshold of activation for semantic features by which the subset of shared features is selected, together with a tagging mechanism that enables the model to bind shared features to their list positions retrospectively.

Using these principles, the architecture can explain most of the empirical observations made so far on the semantic similarity effect.

Similarity and Confusion Errors

Phonological similarity impairs order recall for short lists of words (Baddeley, 1966; Fallon et al., 2005; Gupta et al., 2005; Nimmo & Roodenrys, 2005; Roodenrys et al., 2022). This effect has been taken as evidence that WM stores information in a phonological format. Whether semantic similarity also impairs order recall has been the object of debates. Early studies did not find a detrimental effect of semantic similarity on order recall (Neale & Tehan, 2007; Poirier & Saint-Aubin, 1995; Saint-Aubin & Poirier, 1999a). Sometimes, a detrimental effect was found (Saint-Aubin et al., 2005; Tse, 2010; Tse et al., 2011), and some have argued that these discrepancies might be explained by the metric used to manipulate semantic similarity (Ishiguro & Saito, 2020). Recent studies have shown multiple times that semantic similarity has no negative impact on order recall (Kowialiewski, Krasnoff, et al., 2023; Kowialiewski, Majerus, et al., 2023), even when different similarity metrics are used (Neath et al., 2022), including the metric recently proposed by Ishiguro and Saito (Ishiguro & Saito, 2024; Kowialiewski et al., 2023). As similarity effects have been previously taken as evidence that WM relies on a particular kind of information, the null effect of semantic similarity on order recall could be taken to imply that WM does not encode semantics.

A recent study has shown that semantic similarity can, under some circumstances, lead to a slight detrimental effect on order recall (Guitard et al., 2025). To test whether the inability of earlier studies to detect this effect was due to a lack of statistical power, we combined results from five experiments using comparable methodologies (i.e., serial recall and order reconstruction, set size 6), resulting in a dataset of 270 participants. The reanalysis of this dataset, reported in **Appendix B**, provides strong evidence against a semantic similarity effect in order reconstruction. In serial recall, the evidence is ambiguous, suggesting that the detrimental effect of semantic similarity on order recall is not a consistent phenomenon. One possible explanation for this small and inconsistent effect is that participants sometimes remember the shared category of the items in semantically similar lists and guess within that category when unable to recall an item. When list items are typical members of that category, they have a high chance of being produced through informed guessing. In this way, failures of item memory can look like failures of order memory. This interpretation can explain why a semantic-similarity effect on order memory has only ever been reported for the order scoring of serial recall, but not for reconstruction of order, as reported in **Appendix B**. Because extra-list intrusions are not possible in order reconstruction, this procedure prevents false classification of guesses as transposition errors.

Does Working Memory Encode Semantics?

To explain the null effect of semantic similarity on order recall, one could assume that WM does not encode semantics by binding semantic features to context. This idea has recently been proposed by Kowialiewski & Majerus (2020) in a model in which the beneficial effect of semantic similarity is explained via a spreading activation mechanism. Basically, the presentation of an item triggers the activation of its concept in a semantic network, and this activation lies outside of the core item-context binding representation – it plays out in what we refer to as the lexical layer L in the present model. Activation then spreads to other semantically similar concepts. For instance, when the concept "tiger" gets activated in the network, it automatically activates neighbor concepts such as “puma” and “cheetah” through spreading activation. When several semantically similar items are presented in the same list, this results in higher activation levels for semantically similar compared to dissimilar concepts, because the similar concepts reinforce each other’s activation in the semantic network. Thanks to this property, such a model can capture the beneficial effect of semantic similarity on item recall (Kowialiewski, Lemaire, & Portrat, 2021; Kowialiewski & Majerus, 2020). In addition, it also predicts a null effect on order recall, because semantic features are not bound to context and are therefore not part of the representation of order, and cannot therefore be used to discriminate between the items.

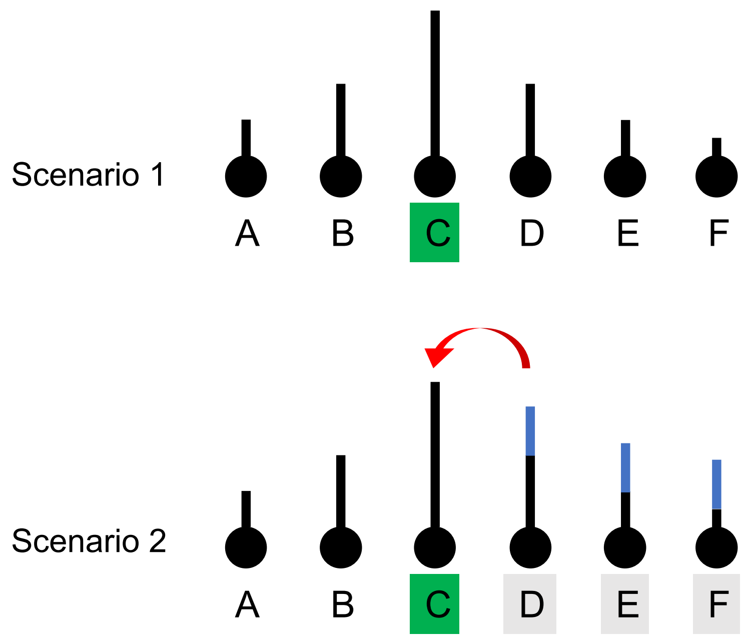
We considered this spreading activation mechanism as an alternative to the model presented in this study. We report additional simulations involving the spreading activation model in **Appendix A**. To summarize the results, such a model assuming that semantic features are not bound to context performs surprisingly well at accounting for several benchmarks. However, simulation results in **Appendix A** reveal two major limitations that we detail below.

First, the spreading activation explanation holds if pure lists of semantically similar and dissimilar items are used. However, this model falls short at accounting for empirical data when experimental conditions involve several items drawn from different semantic categories, for instance in lists such as “**lion** – **leopard** – **cheetah** – *piano* – *flute* – *violin*”. When processing such lists, the spreading activation mechanism fails to account for the fact that items’ semantic categories constrain transposition errors. This is because the spreading activation model does not bind semantic features to context. Without a way to bind semantic information to context, the model has no knowledge regarding which category belongs to which position, and therefore fails at capturing this benchmark.

Second, the spreading activation mechanism fails when similar items are presented at the end of a to-be-remembered list, such as “leopard – bike – table – **Mars – Jupiter – Venus**”. For these lists, the mechanism predicts the production of massive *anticipation errors*, leading to a deleterious retroactive impact on order recall, which conflicts with the empirical data. This result reflects a general problem in models in which items’ relative activation levels affect serial order errors (Kowialiewski, Lemaire, Majerus, et al., 2021). Basically, because semantically similar items in positions 4, 5 and 6 reinforce each other in the semantic network, they have a higher activation level than semantically dissimilar items in positions 1, 2, and 3. When trying to retrieve item 3 (for instance), the activations of items 4, 5 and 6 give them a large advantage in the competition for retrieval over the target item in position 3. Therefore, semantically similar items in later list positions have a higher probability to be recalled at position 3, and in general, to migrate toward earlier serial position, displacing the items in those positions (i.e., anticipations). The problem posed by these families of model is illustrated in **Figure 29**.

Figure 29

Illustration of the Increased Anticipation Errors Problem



Note. This image illustrates the pattern of activation values generated in the spreading activation model when retrieving "item C" in dissimilar lists (Scenario 1) and in lists in which items D, E and F are semantically similar (Scenario 2). The black lines represent the pattern of activation generated by the cueing process. The blue lines represent values from the lexical layer L. In Scenario 1, the pattern of activation generated behaves as usual, with items B and D being equally likely to be retrieved in case of a transposition error. In Scenario 2, the higher activation values of items D, E, and F cause an increase of anticipation errors, because the selection process in our WM models is based on relative activation of recall candidates. This pattern of results is not observed in the data.

Category Encoding and Tagging

When semantically similar items are presented in semantic sub-groups (e.g., "**lion** – **leopard** – **cheetah** – *piano* – *flute* – *violin*"), transposition errors tend to follow the semantic structure imposed by the experimental lists: When items migrate, they are more likely to do so towards the position of another semantically similar than a semantically dissimilar item. This result is difficult to explain without assuming that semantic features are bound in some way to their contexts. Results reported by Kowialiewski, Gorin, et al. (2021) constitute therefore the best evidence we have so far that semantics must be bound in some way to contexts. However, binding semantic features to context in the way same as phonological features necessarily leads to increased confusion errors for lists of semantically similar than dissimilar items. How can this contradiction be resolved? Here we provide a solution based on the idea that WM encodes only the semantic features shared by the items (Kowialiewski, Majerus, et al., 2024). By encoding only the features shared between memoranda, the model not only predicts an item recall advantage for semantically similar items, but also an absence of a detrimental effect on order recall, because the encoded features are common to all items. In that way, the semantic information isn't a useful cue to discriminate between items in lists composed of pure semantically similar items. In lists of dissimilar items, the semantic information is not encoded and is therefore also useless to discriminate between items. However, in mixed lists involving different semantic categories, encoding a semantic category reduces transposition errors from one category to another, which explains the constraint that semantic similarity has on transposition errors.

When designing such a model, we faced an important problem. When encoding a list such as "knife – fork – spoon", we need to assume that their shared semantic features are bound to the positional context of all three words. This is necessary because the semantic similarity effect can already be observed for the first encoded item when input position is deconfounded from output position (see also **Figure 3**). However, when the first word, "knife", is presented, the person does not know yet what, if any, semantic features it will share with subsequent words. That becomes clear only after at least one more item has been presented. Therefore, there must be a mechanism which updates the WM representations at input position N, based on what is encoded at input position N+X. To do this, we first considered a rehearsal mechanism which would go back through the preceding items and would re-encode them. However, we found this explanation implausible, because the semantic similarity effect is found even under concurrent articulatory suppression (Saint-Aubin & Poirier, 1999a). The only explanation we found uses a tagging mechanism in which associations are tagged instead of being directly formed (Rombouts et al., 2015). We combined this with a mechanism that filters semantic features for their relevance in the context of the entire list: In the semantic-ground layer, features are activated to the extent that they are shared by several items presented in close succession, and as such, represent the semantic category or theme that several, or all, list items have in common. Once semantic features have been activated in the semantic-ground layer, and thereby have been identified as being shared by several items during the current trial, the tagged association is transformed into an actual association.

When incorporating semantic representations into the current architecture, we used similarity values from word2vec semantic vectors, which have been shown to account for human performance across various semantic paradigms (Mandera et al., 2017). A potential direction for future research would be to incorporate phonological similarity values based on such principled metrics. A recent study by Zhang and Osth (2024) compared the ability of various orthographic similarity metrics to account for episodic recognition performance within a global matching model and found support for open-bigram representations. Similarly constructed phonological similarity values could prove valuable for two reasons. First, they would enable predicting the identity of extra-list intrusions, as the majority of such errors are phonological (e.g., Romani et al., 2008). Second, incorporating these metrics may help disentangling different ways of representing phonological information, for instance by implementing different theoretical assumptions and comparing their ability to predict transposition errors and extra-list intrusions, thereby advancing our understanding of the underlying structure of WM content.

The Proactive Benefit

Our model fails to capture the proactive benefit of semantic similarity due to the absence of a mechanism that operates globally on the items. There exist two explanations for the proactive benefit. However, both of these explanations have issues which prevent us from including them in our WM architecture.

Decay & Refreshing. One way to explain the proactive benefit is via a compression mechanism coupled with a decay and refreshing architecture, as previously proposed (Kowialiewski, Lemaire, et al., 2024; Kowialiewski, Lemaire, & Portrat, 2021). When encountering a list of semantically similar items, people could extract the common category shared by the similar items and use it to maintain the similar items more easily. Coupled with a decay and refreshing architecture, fewer refreshing attempts are required to maintain the similar items, thus leaving more time to refresh the subsequent items. This leads to a proactive benefit.

Simulations have shown the ability of decay and refreshing models to create a proactive benefit, and no retroactive effect. There are, however, problems posed by this explanation. First, in a list composed of completely similar items, better order recall for the similar items themselves is expected, because items will be more strongly encoded via item-context binding, thus violating benchmark #1 (see also **Figure 30** and associated explanation in the next paragraphs).

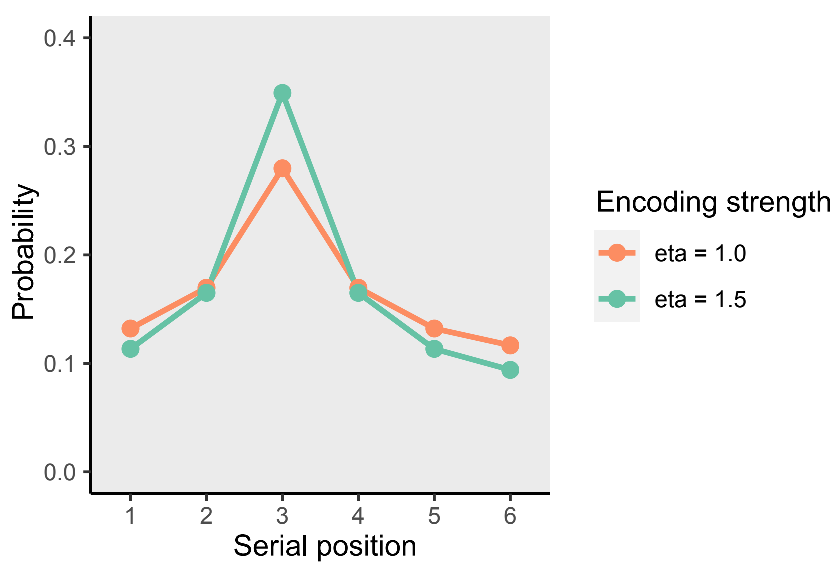
Second, there is compelling evidence against decay of verbal information in WM (Farrell et al., 2016; Lewandowsky et al., 2009). In addition, refreshing itself currently lacks direct empirical support (Oberauer & Souza, 2020), and studies have failed to find evidence that increasing refreshing rate increases memory performance (Souza & Oberauer, 2018). Therefore, attributing the proactive benefit to decay and refreshing mechanisms remains risky without robust empirical evidence supporting them. This issue reflects a more general problem: As long as the fundamental properties of WM are not established, making interpretation regarding potential interactions between WM and other cognitive functions becomes risky, because we are dealing with too many unknown mechanisms.

Encoding Resource. Another way to explain the proactive benefit is by including an encoding-resource mechanism (Popov & Reder, 2020). In this mechanism, encoding an item depletes a proportion of a limited resource. Encoding strength, in turn, is proportional to the amount of resource available. One could assume that similar items deplete less of that resource, because they are easier to activate, for instance via spreading of activation in the semantic network (Kowialiewski, Lemaire, et al., 2022; Kowialiewski, Lemaire, & Portrat, 2021). This mechanism predicts a proactive benefit, because when encoding “leopard – lion – puma”, these items should deplete a smaller part of the encoding resource than three dissimilar items. The saved resource can subsequently be used to encode more strongly the following items. However, including this mechanism creates additional problems. First, for all-similar compared to all-dissimilar lists, this mechanism predicts increased semantic similarity benefits across input position. This occurs because the savings in encoding resource should accumulate with each additional encoded item in the similar compared to the dissimilar condition. A growing similarity benefit is not observed when input position is deconfounded with output position, as shown in

Figure 3. Second, the encoding-resource mechanism predicts better order recall performance for lists of purely similar vs. dissimilar items: Stronger encoding into WM means increasing the item-context binding. The phenomenon is illustrated in **Figure 30**, which shows the probability to retrieve each item given a retrieval cue (i.e., in this case, positional marker 3), before and after multiplying encoding strength by 1.5. As can be seen, increasing item-context binding translates to increased probability to select the target item, and reduced probability to select other list-items, leading to increased order recall. This prediction is in contradiction with the absence of semantic similarity effect on order recall performance.

Figure 30

Probability of Retrieving a Target Item Compared to a Non-Target Item as a Function of Binding Strength



Note. In this hypothetical scenario, the model tries to retrieve “item 3”. The third item therefore has the strongest probability to be retrieved. Because of positional overlap, other list items also have a non-zero probability to be retrieved. As can be seen, increasing encoding strength from

1.0 to 1.5 increases the probability to retrieve the target item, and decreases the probability to retrieve other list-items. Stronger item-context binding therefore leads to improved order recall.

The reason why semantic similarity proactively impacts WM performance remains to be explained. Without further empirical explorations regarding the boundary conditions of this phenomenon, it remains yet challenging to uphold a robust theoretical explanation. It must be noted that proactive benefits have been observed across a wide range of experimental manipulations, such as word frequency (Miller & Roodenrys, 2012), chunking (Thalmann et al., 2019), Hebb learning (Mizrak & Oberauer, 2021b), and temporal gaps (Mizrak & Oberauer, 2021a). Therefore, explaining why memory performance in one part of the list improves when the preceding part is easier to process is likely to be a general problem for models of memory, and goes beyond the assumptions we include to explain our semantic similarity effects.

Alternative Architectures

The principles we exposed in this work are not restricted to any particular kind of architecture. We chose to integrate semantic representations in an architecture using direct bindings between item and positional vectors, as in previous models (Lewandowsky & Farrell, 2008; Oberauer et al., 2012; Oberauer & Lin, 2024). There is no reason to believe that other architectures wouldn't be able to achieve the same results. An example is the feature model (Nairne, 1990) and its recently revised version (Saint-Aubin et al., 2021). One issue with the feature model relates to its feature-overwriting mechanism, according to which features of item N overwrite the features of previously encoded items with a certain probability if they are shared with item N. When an item has its features overwritten, it is *weakened.* This implies that this mechanism predicts a detrimental (as opposed to beneficial) effect of semantic similarity on item recall, which contrasts with the empirical data. Further, as any other model of serial recall, it predicts a detrimental effect of semantic similarity on order recall. We assume that the feature model could account for the benchmarks reported in the present literature review by revising the way it represents semantics, just as we did in the present architecture. Such modifications could include dropping the feature-overwriting mechanism for the semantic features, and implementing a mechanism preventing confusion errors between semantically similar items, such as the tagging mechanism proposed in the current manuscript.

Conclusion

Semantic similarity behaves in a qualitatively different way than other similarity effects, such as phonological similarity, notably by not increasing confusion errors. This may suggest that semantic information plays no role in the WM representation. In contradiction to this, experiments using lists composed of multiple semantic subgroups revealed that people remember information about the position of categorical information. We resolved this apparent contradiction by proposing a mechanism wherein only semantic features shared by several items in a list are encoded by binding them to positional contexts. This process is made possible via two core mechanisms: A tagging mechanism which allows semantic features to be bound retroactively based on their relevance for a particular trial, and a threshold mechanism which filters activation of semantic features based on their frequency of appearance. These combined mechanisms can explain most of the semantic similarity effects which have been observed so far in the literature.

Appendix A – The Spreading Activation Mechanism

This section presents a spreading activation mechanism as an alternative to the category-encoding assumption used in the main text. In the spreading activation mechanism, the presentation of an item triggers the activation of its lexical unit, which spread activation towards the semantic features it connects to. Contrary to the category-encoding assumption, these semantic features are not encoded into WM by binding them to contexts. Instead, the semantic features spread activation back towards items' lexical units. Due to this mechanism, semantically similar items reinforce each other via their shared features. For instance, when a list such as "leopard – lion" is presented, the features of the word "leopard" become activated. When activation spreads back to the lexical units, this will not only re-activate the concept "leopard", but also the concept "lion", because both concepts share features. Similarly, when encoding "lion", this concept will activate "leopard" through their shared features. This results in higher activation level in the items' localist units as compared to a situation where items are semantically dissimilar. These additional activations are then used to help the items surpass the omission threshold, resulting in higher item recall.

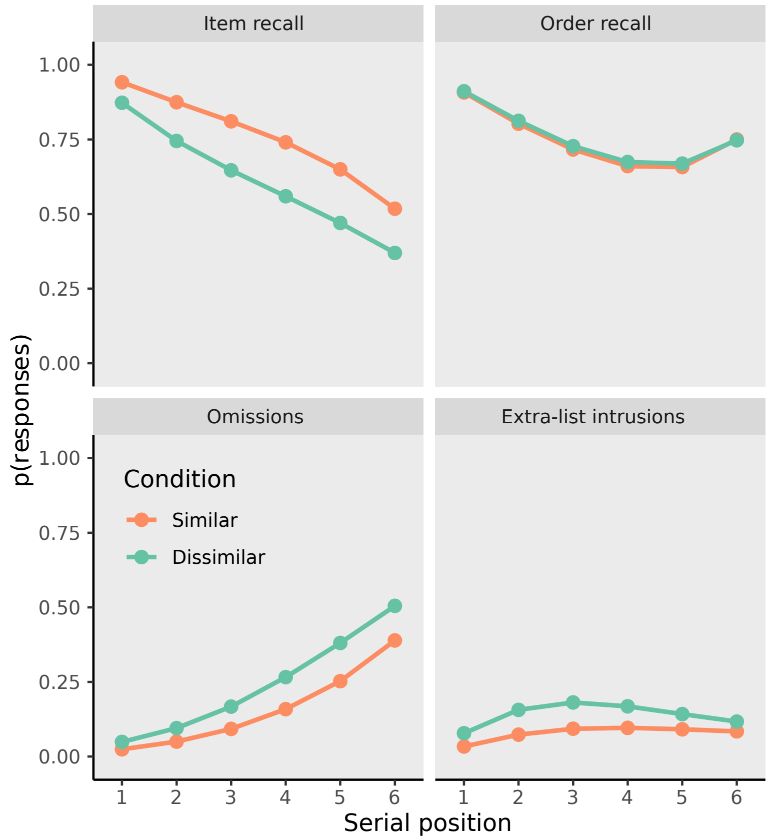
More formally, the dynamics of the activation for the semantic features is identical to those described in Eq. 11 through 14. We kept the threshold mechanism, because it ultimately allows the model to simulate the separation effect. Contrary to the category-encoding assumption, the activated semantic features are not directly used to create item-context associations. Instead, the semantic features send their activation back to the lexical units they connect to at the end of encoding:

$$\Delta L_i = \mu_{Fj} \qquad \textit{Eq. 24}$$

Figure A1 shows simulation results for immediate serial recall. The model predicts an item recall advantage for semantically similar vs. dissimilar items. As expected, the higher activation provided by the spreading activation mechanism allows the list-items to surpass the omission threshold more often in the similar than in the dissimilar condition. In addition, this boost of activation makes list-items recalled more often than non-list items in the similar than in the dissimilar condition, reducing the production of extra-list intrusions. Finally, the model does not predict any effect of semantic similarity on order recall, because it does not encode semantic features by binding them to contexts. Therefore, the model performs very well on these benchmarks.

Figure A1

Semantic Similarity in Immediate Serial Recall



Next, **Figure A2**, left panel, shows that the spreading activation mechanism can predict the separation effect after fitting the model on the data reported by Kowialiewski, Majerus, and colleagues (2023). The fact that this mechanism produces a separation effect has nothing to do with the specificity of the spreading activation mechanism *per se*, but is due to the threshold mechanism we implement throughout all simulations. More importantly, the model does not predict better order recall performance for grouped vs. dissimilar lists, as can be seen in the right panel. This misprediction is an important one, because it shows one of the main limitations of the model that we explain in the next paragraphs.

Figure A2

Simulations of the Separation Effect

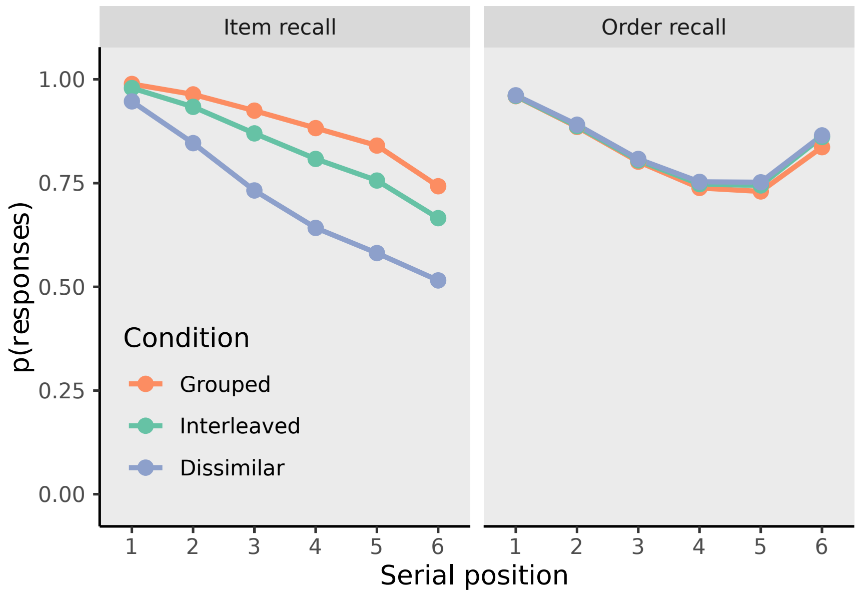
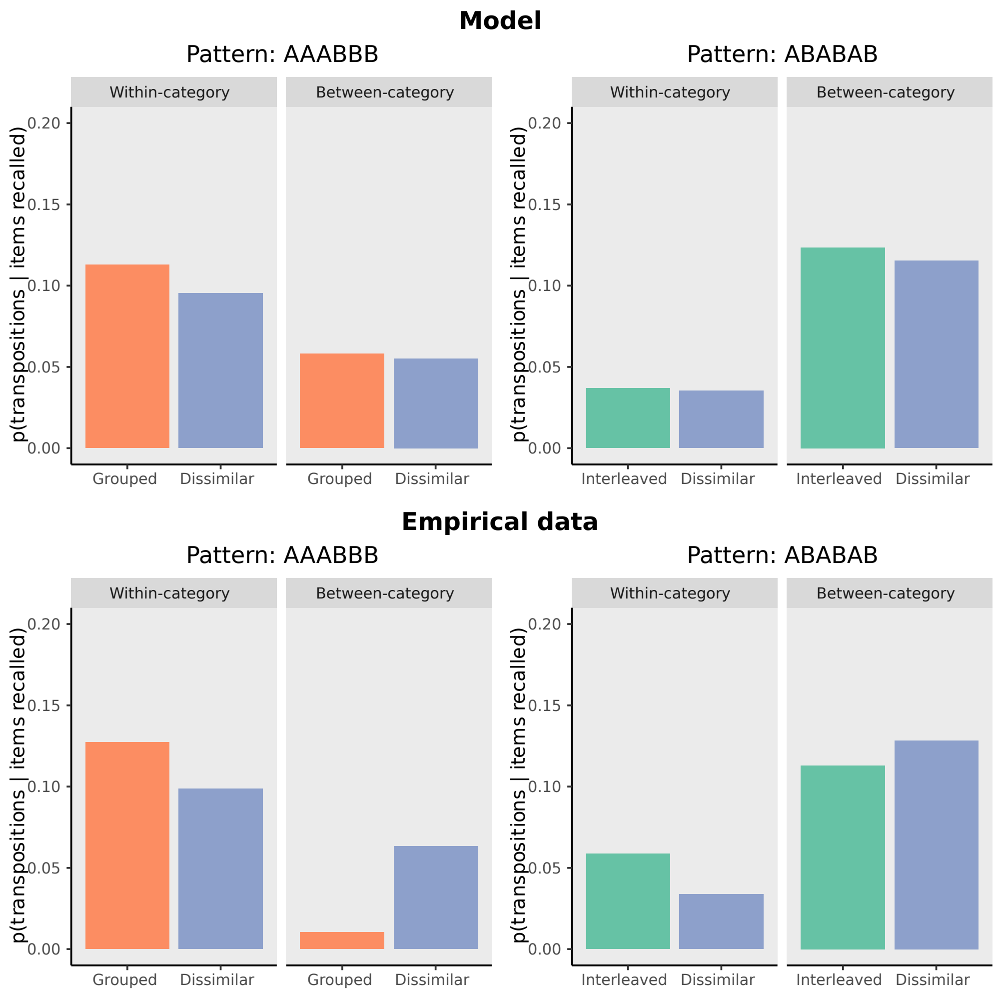


Figure A3 shows the same simulations results as in **Figure A2**, except that we display this time the pattern of transposition errors. The spreading activation model does not predict an effect of list structure on transposition errors. The reason why this happens is trivial: Since the model does not bind semantic features to context, it has no information regarding which item belonged where based on that item's semantic content. This also explains why the model does not predict better order recall performance in the grouped vs. dissimilar condition of **Figure A2**.

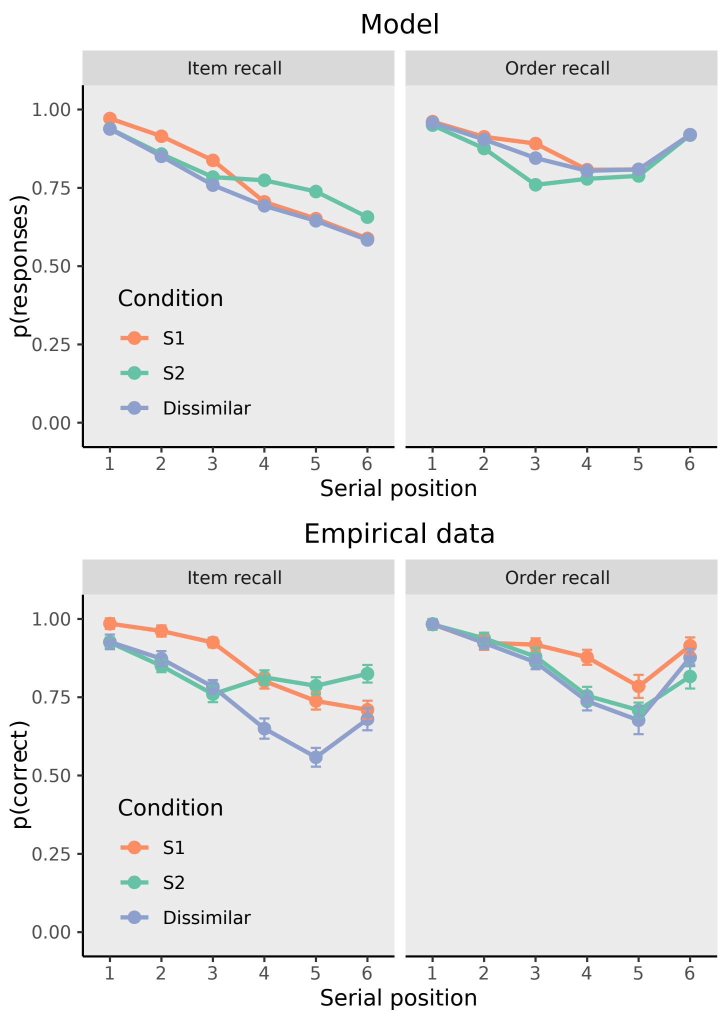
Figure A3



Last, **Figure A4** shows the model's predictions on proactive and retroactive benefits. As can be seen, the spreading activation mechanism shows a substantial deleterious retroactive impact on order recall, contrary to experimental data. This problem is explained in further details in the General Discussion sections. Briefly, when items 4, 5 and 6 are semantically similar, they benefit from a boost of activation in the output layer $\boldsymbol{O}$. Due to this boost of activation, these items will always have a competitive advantage for retrieval, regardless of the retrieval cue currently used. This means that they will often outcompete the correct items in earlier serial positions, leading to increased anticipation errors and a detrimental retroactive effect. This problem also explains why the similarity benefit is so small in the spreading activation model: During model fitting, small parameter values for similarity are favored to prevent a high number of anticipation errors, which would lead to a strong deviation from the data. This is another important misprediction from the model.

Figure A4

Proactive and Retroactive Effects



Appendix B – Reanalyzing semantic similarity data

In this section, we report a re-analysis of the following datasets:

- Kowialiewski et al. (2023), Experiment 1
- Kowialiewski et al. (2024), Experiment 3
- Neath et al. (2023), Experiments 1, 2 and 3

These datasets were chosen because they all used comparable experimental procedures. Specifically, all these studies involved a paradigm requiring participants to encode and serially recall lists of 6 items, tested in two ways: Serial recall and order reconstruction. Together, these data sets form a sample of 270 participants. Because it is possible that the two procedures for assessing memory for order lead to different outcomes, the figure below shows the data split as a function of test procedure: order reconstruction and serial recall (conditional order score) in the left and right panels, respectively.

As can be seen, the order reconstruction tasks provide strong evidence for the absence of an effect of semantic similarity (M_{diff} = 0.17%, d = 0.002), as supported by the Bayes factor in favor of the null hypothesis (BF_{01} = 14.66). In contrast, the serial recall data show a small difference (M_{diff} = 1.39%, d = 0.136), which is not credibly supported (BF_{01} = 1.95).

References

Baddeley, A. D. (1966). Short-term Memory for Word Sequences as a Function of Acoustic, Semantic and Formal Similarity. *Quarterly Journal of Experimental Psychology*, *18*(4), 362–365. https://doi.org/10.1080/14640746608400055

Binder, J. R., Conant, L. L., Humphries, C. J., Fernandino, L., Simons, S. B., Aguilar, M., & Desai, R. H. (2016). Toward a brain-based componential semantic representation. *Cognitive Neuropsychology*, *33*(3–4), 130–174. https://doi.org/10.1080/02643294.2016.1147426

Binder, J. R., & Desai, R. H. (2011). The neurobiology of semantic memory. *Trends in Cognitive Sciences*, *15*(11), 527–536. https://doi.org/10.1016/j.tics.2011.10.001

Binder, J. R., Desai, R. H., Graves, W. W., & Conant, L. L. (2009). Where Is the Semantic System? A Critical Review and Meta-Analysis of 120 Functional Neuroimaging Studies. *Cerebral Cortex*, *19*(12), 2767–2796. https://doi.org/10.1093/cercor/bhp055

Brooks, J. O., & Watkins, M. J. (1990). Further Evidence of the Intricacy of Memory Span. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*(6), 1134–1141. https://doi.org/10.1037/0278-7393.16.6.1134

Bunting, M., Cowan, N., & Scott Saults, J. (2006). How does running memory span work? *Quarterly Journal of Experimental Psychology*, *59*(10), 1691–1700. https://doi.org/10.1080/17470210600848402

Burgess, N., & Hitch, G. J. (1999). Memory for Serial Order: A Network Model of the Phonological Loop and Its Timing. *Psychological Review*, *106*(3), 551–581.

Burgess, N., & Hitch, G. J. (2006). A revised model of short-term memory and long-term learning of verbal sequences. *Journal of Memory and Language*, *55*(4), 627–652. https://doi.org/10.1016/j.jml.2006.08.005

Cowan, N. (1999). An Embedded-Processes Model of working memory. In *Models of working memory: Mechanisms of active maintenance and executive control* (Cambridge University Press, pp. 62–101).

Cowan, N., Day, L., Saults, J. S., Keller, J., Johnson, T., & Flores, L. (1992). The role of verbal output time in the effects of word length on immediate memory. *Journal of Memory and Language*, *31*(1), 1–17. https://doi.org/10.1016/0749-596X(92)90002-F

Cowan, N., Saults, J. S., Elliott, E. M., & Moreno, M. V. (2002). Deconfounding Serial Recall. *Journal of Memory and Language*, *46*(1), 153–177. https://doi.org/10.1006/jmla.2001.2805

Craik, F. I., & Birtwistle, J. (1971). Proactive inhibition in free recall. *Journal of Experimental Psychology*, *91*(1), 120.

Deese. (1959). On the prediction of occurrence of particular verbal intrusions in immediate recall. *Journal of Experimental Psychology*, *58*(1), 17.

Devereux, B. J., Tyler, L. K., Geertzen, J., & Randall, B. (2014). The Centre for Speech, Language and the Brain (CSLB) concept property norms. *Behavior Research Methods*, *46*(4), 1119–1127. https://doi.org/10.3758/s13428-013-0420-4

Fallon, A. B., Mak, E., Tehan, G., & Daly, Charmaine. (2005). Lexicality and phonological similarity: A challenge for the retrieval-based account of serial recall? *Memory*, *13*(3–4), 349–356. https://doi.org/10.1080/09658210344000215

Farrell, S., Oberauer, K., Greaves, M., Pasiecznik, K., Lewandowsky, S., & Jarrold, C. (2016). A test of interference versus decay in working memory: Varying distraction within lists in a complex span task. *Journal of Memory and Language*, *90*, 66–87. https://doi.org/10.1016/j.jml.2016.03.010

Goh, W. D., & Goh, C. K. (2006). The roles of semantic similarity and proactive interference in the word length effect. *Psychonomic Bulletin & Review*, *13*(6), 978–984. https://doi.org/10.3758/BF03213912

Guérard, K., & Saint-Aubin, J. (2012). Assessing the effect of lexical variables in backward recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*(2), 312–324. https://doi.org/10.1037/a0025481

Guérard, K., Tremblay, S., & Saint-Aubin, J. (2009). Short Article: Similarity and Binding in Memory: Bound to be Detrimental. *Quarterly Journal of Experimental Psychology*, *62*(1), 26–32. https://doi.org/10.1080/17470210802215277

Guitard, D., Saint-Aubin, J., Nick Reid, J., & Jamieson, R. K. (2025). An embedded computational framework of memory: Accounting for the influence of semantic information in verbal short-term memory. *Journal of Memory and Language*, *140*, 104573. https://doi.org/10.1016/j.jml.2024.104573

Gupta, P., Lipinski, J., & Aktunc, E. (2005). Reexamining the phonological similarity effect in immediate serial recall: The roles of type of similarity, category cuing, and item recall. *Memory & Cognition*, *33*(6), 1001–1016. https://doi.org/10.3758/BF03193208

Henson, R. N. A. (1998). Short-Term Memory for Serial Order: The Start-End Model. *Cognitive Psychology*, *36*(2), 73–137. https://doi.org/10.1006/cogp.1998.0685

Hockey, R. (1973). Rate of Presentation in Running Memory and Direct Manipulation of Input-Processing Strategies. *Quarterly Journal of Experimental Psychology*, *25*(1), 104–111. https://doi.org/10.1080/14640747308400328

Hockey, R., & Hamilton, P. (1977). The Basis of the Primacy Effect: Some Experiments with Running Memory. *Quarterly Journal of Experimental Psychology*, *29*(1), 49–63. https://doi.org/10.1080/00335557743000026

Hulme, C., Maughan, S., & Brown, G. D. A. (1991). Memory for familiar and unfamiliar words: Evidence for a long-term memory contribution to short-term memory span. *Journal of Memory and Language*, *30*(6), 685–701. https://doi.org/10.1016/0749-596X(91)90032-F

Ishiguro, S., & Saito, S. (2020). The detrimental effect of semantic similarity in short-term memory tasks: A meta-regression approach. *Psychonomic Bulletin & Review*. https://doi.org/10.3758/s13423-020-01815-7

Ishiguro, S., & Saito, S. (2024). The Semantic Similarity Effect on Short-Term Memory: Null Effects of Affectively Defined Semantic Similarity. *Journal of Cognition*, *7*(1), 24. https://doi.org/10.5334/joc.349

Kowialiewski, B., Gorin, S., & Majerus, S. (2021). Semantic knowledge constrains the processing of serial order information in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. https://doi.org/10.1037/xlm0001031

Kowialiewski, B., Krasnoff, J., Mizrak, E., & Oberauer, K. (2022). The Semantic Relatedness Effect in Serial Recall: Deconfounding Encoding and Recall Order. *Journal of Memory and Language*, *127*. https://doi.org/10.1016/j.jml.2022.104377

Kowialiewski, B., Krasnoff, J., Mizrak, E., & Oberauer, K. (2023). Verbal working memory encodes phonological and semantic information differently. *Cognition*, *233*(105364). https://doi.org/10.1016/j.cognition.2022.105364

Kowialiewski, B., Lemaire, B., Majerus, S., & Portrat, S. (2021). Can activated long-term memory maintain serial order information? *Psychonomic Bulletin & Review*, *28*, 1301–1312. https://doi.org/10.3758/s13423-021-01902-3

Kowialiewski, B., Lemaire, B., & Portrat, S. (2021). How does semantic knowledge impact working memory maintenance? Computational and behavioral investigations. *Journal of Memory and Language*, *117*(104208). https://doi.org/10.1016/j.jml.2020.104208

Kowialiewski, B., Lemaire, B., & Portrat, S. (2022). Between-item Similarity Frees Up Working Memory Resources Through Compression: A Domain-General Property. *Journal of Experimental Psychology: General*, *151*(11), 2641–2665.

Kowialiewski, B., Lemaire, B., & Portrat, S. (2024). Similarity-Based Compression in Working Memory: Implications for Decay and Refreshing Models. *Computational Brain & Behavior*, *7*, 163–180. https://doi.org/10.1007/s42113-023-00179-0

Kowialiewski, B., & Majerus, S. (2018). The non-strategic nature of linguistic long-term memory effects in verbal short-term memory. *Journal of Memory and Language*, *101*, 64–83. https://doi.org/10.1016/j.jml.2018.03.005

Kowialiewski, B., & Majerus, S. (2020). The varying nature of semantic effects in working memory. *Cognition*, *202*, 104278. https://doi.org/10.1016/j.cognition.2020.104278

Kowialiewski, B., Majerus, S., & Oberauer, K. (2024). Does semantic similarity affect immediate memory for order? Usually not, but sometimes it does. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *50*(1), 68–88. https://doi.org/10.1037/xlm0001279

Lambon-Ralph, M., Jefferies, E., Patterson, K., & Rogers, T. T. (2017). The neural and computational bases of semantic cognition. *Nature Reviews Neuroscience*, *18*(1), 42–55. https://doi.org/10.1038/nrn.2016.150

Landauer, T. K., & Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, *104*(2), 211.

Lewandowsky, S., & Farrell, S. (2008). Short-Term Memory: New Data and a Model. In *Psychology of Learning and Motivation* (Vol. 49, pp. 1–48). Elsevier. https://doi.org/10.1016/S0079-7421(08)00001-7

Lewandowsky, S., Oberauer, K., & Brown, G. D. A. (2009). No temporal decay in verbal short-term memory. *Trends in Cognitive Sciences*, *13*(3), 120–126. https://doi.org/10.1016/j.tics.2008.12.003

Lian, A., & Karlsen, P. J. (2004). Advantages and disadvantages of phonological similarity in serial recall and serial recognition of nonwords. *Memory & Cognition*, *32*(2), 223–234. https://doi.org/10.3758/BF03196854

Mandera, P., Keuleers, E., & Brysbaert, M. (2017). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, *92*, 57–78. https://doi.org/10.1016/j.jml.2016.04.001

Miller, L. M., & Roodenrys, S. (2012). Serial recall, word frequency, and mixed lists: The influence of item arrangement. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *38*(6), 1731–1740. https://doi.org/10.1037/a0028470

Mizrak, E., & Oberauer, K. (2021a). What Is Time Good for in Working Memory? *Psychological Science*, *32*(8), 1325–1337.

Mizrak, E., & Oberauer, K. (2021b). Working memory recruits long-term memory when it is beneficial: Evidence from the Hebb effect. *Journal of Experimental Psychology: General*. https://doi.org/10.1037/xge0000934

Monnier, C., & Bonthoux, F. (2011). The semantic-similarity effect in children: Influence of long-term knowledge on verbal short-term memory: Semantic contribution to serial recall in children. *British Journal of Developmental Psychology*, *29*(4), 929–941. https://doi.org/10.1111/j.2044-835X.2010.02024.x

Mueller, C. W., & Watkins, M. J. (1977). Inhibition from part-set cuing: A cue-overload interpretation. *Journal of Verbal Learning and Verbal Behavior*, *16*(6), 699–709. https://doi.org/10.1016/S0022-5371(77)80029-7

Nairne, J. S. (1990). A feature model of immediate memory. *Memory & Cognition*, *18*(3), 251–269. https://doi.org/10.3758/BF03213879

Nairne, J. S., & Kelley, M. R. (2004). Separating item and order information through process dissociation. *Journal of Memory and Language*, *50*(2), 113–133. https://doi.org/10.1016/j.jml.2003.09.005

Neale, K., & Tehan, G. (2007). Age and redintegration in immediate memory and their relationship to task difficulty. *Memory & Cognition*, *35*(8), 1940–1953. https://doi.org/10.3758/BF03192927

Neath, I., Saint-Aubin, J., & Surprenant, A. M. (2022). Semantic Relatedness Effects in Serial Recall But Not in Serial Reconstruction of Order. *Experimental Psychology*, 1618-3169/a000557. https://doi.org/10.1027/1618-3169/a000557

Neath, I., & Surprenant, A. M. (2019). Set size and long-term memory/lexical effects in immediate serial recall: Testing the impurity principle. *Memory & Cognition*, *47*(3), 455–472. https://doi.org/10.3758/s13421-018-0883-8

Nee, D. E., & Jonides, J. (2008). Neural correlates of access to short-term memory. *Proceedings of the National Academy of Sciences*, *105*(37), 14228–14233. https://doi.org/10.1073/pnas.0802081105

Nimmo, L. M., & Roodenrys, S. (2004). Investigating the phonological similarity effect: Syllable structure and the position of common phonemes. *Journal of Memory and Language*, *50*(3), 245–258. https://doi.org/10.1016/j.jml.2003.11.001

Nimmo, L. M., & Roodenrys, S. (2005). The phonological similarity effect in serial recognition. *Memory*, *13*(7), 773–784. https://doi.org/10.1080/09658210444000386

Oberauer, K. (2002). Access to information in working memory: Exploring the focus of attention. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *28*(3), 411–421. https://doi.org/10.1037/0278-7393.28.3.411

Oberauer, K. (2003). Understanding serial position curves in short-term recognition and recall. *Journal of Memory and Language*, *49*(4), 469–483. https://doi.org/10.1016/S0749-596X(03)00080-9

Oberauer, K. (2009). Design for a Working Memory. In *Psychology of Learning and Motivation* (Vol. 51, pp. 45–100). Elsevier. https://doi.org/10.1016/S0079-7421(09)51002-X

Oberauer, K., & Lewandowsky, S. (2011). Modeling working memory: A computational implementation of the Time-Based Resource-Sharing theory. *Psychonomic Bulletin & Review*, *18*(1), 10–45. https://doi.org/10.3758/s13423-010-0020-6

Oberauer, K., Lewandowsky, S., Awh, E., Brown, G. D. A., Conway, A., Cowan, N., Donkin, C., Farrell, S., Hitch, G. J., Hurlstone, M. J., Ma, W. J., Morey, C. C., Nee, D. E., Schweppe, J., Vergauwe, E., & Ward, G. (2018). Benchmarks for models of short-term and working memory. *Psychological Bulletin*, *144*(9), 885–958. https://doi.org/10.1037/bul0000153

Oberauer, K., Lewandowsky, S., Farrell, S., Jarrold, C., & Greaves, M. (2012). Modeling working memory: An interference model of complex span. *Psychonomic Bulletin & Review*, *19*(5), 779–819. https://doi.org/10.3758/s13423-012-0272-4

Oberauer, K., & Lin, H.-Y. (2024). An interference model for visual and verbal working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *in press*.

Oberauer, K., & Souza, A. S. (2020). How fast can people refresh and rehearse information in working memory? *Memory & Cognition*. https://doi.org/10.3758/s13421-020-01062-0

Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, *105*(4), 761.

Patterson, K., Graham, N., & Hodges, J. R. (1994). The Impact of Semantic Memory Loss on Phonological Representations. *Journal of Cognitive Neuroscience*, *6*(1), 57–69. https://doi.org/10.1162/jocn.1994.6.1.57

Poirier, M., & Saint-Aubin, J. (1995). Memory for Related and Unrelated Words: Further Evidence on the Influence of Semantic Factors in Immediate Serial Recall. *The Quarterly Journal of Experimental Psychology Section A*, *48*(2), 384–404. https://doi.org/10.1080/14640749508401396

Poirier, M., Saint-Aubin, J., Mair, A., Tehan, G., & Tolan, A. (2015). Order recall in verbal short-term memory: The role of semantic networks. *Memory & Cognition*, *43*(3), 489–499. https://doi.org/10.3758/s13421-014-0470-6

Pollack, I., Johnson, L. B., & Knaff, P. R. (1959). Running memory span. *Journal of Experimental Psychology*, *57*(3), 137–146. https://doi.org/10.1037/h0046137

Popov, V., & Reder, L. M. (2020). Frequency effects on memory: A resource-limited theory. *Psychological Review*, *127*(1), 1–46. https://doi.org/10.1037/rev0000161

Roediger, & McDermott, K. B. (1995). Creating False Memories: Remembering Words Not Presented in Lists. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(4), 803.

Romani, C., Mcalpine, S., & Martin, R. C. (2008). Concreteness Effects in Different Tasks: Implications for Models of Short-Term Memory. *Quarterly Journal of Experimental Psychology*, *61*(2), 292–323. https://doi.org/10.1080/17470210601147747

Rombouts, J. O., Bohte, S. M., & Roelfsema, P. R. (2015). How Attention Can Create Synaptic Tags for the Learning of Working Memories in Sequential Tasks. *PLOS Computational Biology*, *11*(3), e1004060. https://doi.org/10.1371/journal.pcbi.1004060

Roodenrys, S., Guitard, D., Miller, L. M., Saint-Aubin, J., & Barron, J. M. (2022). Phonological similarity in the serial recall task hinders item recall, not just order. *British Journal of Psychology*, bjop.12575. https://doi.org/10.1111/bjop.12575

Rosselet-Jordan, F. L., Abadie, M., Mariz-Elsig, S., & Camos, V. (2022). Role of attention in the associative relatedness effect in verbal working memory: Behavioral and chronometric perspectives. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. https://doi.org/10.1037/xlm0001102

Saint-Aubin, J., Guérard, K., Chamberland, C., & Malenfant, A. (2014). Delineating the contribution of long-term associations to immediate recall. *Memory*, *22*(4), 360–373. https://doi.org/10.1080/09658211.2013.794242

Saint-Aubin, J., Ouellette, D., & Poirier, M. (2005). Semantic similarity and immediate serial recall: Is there an effect on all trials. *Psychonomic Bulletin & Review*, *12*, 171–177.

Saint-Aubin, J., & Poirier, M. (1999a). Semantic Similarity and Immediate Serial Recall: Is There a Detrimental Effect on Order Information? *The Quarterly Journal of Experimental Psychology Section A*, *52*(2), 367–394. https://doi.org/10.1080/713755814

Saint-Aubin, J., & Poirier, M. (1999b). The Influence of Long-term Memory Factors on Immediate Serial Recall: An Item and Order Analysis. *International Journal of Psychology*, *34*(5–6), 347–352.

Saint-Aubin, J., Yearsley, J. M., Poirier, M., Cyr, V., & Guitard, D. (2021). A model of the production effect over the short-term: The cost of relative distinctiveness. *Journal of Memory and Language*, *118*, 104219. https://doi.org/10.1016/j.jml.2021.104219

Saito, S., Logie, R. H., Morita, A., & Law, A. (2008). Visual and phonological similarity effects in verbal immediate serial recall: A test with kanji materials. *Journal of Memory and Language*, *59*(1), 1–17. https://doi.org/10.1016/j.jml.2008.01.004

Schneegans, S., & Bays, P. M. (2017). Neural Architecture for Feature Binding in Visual Working Memory. *The Journal of Neuroscience*, *37*(14), 3913–3925. https://doi.org/10.1523/JNEUROSCI.3493-16.2017

Schweickert, R. (1993). A multinomial processing tree model for degradation and redintegration in immediate recall. *Memory & Cognition*, *21*(2), 168–175. https://doi.org/10.3758/BF03202729

Smith, F. (2012). *Understanding reading: A psycholinguistic analysis of reading and learning to read*. Routledge.

Souza, A. S., & Oberauer, K. (2018). Does Articulatory Rehearsal Help Immediate Serial Recall? *Cognitive Psychology,* 55.

Stadler, M. A., Roediger, H. L., & McDermott, K. B. (1999). Norms for word lists that create false memories. *Memory & Cognition*, *27*(3), 494–500. https://doi.org/10.3758/BF03211543

Tehan, G. (2010). Associative relatedness enhances recall and produces false memories in immediate serial recall. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, *64*(4), 266–272. https://doi.org/10.1037/a0021375

Thalmann, M., Souza, A. S., & Oberauer, K. (2019). How does chunking help working memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *45*(1), 37–55. https://doi.org/10.1037/xlm0000578

Tse, C.-S. (2009). The role of associative strength in the semantic relatedness effect on immediate serial recall. *Memory*, *17*(8), 874–891. https://doi.org/10.1080/09658210903376250

Tse, C.-S. (2010). A negative semantic similarity effect on short-term order memory: Evidence from recency judgements. *Memory*, *18*(6), 638–656. https://doi.org/10.1080/09658211.2010.499875

Tse, C.-S., Li, Y., & Altarriba, J. (2011). The effect of semantic relatedness on immediate serial recall and serial recognition. *Quarterly Journal of Experimental Psychology*, *64*(12), 2425–2437. https://doi.org/10.1080/17470218.2011.604787

Walker, I., & Hulme, C. (1999). Concrete Words Are Easier to Recall Than Abstract Words: Evidence for a Semantic Contribution to Short-Term Serial Recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(5), 1256–1271.

Watkins, M. J., & Watkins, O. C. (1976). Cue-overload theory and the method of interpolated attributes. *Bulletin of the Psychonomic Society*, *7*(3), 289–291. https://doi.org/10.3758/BF03337192

Wickens, D. D. (1970). Encoding categories of words: An empirical approach to meaning. *Psychological Review*, *77*(1), 1–15. https://doi.org/10.1037/h0028569

Xu, Y., Wang, X., Wang, X., Men, W., Gao, J.-H., & Bi, Y. (2018). Doctor, Teacher, and Stethoscope: Neural Representation of Different Types of Semantic Relations. *The Journal of Neuroscience*, *38*(13), 3303–3317. https://doi.org/10.1523/JNEUROSCI.2562-17.2018

Zhang, L., & Osth, Adam. F. (2024). Modelling orthographic similarity effects in recognition memory reveals support for open bigram representations of letter coding. *Cognitive Psychology*, *148*, 101619. https://doi.org/10.1016/j.cogpsych.2023.101619