UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Master Degree in Physics of Data

Final Dissertation

A first principle model of free recall

Thesis supervisor

Prof. Samir Suweis

Thesis co-supervisor

Prof. Carlo Sestieri

Candidate

Francesco Manzali

Academic Year 2020/2021

Abstract

Cognitive functions, such as memory, are traditionally thought to be too complex and variable across subjects to be described by the same kind of fundamental laws that explain physical events. One example is free recall, in which participants are briefly shown a list of words, and then try to recall the list items in an arbitrary order. Psychological studies have demonstrated that this basic type of memory highly depends on experimental conditions, such as the encoding time. As a consequence, recall has only been previously described through complex models with many free parameters, which do not have a clear biological basis and cannot be set before data collection.

Yet a recent physics-inspired approach [1], starting from few simple first principles, has shown how the average number of recalled items can be well predicted by the mathematical law $\sqrt{1.5\pi M}$, where M is estimated from the average number of items that participants would correctly identify in a two-alternatives forced-choice recognition test from a list of the same length. The agreement between the data and the theoretical predictions is particularly remarkable, as the model requires no free parameters. However, this universal law has only been tested with random memoranda, which differ considerably from the type of structured memories we experience in everyday life.

In the present work, we first present and explain in detail the above model and its foundations. We then test it on a free recall experiment with either lists of $L = 64$ random words or lists of $L = 64$ highly correlated words, each belonging to one of two categories, a situation known to emphasize the use of recall strategies. To do that, we have developed a web-app platform to perform these types of recall experiments.

Results show that, compared to lists of random words, participants recall significantly more items in the *categorized* case, while their performance in recognition remains the same. Moreover, they tend to recall more words from a category than the other, and to report items in short clusters.

These results are incompatible with the predictions from the above mentioned first-principle model, even when the model is extended to include semantic correlations between items. This is possibly because, in the categorized case, participants use higher-order strategies to go beyond the limits of pure retrieval processes.

Data and Code

The whole data collected from the experiments, along with the Python code used for analyzing it and producing the plots shown in chapter 2, is available upon request to the author.

Notation

- Italic nonbold typeface for scalars (e.g., a, V_i)
- Bold letters for vectors (e.g. $\boldsymbol{c}, \boldsymbol{v}$)
- Regular “upright” capital letters for matrices (e.g. M, S)

Contents

List of Figures

List of Tables

Introduction

Memory, the ability of storing information for later retrieval, is one of the most fundamental faculties of the human brain. It serves as a window on the past, a way to hold on acquired knowledge, skills, and personal experiences.

At first glance, memory appears to be extremely flexible. People can remember sensory information, feelings, episodes, sentences, or any arbitrary association between those. There are memories that require no verbal component, and memories that do not even need conscious thought, such as those encoding mechanical skills like riding a bicycle or playing an instrument. Memories can be formed at any moment, and they can last for a lifetime.

Yet, human memory does not work as binary storage on a hard disk. What goes in or out of memory is filtered by other context-dependent cognitive processes.

For instance, information is better stored if one pays attention to it. Conversely, a total lack of attention makes most details of a boring wait lasting hours immediately vanish. Even recall can modify a stored memory by filling gaps with some consistent, but new, information. Moreover, memories that are not recalled tend to slowly vanish into nothingness.

From this brief introduction, it should be clear that memory, as many aspects of the human brain, is incredibly complex. A complete understanding of how all of these phenomena are physically implemented in the brain is still very far away.

However, by sufficiently restricting the field, and by making the necessary simplifications, one may find some key insights regarding the inner workings of memory.

Outline

This work focuses on studying only *verbal memory*, that is the faculty of remembering words. Moreover, attention is paid mostly to the act of *recall*, i.e. that of retrieving words that are already “stored” somewhere in the brain. A brief review of the general characteristics of human memory is presented in the first part of chapter 1.

Although tackling a relevant problem in neuroscience, in this work we will follow a physics approach, which can be summarized by the following steps:

1. Start by formulating a few *first principles*, i.e. general basic *a priori* propositions that are consistent with the current experimental evidence.

2. Construct models that implement these principles. Simple models with fewer parameters (or none) that can be treated analytically are preferred to more complex models requiring extensive fine-tuning.

3. Compare the models' predictions with experimental data, either validating or rejecting their assumptions.

Chapter 1 follows this approach, reviewing the recent work on verbal memory by S. Romani, S. Recanatesi, M. Katkov, M. Naim, and their collaborators in a series of papers from 2013 to 2021 [2] [3] [4] [5] [1]. All relevant theory, mainly related to *Hopfield networks*, is presented as needed.

Their main result is a first principle model that can accurately predict the number of words that someone will, on average, be able to recall from a list of quickly presented random words. Since the focus is on *recall* only, to make the above prediction it is necessary to know the average number of words that are "available in memory" after the exposure, noting that only a subset of them may effectively be freely recalled, while the others necessarily need some additional hint to be accessed. Thus, the model requires as input a preliminary measure: a different list is presented using the same experimental setup, and the average number of words that can be *recognized* as belonging to it is estimated.

Other than that, the model from Naim et al. [1] has no adjustable parameters, and thus it is not (and cannot) be tuned to experimental data. Nevertheless, it is found to be surprisingly accurate.

The following chapter 2 aims to provide an independent verification of these results, by replicating the free recall experiment done by Naim et al. for lists of $L = 64$ words. Moreover, one of the core assumptions of the model — that of independent "strengths of association" between words stored in memory — is challenged by measuring recall performance on lists of words that are strongly semantically related.

After analyzing the data from this newly designed experiment, conclusions are drawn in the final chapter 3.

Chapter 1

Models of Memory

1.1 Preliminaries

In the past few decades, work in Psychology and Cognitive Neuroscience has led to the discovery of many characteristics of human memory.

First, memory is not a monolithic system, but consists of at least three "stores" [6, Ch. 1]:

- **Sensory memory**, which allows sensory information to persist in the brain for a brief moment ($< 1\,\mathrm{s}$), so that it can be processed and eventually more permanently stored. For instance, moving a sparkler in a dark room leaves a quickly fading trail, a kind of "afterimage" of the stimulus that produced it. Visual sensory memory (also known as *iconic memory*) can hold up to 12 items, but degrades so quickly that subjects have no time to report its full contents [7].

- **Short-Term Memory** (STM), which allows storing 7 ± 2 items of arbitrary information for a few seconds, up to a minute, without rehearsal [8]. A related concept is **Working Memory**, which enables accessing useful information needed for complex mental manipulation tasks (e.g. addition).

- **Long-Term Memory** (LTM), which can store arbitrary information for years, or even an entire lifetime, with apparently boundless capacity [9]. It is divided in **explicit** (or *declarative*) memory and **implicit** (or *nondeclarative*) memory. The former holds all memories that can be consciously recalled, such as personal experiences (*episodic memory*) or general knowledge (*semantic memory*), while the latter is devoted to skills and performance (e.g., riding a bike).

In general, there are three stages to consider for memory:

- **Encoding**: how external information is converted to a pattern of neural activities;

- **Storage**: how and where that pattern is conserved within the brain;

- **Retrieval**: how the internal representation is accessed as needed.

In practice, all memory stores and all memory stages interact with each other. For instance, content in LTM can drive attention towards specific features, which then affect the contents of both sensory memory and STM, determining what will be encoded. Things that are deemed particularly salient leave a stronger mark, and thus can better survive the passage of time and be recalled with more detail. However, retrieval can itself alter other stored memories or drive the encoding of additional sensory data.

1.1.1 Retrieval

To avoid most of this complexity, this work focuses mainly on the **retrieval** stage, ignoring as much as possible effects from the other two steps.

Thus, the initial assumption is that some information has been successfully stored in the brain. Surprisingly, this does not guarantee that it can be accessed.

From a cognitive point of view, retrieval consists of a search for a certain *target memory* [6, Ch. 8], guided by some snippets of information known as *retrieval cues* (or just *cues*). This can happen either *incidentally*, when some stimulus makes a memory pop up in thought (e.g. the smell of the sea reminding of a specific party at the beach), or *intentionally*, when one actively searches for a specific tidbit (e.g., the capital of France).

One popular theory on retrieval is the **spreading theory** [10], in which each memory is associated with an *activation level*. Only if this activation surpasses a threshold it is possible to access the memory in full. The search starts from the cues, which *spread* their activation to the other memories they are associated with, depending on how strongly they are linked. For instance, if one thinks of a *table*, a strong link would be *chair*, while a much weaker link would be *doily*. Associated memories spread their activation too, until the *target memory* receives enough activation to be accessed.

The picture becomes even more complex when one considers that memories are not monolithic, but are instead sets of features, which can act as cues by themselves. Then, to access the *target memory*, enough of its features must be activated, such that the whole *pattern* may be *completed*. This kind of database, in which the content of each item can be used to find it, is known as a **content addressable memory**.

There are many ways in which all of this may fail, resulting in being unable to access a memory, even with it being correctly stored in the brain. That is why sometimes keys get lost, with owners being unable to remember where they put them, until they are found again, and suddenly their location becomes obvious, with the steps leading to it appearing vivid in memory.

In these circumstances, the cues used for the memory search may have been not relevant, or too weakly associated with the target, or too few. Recalling may fail even if the context (or frame of mind, or mood) is different from the one that was present when the target memory was stored [6, Ch. 8]. That is why sometimes memories that seem forgotten are immediately recalled when the correct *cues* are encountered.

This suggests that when the stimulus that generated the target memory is presented, that memory should be more easily recalled, or at least it should feel *familiar*. This is the basis of **recognition memory**, i.e. the "ability to decide whether a given stimulus has been previously encountered in a particular context" [6, Ch. 8]. When encountering some stimulus, one can usually classify it as *new*, i.e. not seen before, or *old*, if it evokes a sense of familiarity ("*know*" response) or if they can recollect some specific information about its previous presentation (“*remember*” response).

1.1.2 Free recall

Experimentally, one common paradigm to study retrieval is **free recall**. Participants are asked to remember a sequence of items (usually words, but they can be also pictures or sounds), which are presented in a controlled fashion, generally at a fixed rate. Then they *freely* report all the items that they can recall, in any order.

The performance is highly dependent on the details of the experiment, especially the presentation rate. However, there are some regularities that can be studied [11]. For instance, the probability that a word is recalled depends on the position it appeared during presentation [12]. People tend to better recall the words shown at the very start of the experiment (known as *primacy* effect). This is because, when one is asked to remember a sequence, they usually start *rehearsing* the few first items. This strategy, however, becomes quickly infeasible as more words are presented, and the small STM buffer is filled. The primacy effect usually involves only the first 3 to 4 items in the sequence, and is more prevalent when the rate of presentation is sufficiently slow, since one needs time to properly rehearse.

Conversely, the words that are presented last are also better recalled (known as *recency* effect). This is because, when recall is initiated immediately after presentation, these items are still contained in the STM buffer. The recency effect affects up to the last 8 words, and is reduced when recall is delayed, and especially when a distracting task, such as counting digits, is interposed between presentation and recall [13].

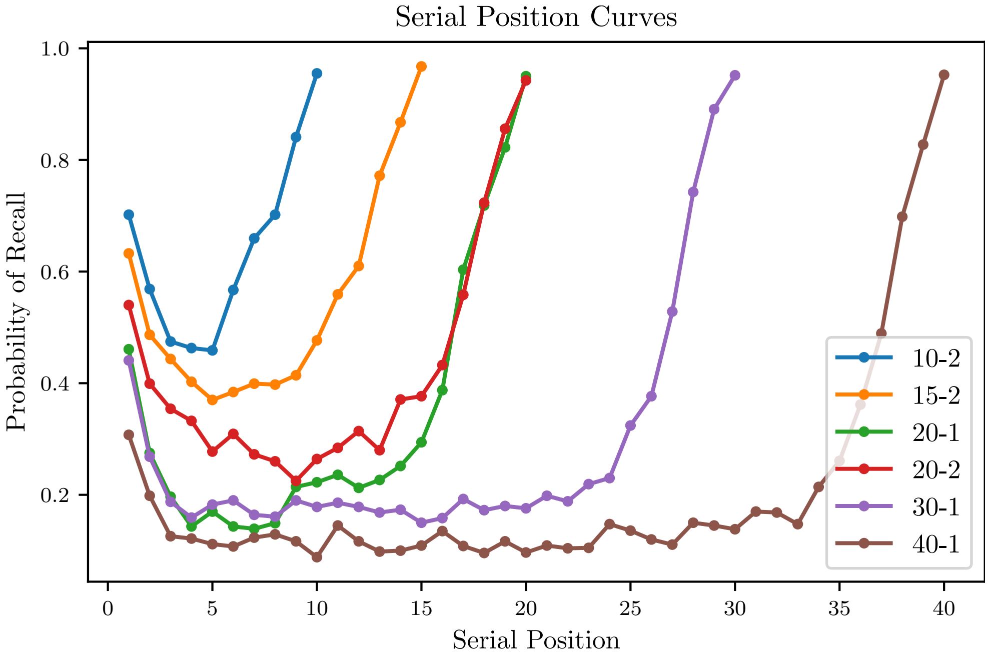


Figure 1.1 – Serial position curves for the free recall experiments done by B. Murdock in 1962 [12]. Each curve is labelled by a tuple x-y, where x is the number of words included in the presented list, and y is the number of seconds reserved to each word (s/word). So, for instance, 40-1 denotes a setup in which 40 words are presented, each for just 1 s.

The plot of the probability of recalling a word given its serial position is known as the *serial position curve*. Results from the experiment performed by B. Murdock in 1962 [12] are shown in fig. 1.1. Note how the probability is high for the first positions (*primacy* effect) and for the last ones (*recency* effect), while it is approximately constant for the middle elements, at least in the case of sufficiently long lists.

Additionally, the *order* of recall is not random. Participants tend to report together words that were adjacent in the presented list, an effect known as *contiguity effect* [14].

However, what about general performance? The fraction of recalled items surely depends on the rate of presentation: a slower pace improves recall. For instance, in the experiments by B. Murdock (1962) [12, Table 1] the average number of recalled words for a list of 20 items was 6.87 ± 1.16 for a presentation at 1 s/word, and increased to 8.53 ± 2.08 when the presentation was slowed to 2 s/word.

More interestingly, if the pace is kept constant, but more words are presented, then more words will be recalled. For example, in the case of 40 words at 1 s/word on average 8.24 ± 1.08 are typically recalled [1], and the pattern holds for even longer lists. When presented with 64 or more words, people can usually recall more than 10 words [1, fig. 2.a], meaning that there is sufficient capacity for storing all items of the shorter lists. Yet, lists of 10 items shown at the same rate are not perfectly recalled.

1.1.3 Tests of recognition

Perhaps, the items are in memory, but they cannot be easily accessed during the free recall experiment, because the correct cue is missing [15]. This is a widespread occurrence: for instance, most have experienced a "tip-of-the-tongue" state, in which they have something specific in mind, but cannot find the correct word for it — even if they can accurately remember some of its features [16].

Therefore, a better way for testing if an item is in memory is to measure *recognition* instead. The idea is to present participants with a set of stimuli. Some are from the sequence that was previously shown, while the others (called *lures*, *foils* or *distractors*) are not. Then, for each of them, participants are asked to choose whether that item was shown or not.

Alternatively, one *old* item is shown together with one or more lures, and the subject must correctly identify it between all the options. This is known as a *forced-choice recognition test*.

However, in this case, if a participant chooses the correct option in a trial, one cannot be sure that they have effectively recognized the item. Perhaps, all choices had an equal level of familiarity, and the subject just picked one at random.

Suppose that the actual probability of correctly recognizing an item among k alternatives is p. This means that over N recognition trials, the subject will perceive, on average, pN items as being distinctively familiar. In each of these cases, they will select the correct choice with probability 1. In all others, they will instead guess, choosing any option with equal probability. Then, the average number of items that are correctly identified S, either by recognition or random chance, is:

$$\begin{aligned}
S = N\Big[&\mathbb{P}[\text{item recognized}]\mathbb{P}[\text{correct choice}|\text{item recognized}]+\\
&+\mathbb{P}[\text{item not recognized}]\mathbb{P}[\text{correct choice}|\text{item not recognized}]\Big] =\\
&= N\Big[p \cdot 1 + (1-p)\cdot \frac{1}{k}\Big]
\end{aligned}$$

Solving for p leads to:

$$p = \frac{1}{k-1}\Big(\frac{kS}{N} - 1\Big)$$

Generally, forced-choice recognition tests involve just two alternatives ($k = 2$): one *old* item and one *lure*. Then the probability of recognizing a word is:

$$p = 2\frac{S}{N} - 1 \tag{1.1}$$

where S/N is the fraction of correctly identified items S in the N recognition trials. Then, if the presented list has L words, the number M of words that would be recognized on average is:

$$M = pL = 2L\frac{S}{N} - L$$

In fact, generally, recognition trials are done only on a subset of the shown items.

Clearly, this is a simplification: "familiarity" is a graded response, not an all-or-none signal. Perhaps there are trials where the *old* item has a level of familiarity which is only slightly above that of *lures*, meaning that it will be selected with a probability above that of chance. These cases are neglected for the sake of simplicity. In fact, people tend to have different personal *thresholds* for familiarity, and these will (hopefully) average out when considering a large enough sample.

In any case, the number of words that can be recognized is significantly larger than that of the words that can be recalled. For instance, in a 1973 experiment by L. Standing, when 200 words are visually shown at 5 s/word, participants recognized (on average) 134.5 (67%) of them, but were able to recall only 24.5 ± 10.0 (12%) of them [9, Table IV].

The same study shows that pictures are significantly easier to recognize than words. For instance, short sequences of 20 or 40 images can be recognized with 90% probability, or even with certainty (100%) if the images are chosen to be particularly vivid. In a remarkable finding, after a single presentation of 10 000 images, participants were estimated to be able to recognize 6600 (66%) of them.

1.1.4 Power laws

When increasing the list length, the absolute number of items that can be recognized or recalled increases sublinearly, according to a power law [9]:

$$R = k\,P^m$$

where R is the number of recognized/recalled items, P is the number of presented items, while k and m are the fit parameters, with the exponent m being ≤ 1. For recall $m \approx .5$, while it is higher for recognition, at $m \approx .9$.

Interestingly, this kind of relation seems characteristic of retrieval processes in general. For instance, it reappeared in a "free output" task proposed by D. J. Murray [17], where participants were asked to produce words belonging to some specific categories (e.g. "a precious stone”). In this case, no list was presented during the experiment, but subjects recalled words from their own vocabulary. The number of words of a given category that people were expected to know was estimated from a separate experiment. In this way, one could plot the relation between the number of recalled words against the number of known words, and a power law dependency was found.

The same kind of relation appeared when a graphemic cue was used instead [18], for instance by asking for words starting with a certain letter, or that have a given letter in a specific position.

Power laws appeared also in *cued* recall experiments [19]. In this case, words were presented along with specific cues (e.g. `stallion` — `a horse`). Then, during retrieval, the cue was shown, and the participants were asked to report the word it was associated with. Again, the average number of words successfully recalled was found to scale as a power function of the number of words that were presented. The same scaling, just with different parameters, appeared for all other possible setups: with or without cues at presentation, and with or without cues during retrieval.

All these findings suggest some fundamental principles behind recall. Perhaps retrieval involves a single base algorithm, which is adapted to different tasks. This would explain why distinct kinds of experiments find the same functional relation between the number of words that can (in principle) be accessed, and the number of words that are recalled.

1.2 Previous models of retrieval

Lots of data has been collected regarding retrieval processes, mostly with free recall or forced-choice recognition experiments. Preliminary data analysis can provide fits, for instance, of the serial position curve (fig. 1.1), or of the power law scaling of the number of items recalled/recognized as a function of the number of items presented.

From the previous discussion, it is clear that retrieval involves many interacting cognitive processes, which are not understood well enough to build accurate computational models, especially at the level of biology. There have been bottom-up approaches, such as the one in [20], starting at the level of networks of neurons, but they fail to reproduce the experimentally observed scaling of recalled/recognized items (N_r) with respect to the presented list length (L).

1.2.1 Search for Associative Memories (SAM)

On the other hand, there are phenomenological models that can successfully fit empirical data. One of the most popular is the “Search of Associative Memory”, or SAM [21].

For free recall, SAM postulates that, during presentation, a “structure” linking items is generated and stored in LTM, defining the strength of associations between items. Namely, there is a buffer of size r, representing the set of items that a subject can rehearse during presentation. The buffer starts empty, and items are added to it until it is full. Then, each new item replaces a random old item in the buffer.

Each item I_i is associated with the general context C proportionally to the time t_i spent in the buffer: $S(C, I_i) = a\, t_i$, where $S(C, I_i)$ denotes the association strength between C and I_i.

Similarly, items I_i and I_j that co-occur in the buffer for a time $t_{ij} \neq 0$ are associated to each other with a strength $S(I_i, I_j) = S(I_j, I_i) = b\, t_{ij}$. There is also a self-association proportional to time in the buffer: $S(I_i, I_i) = c\, t_i$.

Finally, the items I_i and I_j that never appear together in the buffer still have a "residual" association with strength $S(I_i, I_j) = d$ (for $t_{ij} = 0$).

When retrieval begins, first all items still present in STM are reported, then the LTM structure is used to remember the others. Items are sampled using *cues* $Q_1, \dots, Q_M$, according to the following multiplicative rule:

$$P_S(I_i | Q_1, Q_2, \dots, Q_M) = \frac{\prod_{j=1}^M S(Q_j, I_i)}{\sum_{k=1}^L \prod_{j=1}^M S(Q_j, I_k)} \tag{1.2}$$

where L is the number of presented items, while M is the number of cues used to search for an item. The idea behind this rule is that the memories with the highest probability of being recalled are the ones that share a strong association with all the cues $Q_1, \dots, Q_M$, meaning that they are most likely to be "completed" by the pattern of cues.

At the start of free recall, the only cue available is the context C, and an item I_i is sampled using (1.2). Then, a check is done to see if I_i can be recalled, i.e. if the information that was sampled from the previous step is enough to report the "name" of I_i (e.g. the word). This happens with a probability given by, in the general case:

$$P_R(I_i | Q_1, Q_2, \dots Q_M) = 1 - \exp\Big(-\sum_{j=1}^M S(Q_j, I_i) \Big)$$

Note that this involves an additive combination of associative strengths, meaning that I_i is more likely to be recalled if it is strongly associated to any of the cues. As expected, as $\sum_j S(Q_j, I_i)$ varies between 0 and $+\infty$, P_R goes from 0 to 1.

If recovery succeeds, I_i is reported. This alters the LTM structure, since retrieval can form new memories. In particular, the context association $S(C, I_i)$ increases by e, and the self-association $S(I_i, I_i)$ increases by g.

Then, the process is repeated. This time, both C and I_i are cues for sampling another item I_j. If I_j was already sampled before, or if the retrieval was unsuccessful, then a failure event is recorded. In general, whenever $K_{\max}$ such failures occur, recall stops definitively.

Moreover, each cue combination that does not produce new recalls can be used at most for $L_{\max}$ times. If this limit is surpassed, the last item is removed from the cues, and only the context C is used to sample the following items.

Finally, if a cue I_i is successfully used to retrieve a new I_j, then also the association $S(I_i, I_j)$ is increased by an amount f.

1.3 A first principle approach

While SAM can produce good fits of experimental data (e.g., it can account for the *primacy* and *recency* effects, and predict how many items will be recalled), it does so using many parameters (a, b, c, d, e, f, g and r just for free recall, along with the additional $K_{\max}$ and

$L_{\max}$), which lack a definite biological meaning. Moreover, several combinations of them can achieve good results, and it is not clear which to choose or discard when modelling a new task, meaning that the model must be "tuned" every time. There is also evidence that some of its assumptions may be flawed [22].

Other popular models for retrieval, such as the "Temporal Context Model" (TCM) [23], the Context Maintenance and Retrieval (CMR) model [24], the SIMPLE model [25] and the OSCillator-based Associative Recall (OSCAR) [26] exhibit a similar trend, with many free parameters needed to account for experiments, which are varied between experimental conditions, or even between different list lengths within the same experiment. In certain cases, the number of parameters approaches that of data points: in the CMR paper [24, Table 1] as many as 14 parameters are fitted to just 93 points. Inevitably, this has led to harsh criticism [27].

In a review from 2011 [28], D. Hintzman expressed how memory research has become fixated on precisely fitting the details of standard and limited experimental paradigms instead of focusing on a wide understanding of memory in general.

In this picture, the recent work of M. Katkov, S. Romani, and M. Tsodyks [5] stands aside, in that it aims for a model of general features of all recall processes, possessing few or no tunable parameters [1]. Before assuming complex mechanisms, they start by building good first principles, following the approach of physics.

The main idea is that, while there is not yet enough understanding to fully model human memory, or even just retrieval processes, progress can be made by identifying a key set of general postulates, consistent with current knowledge, which can help to build predictive frameworks. In particular, such "first principles" should be both "basic" and "unifying". They should not be focused on details, but rather link descriptions at multiple levels — for instance, biological systems (networks of neurons) to cognitive processes (memory association, search and retrieval).

Newton's laws provide an example of "good" first principles in physics. They do not specify any exact nature of forces, or how they can be produced, and so can be applied to lots of different phenomena. At the same time, they link motion at small and large scales within the same framework: as an apple falls on the ground, a planet orbits a star.

In the case of the retrieval of memories, the proposed first principles are as follows [5]:

1. **Encoding Principle**: memory items are stored in the brain by groups of neurons in a memory network. Retrieval happens when one of these groups of neurons is activated.
2. **Associativity Principle**: in the absence of other sensory cues, each recalled item serves as an **internal cue** for retrieving the next item.

According to these principles, memories are encoded in a distributed way, and when a group of neurons is activated (due to the retrieval of an item), it triggers the retrieval of another item. In other words, there is a link between a biological process (neurons firing in a pattern) and a cognitive process (the recall of an item). Moreover, the association is simplest when there are no other sensory cues, as it happens during a free recall experiment.

In any case, all details of these processes are not specified. For instance, the first principle says nothing about where in the brain items are encoded, or if they are organized according to some scheme or hierarchy. Similarly, the associativity principle does not specify the mechanics of association, or how a group of neurons can activate another group to trigger a new retrieval.

However, the two principles define a framework for studying recall. Following it, one can fill in the details to obtain a model and then experimentally test its predictions. This is explored in the next few sections.

1.4 Neural network model

This section reviews the first model of recall by Romani et al. [2], which follows the two principles stated in sec. 1.3. Specifically:

- The **encoding principle** is implemented by considering an attractor neural network, i.e. a system of interconnected units (neurons) which evolves according to a set of dynamical rules towards certain states (attractors) forming the "neural representation" of memory items. If such a system starts sufficiently close to an attractor, then it will evolve towards it. This emulates the process of "pattern completion" happening in the brain when a memory is recalled from a set of cues, which is the defining property of a *Content Addressable Memory*.
- An additional cyclic dynamics is added to allow transitions between attractor states, in which one acts as *cue* to activate (retrieve) the other, as specified by the **associativity principle**.

The objective is to explore the possible consequences of the principles, not to build a detailed model of the biological processes underlying retrieval. This will later allow to build a simplified model, which can be studied analytically and compared with data.

However, the chosen dynamics should be at least plausible according to the current knowledge of the human brain, except for some deliberate violations done to simplify the computations.

1.4.1 Hopfield Model

There are many types of attractor neural networks. One of the simplest and most studied is the Hopfield Model [29], which is here used as a starting point to model free recall.

Consider a system of N units, called neurons, each with a binary state: $V_i \in \{0, 1\}$. Neurons that are firing have $V_i = 1$ ("on" state), while neurons that are not firing, or that fire at a rate too low, have $V_i = 0$ ("off" state).

Neurons are connected to each other by synapses of different strengths. In particular, the strength of a connection $j \to i$ is denoted as $\mathrm{T}_{ij} \in \mathbb{R}$, with $\mathrm{T}_{ij} = 0$ denoting the absence of such link.

Synapses transmit signals between neurons. Specifically, each neuron receives as input h_i the sum of the activations of the neurons it is connected to, weighted by the strength of the synapses forming the links:

$$h_i \equiv \sum_{\substack{j=1 \\ j\neq i}}^{N} \mathrm{T}_{ij} V_j \tag{1.3}$$

Note that the sum skips over the $j = i$ term, since no self-loops are allowed in this model.

If the input h_i surpasses a local threshold U_i, then the i-th neuron becomes (or remains) active. Conversely, if $h_i < U_i$, the i-th neuron is deactivated:

$$V_i(t+1) = \begin{cases} 1 & h_i(t) > U_i \\ 0 & h_i(t) < U_i \end{cases} \Rightarrow V_i(t+1) = \Theta(h_i(t) - U_i) \tag{1.4}$$

where Θ is the Heaviside step function. Conventionally, when $h_i = U_i$, the neuron's state is left unchanged.

Updates can be asynchronous, in that each neuron independently checks its inputs with a mean attempt rate W, or synchronous, if all neurons are updated together at every time step of the simulation. Both methods eventually lead the network to an attractor state [30, sec. III], with the asynchronous way having better convergence properties [31].

In this work, however, synchronous updates will be considered for both simplicity and speed.

To store memories in the network, the synapse strengths T_{ij} are chosen so that some desired patterns of neural activations, representing the encodings to be stored, are set as attractors for the dynamics. Suppose there are n of them, each a vector of N activations $\{V_i^s\}_{i=1,\ldots,N}$ $(s = 1, \ldots, n)$.

For now, these patterns are assumed to be the realization of uncorrelated and unbiased random variables, so that it is easier to compute averages involving them. In other words, each entry V_i^s has equal probability of being 0 or 1, and these probabilities do not change if the other entries or patterns are known, meaning that their joint distribution p can be completely factorized:

$$p[\{V_i^s\}_{\substack{i=1,\ldots,N \\ s=1,\ldots,n}}] = \prod_{i=1}^{N}\prod_{s=1}^{n} p(V_i^s)$$

$$p(V_i^s) = \frac{1}{2}\delta(V_i^s - 1) + \frac{1}{2}\delta(V_i^s) \Leftrightarrow \mathbb{P}[V_i^s = 0] = \mathbb{P}[V_i^s = 1] = \frac{1}{2}$$

where δ denotes the Dirac delta function.

Thus, when averaging over many possible choices for the patterns, one gets:

$$\langle V_i^s \rangle = \frac{1}{2} \qquad \langle V_i^s V_j^{s^*} \rangle \underset{(i\neq j)}{=} \frac{1}{4} \qquad \langle V_j^{s^*} V_j^s \rangle = \begin{cases} \frac{1}{4} & s \neq s^* \\ \frac{N}{2} & s = s^* \end{cases} \tag{1.5}$$

For instance, in the case of $V_i^s V_j^{s^*}$ with $i \neq j$, different entries of two patterns (or the same one) are always independent. Thus, the full computation would be:

$$\langle V_i^s V_j^{s^*} \rangle = \sum_{a,b \in \{0,1\}} a\, b\, \mathbb{P}[V_i^s = a, V_j^{s^*} = b] = \frac{1}{4} 1 \cdot 1 + \frac{1}{4} 1 \cdot 0 + \frac{1}{4} 0 \cdot 1 + \frac{1}{4} 0 \cdot 0 = \frac{1}{4}$$

Then, the entries T_{ij} must be chosen so that the patterns $\{V_i^s\}$ are attractors for the dynamics. Specifically, this means that if the network's state is set to any pattern $V_i = V_i^s$, then applying the update rule (1.4) would not change the state. In general, this happens for all states $\boldsymbol{V} = (V_1, \ldots, V_N)$ that satisfy:

$$(2V_i - 1)(h_i - U_i) > 0 \qquad \forall i = 1, \ldots, N \tag{1.6}$$

Note that $2V_i - 1 = +1$ if $V_i = 1$, and -1 if $V_i = 0$. If $V_i = 1$, one needs $h_i > U_i$ to maintain it, while for $V_i = 0$, $h_i < U_i$ is necessary. In both cases, both terms of (1.6) share the same sign.

Suppose there is a single pattern $\{V_i^1\}_{i=1,\ldots,N}$, and set the initial state of the network to it. The objective is to "stabilize" it under the dynamics: namely, neurons that must remain *off* should receive a *low* input, while neurons that should remain *on* should receive a *high* input.

Then, one idea is to use all neurons that are active to propagate excitatory/inhibitory signals to all others. In practice, suppose that the first neuron $V_1^1 = 1$, and consider the connection to another neuron V_i^1. The contribution of the first neuron to the activation of the i-th neuron is exactly T_{i1}. Intuitively, this should be *positive* $(= +|c|)$ if V_i^1 should remain $+1$, or *negative* $(= -|c|)$ if instead one needs $V_i^1 = 0$. This results in the following rule:

$$\mathrm{T}_{ij} = |c|(2V_i^1 - 1) \tag{1.7}$$

The constant c is a matter of convention, and it is usually set to $1/N$. This is because, in a fully connected network of N neurons, each neuron receives an input $\propto N$. Thus, by dividing the synapse weights by N, the input of each neuron is normalized to a value independent of the network's size.

In any case, it is mathematically convenient to choose a symmetric T, such that $\mathrm{T}_{ij} = \mathrm{T}_{ji}$, since this will allow to easily prove the convergence to stable patterns (and it also makes calculations easier). This can be done by "symmetrizing" the above expression:

$$\mathrm{T}'_{ij} \equiv N\mathrm{T}_{ij}\mathrm{T}_{ji} = \frac{1}{N}(2V_i^1 - 1)(2V_j^1 - 1)$$

and from this point forward, T'_{ij} will be renamed as T_{ij}.

Note that an asymmetric T_{ij} can still lead to effective retrieval of memories [32], but it adds nonthermal noise to the system, i.e. noise that is not driven by randomness in the update rule.

The above choice for T_{ij} says that if two neurons in a pattern have the same value (0 or 1), they will be connected by a positive weight $\mathrm{T}_{ij} = +1$. Conversely, if two neurons have a different value (one 0, the other 1), then the weight between them will be negative.

Intuitively, this is consistent with the stability requirements. If the state is that of a pattern, two neurons that are active will excite each other, so that they remain active, while an active neuron connected to an inactive one will inhibit it, so that it stays inactive.

The same logic can be applied in the presence of more than one pattern, by just summing over all of them:

$$\mathrm{T}_{ij} = \frac{1}{N}\sum_{s=1}^{n}(2V_i^s - 1)(2V_j^s - 1) \qquad \mathrm{T}_{ii} = 0 \tag{1.8}$$

This rule is nothing else than the Hebbian rule for synaptic learning, which states that "neurons that fire together, wire together" [33], with the additional feature of allowing inhibitory links. As for Hopfield networks, it serves as an *ansatz* for choosing the synaptic strengths so that the patterns $\{V_i^s\}$ behave as attractors for the dynamics, as proposed in the original paper [29].

To verify that it indeed works, consider the input received by each neuron i when the state is set to a pattern $\{V_i^{s^*}\}$, if the T_{ij} from (1.8) are used:

$$\begin{aligned} H_i^{s^*} &\equiv \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\mathrm{T}_{ij}V_j^{s^*} = \frac{1}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\sum_{s=1}^{n}(2V_i^s - 1)(2V_j^s - 1)V_j^{s^*} = \\ &= \frac{1}{N}\sum_{s=1}^{n}(2V_i^s - 1)\Big[\sum_{\substack{j=1\\ j\neq i}}^{N}V_j^{s^*}(2V_j^s - 1)\Big] \end{aligned} \tag{1.9}$$

When N is sufficiently large, the term in the square brackets can be approximated by its average:

$$\sum_{i=1}^{N}X_i \approx N\langle X\rangle$$

Then, if $s \neq s^*$:

$$\sum_{\substack{j=1\\ j\neq i}}^{N}V_j^{s^*}(2V_j^s - 1) = \sum_{\substack{j=1\\ j\neq i}}^{N}[2V_j^{s^*}V_j^s - V_j^{s^*}] \approx (N-1)[2\langle V_j^{s^*}V_j^s\rangle - \langle V_j^s]\rangle \underset{(1.5)}{=} (N-1)\Big[\frac{2}{4} - \frac{1}{2}] = 0 \tag{1.10}$$

For $s = s^*$, instead:

$$\sum_{\substack{j=1\\ j\neq i}}^{N}V_j^{s^*}(2V_j^s - 1) = \sum_{\substack{j=1\\ j\neq i}}^{N}2(V_j^{s^*})^2 - V_j^{s^*} = \sum_{\substack{j=1\\ j\neq i}}^{N}2V_j^{s^*} - V_j^{s^*} = \sum_{\substack{j=1\\ j\neq i}}^{N}V_j^{s^*} \approx (N-1)\langle V_j^{s^*}\rangle \underset{(1.5)}{=} \frac{N-1}{2} \tag{1.11}$$

Substituting (1.10) and (1.11) back into (1.9) leads to:

$$H_i^{s^*} \approx (2V_i^{s^*} - 1)\frac{N-1}{2N} \underset{N\gg 1}{\approx} \frac{1}{2}(2V_i^{s^*} - 1) \tag{1.12}$$

Assuming no threshold ($U_i = 0$), note that $H_i^{s^*} > 0$ if $V_i^{s^*} = 1$, and $H_i^{s^*} < 0$ if $V_i^{s^*} = 0$.

This means that, with the choice from (1.8) for T_{ij}, if the network's state is set to that of a stored pattern, then it will remain constant under the Hopfield dynamics.

However, while the sum over $s \neq s^*$ averages to 0, it has a nonzero variance, which introduces noise in the inputs. To estimate it, consider the square of (1.10), with $s \neq s^*$:

$$\begin{aligned}\Big[\sum_{\substack{j=1\\j\neq i}}^{N} V_j^{s^*}(2V_j^s-1)\Big]^2 &= \sum_{\substack{j=1\\j\neq i}}^{N}\sum_{\substack{k=1\\k\neq i}}^{N} V_j^{s^*}V_k^{s^*}(2V_j^s-1)(2V_k^s-1) = \\ &= \sum_{\substack{j=1\\j\neq i}}^{N}\sum_{\substack{k=1\\k\neq i,j}}^{N} V_j^{s^*}V_k^{s^*}(2V_j^s-1)(2V_k^s-1) + \sum_{\substack{j=1\\j\neq i}}^{N}(V_j^{s^*})^2(2V_j^s-1)^2 \qquad (1.13)\end{aligned}$$

Note that in the term with $k \neq j$, all terms are independent, meaning that their average can be factored. In particular, note that $\langle(2V_j^s-1)(2V_k^s-1)\rangle = \langle(2V_j^s-1)\rangle\langle 2V_k^s-1\rangle = 0$. Thus, the first term contributes nothing to the variance.

On the other hand, since $V_j^s \in \{0,1\}$, $(V_j^{s^*})^2 = V_j^{s^*}$ and $(2V_j^s-1)^2 = 1$. Therefore, the second term, with $j = k$, becomes:

$$\sum_{\substack{j=1\\j\neq i}}^{N} \underbrace{(V_j^{s^*})^2}_{V_j^{s^*}}\underbrace{(2V_j^s-1)^2}_{1} = \sum_{\substack{j=1\\j\neq i}}^{N} V_j^{s^*} \approx \frac{N-1}{2}$$

Taking into account that there are $n-1$ terms of this type (since they are in a sum over $s \neq s^*$), and multiplying by the square of the prefactor $1/N$ (since $\mathrm{Var}(aX) = a^2\mathrm{Var}(X)$), the total variance is $\approx (n-1)/(2N)$, leading to a noise term σ:

$$\sigma \approx \sqrt{\frac{(n-1)}{2N}}$$

which is the effect of the so-called **crosstalk** between patterns. Intuitively, the more patterns are stored in the network, the more they "interfere" with each other. If the noise is too strong, it can change the sign of $H_i^{s^*}$, introducing errors in the stored memories:

$$H_i^{s^*} \approx \frac{1}{2}(2V_i^{s^*}-1) \pm \sigma$$

The value of σ can be used to estimate the probability of these errors, which is given by the probability that the sum over $s \neq s^*$ surpasses, due to a random fluctuation, the term with $s = s^*$. Using a Gaussian approximation (valid for $nN \gg 1$), one gets:

$$\mathrm{P}_{\text{error}} = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{\frac{1}{2}}^{+\infty} \exp\Big(-\frac{x^2}{2\sigma^2}\Big)\,\mathrm{d}x \qquad (1.14)$$

For instance, for $N = 100$ and $n = 10$, one gets $\mathrm{P}_{\text{error}} = 0.0092$. So, an estimate of the probability that all 100 bits in a pattern are correctly retrieved is $(1-\mathrm{P}_{\text{error}})^N \approx e^{-N\mathrm{P}_{\text{error}}} \approx 0.4$. This result is compared to a numerical simulation in fig. 1.2.

These arguments suggest that, if N is made very large, a fixed number n of patterns can be

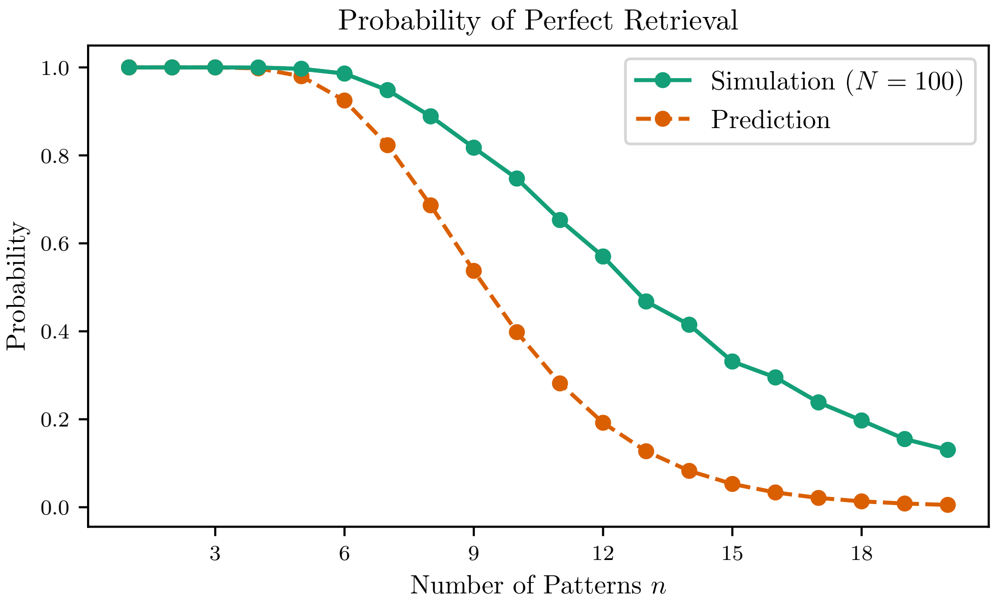


Figure 1.2 – Probability that a memory will be perfectly retrieved, as estimated by $\exp(-N\mathrm{P}_{\mathrm{error}})$ (orange line) or by averaging 1000 simulations (green line). For each simulation, n binary unbiased patterns of $N = 100$ neurons are randomly chosen, and the synaptic strengths T_{ij} are computed. The fraction of patterns that satisfy the stability condition (1.6) forms an estimate of the probability of perfect retrieval, which is then averaged over 1000 trials.

perfectly stored in the network. A more precise probabilistic argument [30, sec. VI] shows that if *all* n patterns are supposed to be perfectly retrieved, then there cannot be more than $N/(4\log N)$ as $N \to \infty$ of them. If a small error rate ($\sim$ 1.5%) is deemed acceptable, then $n \sim 0.14N$, as it can be shown using methods from statistical mechanics [34].

For now, suppose that n patterns have been successfully stored, i.e. they are fixed points for the dynamics. If the network is initialized in any state, will the dynamics converge to one of these states?

It can be shown that the network will converge from any state to a fixed point. The idea is to define an "energy" for the model, show that it is bounded, and that whenever a neuron changes its activation according to the update rule (1.4) the energy decreases.

For the Hopfield model, the energy is defined as follows:

$$H = -\frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} \mathrm{T}_{ij} V_i V_j + \sum_{i=1}^{N} V_i U_i \tag{1.15}$$

For a given choice of T_{ij} and U_i, since $V_i \in \{0, 1\}$, H is clearly bounded. Then, consider a transition of a single neuron k, from $V_k \to V_k'$. The change in energy depends only on the terms involving the k-th neuron, since all others remain the same:

$$\begin{aligned}\Delta H \equiv{}& H(V_0,\dots,V_{k-1},V'_k,V_{k+1},\dots,V_N) - H(V_0,\dots,V_{k-1},V_k,V_{k+1},\dots,V_N) = \\ ={}& -\frac{1}{2}\underbrace{\sum_{\substack{j=1\\ j\neq k}}^{N} \mathrm{T}_{kj}V'_kV_j}_{(i=k)} - \frac{1}{2}\underbrace{\sum_{\substack{i=1\\ i\neq k}} \mathrm{T}_{ik}V_iV'_k}_{(j=k)} + V'_kU_k \\ & + \frac{1}{2}\underbrace{\sum_{\substack{j=1\\ j\neq k}}^{N} \mathrm{T}_{kj}V_kV_j}_{(i=k)} + \frac{1}{2}\underbrace{\sum_{\substack{i=1\\ i\neq k}} \mathrm{T}_{ik}V_iV_k}_{(j=k)} - V'_kU_k\end{aligned}$$

Consider the first two sums. If the index i in the second is renamed to j, one gets:

$$-\frac{1}{2}\sum_{\substack{j=1\\ j\neq k}}^{N} \mathrm{T}_{kj}V_jV'_k - \frac{1}{2}\sum_{\substack{j=1\\ j\neq k}} \mathrm{T}_{jk}V_jV'_k$$

The two sums are then exactly equal for any symmetric choice of the synaptic strengths ($\mathrm{T}_{ij} = \mathrm{T}_{ji}$). In this way:

$$\begin{aligned}\Delta H &= -\sum_{\substack{i=1\\ i\neq k}}^{N} \mathrm{T}_{kj}V'_kV_j + V'_kU_k + \sum_{\substack{i=1\\ i\neq k}}^{N} \mathrm{T}_{kj}V_kV_j - V_kU_k = \\ &= -(V'_k - V_k)\Big[\underbrace{\sum_{j\neq k} \mathrm{T}_{ik}V_j}_{h_k} - U_k\Big]\end{aligned}$$

Since the update happens according to (1.4), if $h_k > U_k$, then k goes from $0 \to 1$, and so $V'_k - V_k = 1$ and $\Delta H < 0$. If instead $h_k < U_k$, then k goes from $1 \to 0$, and so $V'_k - V_k = -1$ and again $\Delta H < 0$.

Thus, any update according to (1.6) decreases the energy of the network, until a local minimum is reached. At that point, no single update can lower H anymore, and so no new transition can happen: the network has reached a fixed point. This happens for any given initial condition, with the only needed assumption being $\mathrm{T}_{ij} = \mathrm{T}_{ji}$.

With statistical mechanics, it can be shown that there are fixed states that do not correspond to the stored patterns, but to "mixtures" of them [35]. However, they are at best metastable, with an energy that is greater than that of "memory states" by an extensive amount [34]. So, if N is sufficiently large to store all n desired patterns, dynamics will always converge to them, especially if a small amount of thermal noise is added in the network, to avoid getting "stuck" in a local minimum.

In summary, this analysis shows the following:

- When a Hopfield network is initialized in a random state, it converges in a finite amount of time to a stable state (attractor), which then remains constant.

- Choosing the synaptic strengths T_{ij} as in (1.8) allows to encode a set of n desired patterns $\{V_i^s\}$ as attractors for the dynamics, as long as their entries are independent and either 0 or 1 with uniform probability.
- At fixed N, if the patterns to store n are too many ($\sim N/(4\log N)$), the interference between them can cause errors during retrieval.

From this point onwards, N will be assumed to be sufficiently high to perfectly store all n patterns.

1.4.2 Sparse encoding

Before discussing the retrieval dynamics, however, it is necessary to address a critical limitation of the model just presented: the patterns are assumed to be *unbiased*. This means that when a pattern is active, on average, half of the neurons in the whole network will be activated.

Biologically, this is implausible: doing so would be very inefficient in terms of energy. Experiments support the idea that the brain uses a "sparse code" to store knowledge, i.e. just a small fraction of neurons is active in a memory network at a time [36].

However, the Hebbian rule (1.8) is not compatible with such *biased* patterns.

To see that, consider the same computation done for (1.12). This time, however, suppose that:

$$\mathbb{P}[V_i^s = 1] = p \qquad \mathbb{P}[V_i^s = 0] = 1 - p$$

It is convenient to define $\xi_i^s \equiv 2V_i^s - 1$, such that:

$$\mathrm{T}_{ij} = \frac{1}{N}\sum_{s=1}^{n} \xi_i^s \xi_j^s$$

Then, the input of the i-th neuron when the network is set to the s^* pattern is given by:

$$H_i^{s^*} = \frac{1}{N}\sum_{s=1}^{n}\sum_{j\neq i}^{N} \xi_i^s \xi_j^s \frac{\xi_j^{s^*} + 1}{2} = \frac{1}{2N}\sum_{s=1}^{n} \xi_i^s \sum_{j\neq i}^{N} \xi_j^s \xi_j^{s^*} + \frac{1}{2N}\sum_{s=1}^{n} \xi_i^s \sum_{j\neq i}^{N} \xi_j^s \tag{1.16}$$

In the second term, note that:

$$\sum_{j\neq i}^{N} \xi_j^s \approx (N-1)\langle \xi_j^s \rangle = (N-1)[p - (1-p)] = (N-1)[2p-1] \equiv (N-1)a$$

where $a \equiv 2p - 1$ for convenience of notation. Substituting back and approximating $(N-1)/N \approx 1$ leads to:

$$\frac{1}{2}\sum_{s=1}^{n} \xi_i^s \sum_{j\neq i}^{N} \xi_j^s \approx \frac{a}{2}\sum_{s=1}^{n} \xi_i^s \tag{1.17}$$

On the other hand, the first term in (1.16) can be split in two components, one with $s = s^*$, and one with $s \neq s^*$:

$$\frac{1}{2N}\sum_{s=1}^{n}\xi_i^s\sum_{j\neq i}^{N}\xi_j^s\xi_j^{s^*} = \underbrace{\frac{1}{2N}\xi_i^{s^*}\sum_{j\neq i}(\xi_j^{s^*})^2}_{(s=s^*)} + \underbrace{\frac{1}{2N}\sum_{s\neq s^*}\xi_i^s\sum_{j\neq i}^{N}\xi_j^s\xi_j^{s^*}}_{(s\neq s^*)}$$

In the first, $(\xi_j^{s^*})^2 = 1$ always, since $\xi_j^{s^*} \in \{\pm 1\}$. In the second, the correlation between different patterns can be estimated as follows:

$$\sum_{j\neq i}\xi_j^s\xi_j^{s^*} \approx (N-1)\langle\xi_j^s\xi_j^{s^*}\rangle = (N-1)\Big[1\cdot p^2 - 1\cdot p(1-p) - 1\cdot(1-p)p + 1\cdot(1-p)^2\Big] =$$
$$= (N-1)(2p-1)^2 = (N-1)a^2$$

Substituting back, and again approximating $(N-1)/N \approx 1$ leads to:

$$\frac{1}{2N}\sum_{s=1}^{n}\xi_i^s\sum_{j\neq i}^{N}\xi_j^s\xi_j^{s^*} = \frac{a^2}{2}\sum_{s\neq s^*}\xi_i^s \tag{1.18}$$

Finally, putting (1.18) and (1.17) back in (1.16) gives:

$$H_i^{s^*} \approx \frac{\xi_i^{s^*}}{2} + \frac{a^2}{2}\sum_{s\neq s^*}^{n}\xi_i^s + \frac{a}{2}\sum_{s=1}^{n}\xi_i^s$$

When $p = 1/2$, then $a = 0$, and this result reduces to the one obtained before (1.12). However, when $|a|$ is particularly high, as it is the case for biologically plausible "sparse" memories, then the stability of learned patterns is impaired.

For instance, if $\xi_i^{s^*} = +1$, then $H_i^{s^*} > 0$ is needed for the i-th neuron value to be stable. However, the two sums can be negative. Stability is then always conserved if the first term plus the minimum of the others remains positive:

$$\frac{1}{2} - \frac{a^2}{2}(n-1) - \frac{a}{2}n > 0$$

The same condition is obtained when requesting the stability of $\xi_i^{s^*} = -1$, after reversing all signs.

Solving for n leads to a bound on the maximum number of patterns that can be surely maintained at a given level of bias:

$$n < \frac{1+a^2}{a+a^2}$$

A reasonable value of sparsity is $p = .05$ (i.e. only 5% of neurons in a pattern are active on average), corresponding to $a = -.9$. Inserting it in the above expression returns $n < 1.05$: only *a single* pattern can be maintained stable, regardless of the number N of neurons!

The solution for this issue is to "subtract the bias" in the Hebbian rule. Namely:

$$\mathrm{T}_{ij} = \frac{1}{N}\sum_{s=1}^{n}(\xi_i^s - a)(\xi_j^s - a) \tag{1.19}$$

In this way, (1.16) becomes:

$$H_i^{s^*} = \frac{1}{2N}\sum_{s=1}^{n}(\xi_i^s - a)\sum_{j\neq i}^{N}(\xi_j^s - a)\xi_j^{s^*} + \frac{1}{2N}\sum_{s=1}^{n}(\xi_i^s - a)\sum_{j\neq i}^{N}(\xi_j^s - a)$$

Now:

$$\sum_{j\neq i}^{N}(\xi_j^s - a) \approx (N-1)[a-a] = 0 \qquad \sum_{j\neq i}^{N}\underbrace{(\xi_j^s - a)\xi_j^{s^*}}_{(s\neq s^*)} \approx (N-1)[a^2 - a^2] = 0$$

and the result is the same from before, even with $a \neq 0$:

$$H_i^{s^*} \approx \frac{\xi_i^{s^*} - a}{2}$$

Another possibility, taken in [2], is to instead use "V-variables" even in the Hebbian rule. This amounts to changing $\xi_i^s \to V_i^s$ and $a \to p$ in (1.19):

$$\mathrm{T}_{ij} = \frac{1}{N}\sum_{s=1}^{n}(V_i^s - p)(V_j^s - p)$$

The advantage is that this leads to better capacity overall [37].

With this choice, the input produced by the pattern s^* becomes:

$$H_i^{s^*} = \sum_{j\neq i}^{N}\mathrm{T}_{ij}V_j^{s^*} = \frac{1}{N}\sum_{j\neq i}^{N}\sum_{s=1}^{n}\tilde{V}_i^s\tilde{V}_j^s V_j^{s^*} \qquad \tilde{V}_j^s \equiv V_j^s - p$$

Again, this is computed by splitting the $s = s^*$ term from the crosstalk term ($s \neq s^*$):

$$H_i^{s^*} = \underbrace{\frac{1}{N}\tilde{V}_i^s\sum_{j\neq i}^{N}\tilde{V}_j^{s^*}V_j^{s^*}}_{(s=s^*)} + \frac{1}{N}\sum_{s\neq s^*}\tilde{V}_i^s\sum_{j\neq i}^{N}\tilde{V}_j^s V_j^{s^*}$$

For the $s = s^*$ term, note that:

$$\sum_{j\neq i}^{N}(V_j^{s^*} - p)V_j^{s^*} = \sum_{j\neq i}^{N}\Big[\underbrace{(V_j^{s^*})^2}_{V_j^{s^*}} - pV_j^{s^*}\Big] = \sum_{j\neq i}V_j^{s^*}(1-p) \approx (N-1)(1-p)\underbrace{\langle V_j^{s^*}\rangle}_{p} =$$
$$= (N-1)p(1-p)$$

While the $s \neq s^*$ terms averages to 0:

$$\sum_{s\neq s^*}^{n} \tilde{V}_i^s \sum_{j\neq i}^{N} \tilde{V}_j^s V_j^{s^*} \approx (N-1) \sum_{s\neq s^*}^{n} \tilde{V}_i^s \langle \tilde{V}_j^s V_j^{s^*} \rangle = 0$$

because the terms in the average are independent:

$$\langle \tilde{V}_j^s V_j^{s^*} \rangle = (\langle V_j^s \rangle - p)\langle V_j^{s^*} \rangle = (p-p)\cdot p = 0$$

Thus:

$$H_i^{s^*} \approx p(1-p)[V_i^{s^*} - p] \tag{1.20}$$

What about the noise in the crosstalk term? Since its average is 0, its standard deviation is just the root of the second moment:

$$\text{Var}(\text{Crosstalk}) = \Big[\frac{1}{N}\tilde{V}_i^s \sum_{j\neq i}^{N} \tilde{V}_j^s V_j^{s^*}\Big]^2 = \frac{(\tilde{V}_i^s)^2}{N^2} \sum_{j\neq i}^{N}\sum_{k\neq i}^{N} \tilde{V}_j^s \tilde{V}_k^s V_j^{s^*} V_k^{s^*}$$

Only the terms with $j = k$ have a nonzero average, as in (1.13). Thus:

$$\begin{aligned}\text{Var}(\text{Crosstalk}) &= \frac{(\tilde{V}_i^s)^2}{N^2} \sum_{j\neq i}^{N} (\tilde{V}_j^s)^2 (V_j^{s^*})^2 \approx \frac{\langle (V_i^s - p)^2 \rangle}{N^2} (N-1) \langle (V_j^s - p)^2 \rangle \langle V_j^{s^*} \rangle = \\ &= \frac{N-1}{N^2} p^3 (1-p)^2\end{aligned}$$

since:

$$\langle (V_j^s - p)^2 \rangle = \langle (V_j^s)^2 \rangle - 2p\langle V_j^s \rangle + p^2 = \langle V_j^s \rangle - 2p^2 + p^2 = p(1-p)$$

There are $n-1$ terms in the sum, meaning that the total noise is:

$$\sigma = \Big[\frac{(N-1)(n-1)}{N^2} p^3 (1-p)^2\Big]^{\frac{1}{2}} \underset{\substack{p\ll 1\\ N\gg 1}}{\approx} \sqrt{\alpha p^3} \qquad \alpha \equiv \frac{n}{N}$$

Finally, it is convenient to modify the normalization of T_{ij} to remove the prefactor in (1.20). This leads to the *final* choice of weights, which forms the basis of the Hopfield model proposed in [2]:

$$\mathrm{T}_{ij} = \frac{1}{Np(1-p)} \sum_{s=1}^{N} (V_i^s - p)(V_j^s - p) \tag{1.21}$$

1.4.3 Adding retrieval dynamics

In the previous sections, a network capable of storing and retrieving sparse patterns was presented. In summary, it consists of N binary units $V_i \in \{0, 1\}$, which are fully connected by the weights T_{ij} as specified in (1.21), and evolve according to the update rule (1.4).

However, this models the retrieval of just a single memory. When the network falls into an attractor state, it will remain there forever, and no other memory will be recalled.

To avoid this, Romani et al. [2] propose the addition of an external signal, acting as a *global inhibition*, which is used to "force" the network to move its state out of an attractor. The new update rule is defined to be:

$$V_i(t+1) = \Theta\Big(\sum_{j=1}^{N} \mathrm{T}_{ij} V_j(t) - \frac{J_0}{Np} \sum_{j=1}^{N} V_j(t) - \mathrm{th}_i \Big) \tag{1.22}$$

where J_0 denotes the amplitude of the global inhibition signal, and th_i are local thresholds for each neuron, which are sampled uniformly from the interval $[-\theta, \theta]$.

The idea is that, when $J_0 \neq 0$, the more neurons are active, the more it becomes difficult to keep them active. In other words, the global inhibition acts as a dynamic activation threshold depending on the activity of the whole network. For J_0 high, it is expected that the only stable patterns will be the ones with the fewest active neurons, and the attractors corresponding to the memories may become unstable, allowing the network to transition to other states as desired.

To see that, recall from (1.20) that the average input for the i-th neuron generated by pattern s is $V_i^s - p$ (the prefactor $p(1-p)$ is cancelled by the new normalization for T_{ij}). Then, considering that on average $\sum_{j=1}^{N} V_j \approx Np$, the update rule for a pattern becomes:

$$V_i^s(t+1) = \Theta(V_i^s - p - J_0 - \mathrm{th}_i)$$

By construction, $\mathrm{th}_i \in [-\theta, \theta]$. To keep $V_i^s = 1$ active, the argument of Θ needs to be positive, and since $\mathrm{th}_i \leq \theta$:

$$1 - p - J_0 - \mathrm{th}_i > 1 - p - J_0 - \theta > 0 \Rightarrow J_0 < 1 - p - \theta$$

Conversely, to keep $V_i^s = 0$ inactive, the argument of Θ must be negative:

$$-p - J_0 - \mathrm{th}_i < -p - J_0 + \theta < 0 \Rightarrow J_0 > \theta - p$$

Thus, the memorized patterns are stable for the following range of J_0 values:

$$\theta - p < J_0 < 1 - \theta - p \tag{1.23}$$

Pushing J_0 above the threshold $1 - \theta - p$ makes the original memories unstable. However, at such high J_0, other new states may be stable. For instance, consider the product of two patterns $\mu \neq \nu$:

$$V_i^{\mu\nu} \equiv V_i^{\mu} V_i^{\nu} \qquad \forall i = 1, \ldots, N$$

These are effectively the intersections between the active neurons of different patterns. As such, they consist of very few active units. Then, intuitively, they should be able to "survive" higher levels of J_0. There may be a range in which J_0 is too high for the original patterns to be stable, but intersections of patterns can be retrieved.

This is indeed what happens. First, note that the mixtures of two patterns are less susceptible to the effect of J_0:

$$-\frac{J_0}{Np}\sum_{j=1}^{N} V_j^\mu V_j^\nu \approx -\frac{J_0}{Np} Np^2 = -J_0 p \tag{1.24}$$

Then, consider the input of the i-th neuron when the state is $V_i^{\mu\nu}$:

$$H_i^{\mu\nu} = \sum_{j\neq i}^{N}\sum_{s=1}^{n} \frac{1}{Np(1-p)}(V_i^s - p)(V_j^s - p)V_j^\mu V_j^\nu$$

As before, all terms with $s \neq \nu, \mu$ have a zero mean. The only two with nonzero contribution are:

$$H_i^{\mu\nu} = \frac{1}{Np(1-p)}\Big[\underbrace{\sum_{j\neq i}^{N} V_j^\mu V_j^\nu (V_i^\mu - p)(V_j^\mu - p)}_{(s=\mu)} + \underbrace{\sum_{j\neq i}^{N} V_j^\mu V_j^\nu (V_i^\nu - p)(V_j^\nu - p)}_{(s=\nu)}\Big]$$

Since:

$$\frac{1}{N}\sum_{j\neq i}^{N} V_j^\mu V_j^\nu (V_j^\mu - p) \approx \langle V_j^\mu V_j^\nu (V_j^\mu - p)\rangle = \langle V_j^\nu \rangle \langle (V_j^\mu)^2 - pV_j^\mu \rangle = p^2(1-p)$$

and analogously for the term with $s = \nu$, one gets:

$$H_i^{\mu\nu} \approx p(V_i^\mu - p) + p(V_i^\nu - p) = p(V_i^\mu + V_i^\nu - 2p) \tag{1.25}$$

Substituting (1.24) and (1.25) in the update rule:

$$V_i^s(t+1) = \Theta(p(V_i^\mu + V_i^\nu - 2p - J_0) - \text{th}_i)$$

To maintain $V_i^s = V_i^\mu V_i^\nu = 1$ ($\Leftrightarrow V_i^\mu = V_i^\nu = 1$), the argument of Θ must be positive:

$$p(2-2p-J_0) - \text{th}_i > p(2-2p-J_0) - \theta > 0 \Rightarrow J_0 < 2 - 2p - \frac{\theta}{p}$$

Conversely, to maintain $V_i^s = V_i^\mu V_i^\nu = 0$, the argument of Θ must be negative. In this case, $V_i^\mu = 0 \vee V_i^\nu = 0$, but the most restricting constraint happens when exactly one of them is 0, because this would make the result higher. Thus:

$$p(1-2p-J_0) - \text{th}_i < p(1-2p-J_0) + \theta < 0 \Rightarrow J_0 > 1 - 2p + \frac{\theta}{p}$$

Thus, the mixtures of two learned patterns are stable if:

$$1 - 2p + \frac{\theta}{p} < J_0 < 2 - 2p - \frac{\theta}{p} \tag{1.26}$$

Compare this with the condition found in (1.23). Note that, if $\theta \to 0$, then $2 - 2p > 1 - p$ for $p < 1$ (i.e. always, since p is a probability), and so there exists a range in which J_0 is sufficiently high for patterns to be unstable, but so that mixtures of two patterns are still stable. Conversely, if $\theta \to 0$, then $1-2p > -p$, meaning that for sufficiently low J_0, only the original patterns are stable, and the mixtures of two patterns are unstable.

This suggests a process to artificially induce transitions between memories:

1. Start at a low J_0, with a random initial state. Wait until the network converges to a stable pattern. This is the first memory recalled.
2. Increase J_0 until the retrieved pattern is no longer stable. The higher global inhibition will cause many neurons to deactivate, and the network will then converge to the mixture between the previously retrieved memory and another random memory, since these are the closest mixtures to the starting state.
3. Decrease J_0 again, until patterns are once again stable. Since the starting state this time is in-between two memories, one of the two will likely be selected as the final stable state.

One immediate issue is that, at point 3, the network may simply return to the memory recalled at point 1. To avoid this, synapses are adapted so that it is difficult for the network to maintain the same pattern for long:

$$\mathrm{th}_i(t+1) - \mathrm{th}_i(t) = -\frac{\mathrm{th}_i(t) - \mathrm{th}_i(0)}{t_{\mathrm{th}}} + \frac{D_{\mathrm{th}} V_i(t)}{t_{\mathrm{th}}} \tag{1.27}$$

Here, the change in each threshold is the sum of two terms [2]:

- A "pullback" term, proportional to $\mathrm{th}_i(t) - \mathrm{th}_i(0)$, that brings each threshold back to its starting value.
- An adaptation term, proportional to $V_i(t)$, that rises the threshold for active neurons.

The two parameters are D_{th}, regulating the amplitude of the adaptation, and t_{th} for its timescale.

With this modification, when at point 3 neurons begin reactivating as J_0 is lowered, the ones that were not active in the previously recalled state will have a lower threshold, and so they will activate earlier. With the correct choice of D_{th} and t_{th}, this makes the network transition to a new pattern, retrieving a new memory.

This completes the theoretical setup of the neural recall model presented in [2]. While global inhibition and synaptic adaptation may be biologically plausible, there is no strong experimental evidence for such mechanisms. Moreover, the model so found depends on several free parameters. However, as it will be shown in the following sections, by simulating the network, and "distilling" some key features, it is possible to construct a simpler model which captures some fundamental characteristics of retrieval in human memory.

1.4.4 Simulations

The model just presented can be numerically simulated to inspect its characteristics.

Since the focus is on the mechanics of retrieval, only $n \ll N$ memories are chosen, so that they can all be stored without error inside the network. The idea is that, in principle, it would be possible to retrieve any of them by using the correct *cue*. However, the recall dynamics always use the previously recalled item as a *cue* for retrieving new memories. This, as it will be shown, results in fewer memories that are actually recalled.

In practice, the parameters used for the simulation are taken from [2, sec. 2.1], namely:

- The number of neurons is set to $N = 3000$. In general, N must be as high as possible. However, building the matrix of synapses requires $O(N^2)$ memory, so N cannot be too high for performance reasons.

- The number n of patterns is set to $n = 16$. This ensures the possibility to perfectly retrieve all of them with no error given that $N = 3000$.

- The sparseness p, i.e. the probability that a given entry of a pattern is $+1$, is set to $p = 0.1$.

  A posteriori, the next section will show that memory retrieval can be best modelled with $p \to 0$. Additionally, at this limit the Hopfield model exhibits increased capacity [37] [38].

  However, numerically p cannot be too small: during the transition between patterns, the network's state will stabilize on the mixture of two patterns, with an average number of active units of Np^2. Thus, to avoid the network completely "shutting down" at each transition, $Np^2 \gg 1$ by at least $1 \div 2$ orders of magnitude.

- The local threshold U_i of each neuron is uniformly sampled from $[-\theta, \theta]$, with $\theta = .015 \ll p$ (otherwise at $J_0 = 0$ the memory patterns would be unstable).

- The global inhibition J_0 varies sinusoidally between $J_0^{\text{min}} = 0.7$ and $J_0^{\text{max}} = 1.2$, with a period of 50 timesteps. The only important constraint is that $1 - \theta - p < J_0^{\text{max}} < 2 - 2p - \theta/p$, i.e. $.885 < J_0 < 1.65$ in this case. In this way, according to (1.23) and (1.26), at J_0^{max} the patterns are unstable, but the mixtures of two patterns are stable.

  Note that $J_0^{\text{min}} \approx 1 - 2p + \theta/p = .65$, i.e. most of the mixtures are unstable when $J_0 = J_0^{\text{min}}$, and so the model correctly converges to one of the original memories.

- Finally, the timescale of adaptation t_{th} and the strength of adaptation D_{th} are tuned to avoid the network returning to patterns just visited, i.e. to minimize the occurrence of loops of just two items in the dynamics. The chosen values are $t_{\text{th}} = 45$, which is slightly lower than the J_0 period, and $D_{\text{th}} = 3\theta$.

  If D_{th} is too low, the network will cycle between two patterns. If it is too high, patterns won't be maintained even when J_0 is sufficiently low, because the thresholds rise too rapidly. In [2], $D_{\text{th}} = 1.9\theta$ is used instead, but the higher value used here is found to perform better.

To see which pattern the network is retrieving, one computes the **overlaps** m^s of the network's state $\boldsymbol{V}(t)$ with each of the original stored patterns $\{\boldsymbol{V}^s\}_{s=1,\dots,n}$:

$$m^s(t) \equiv \frac{1}{Np(1-p)} \sum_{i=1}^{N} (V_i^s - p) V_i(t) \qquad s = 1, \dots, n$$

When $m^s(t) = 0$, the network's state is completely uncorrelated with the s-th pattern, while $m^s(t) = 1$ means that the s-th pattern has been perfectly retrieved.

A plot of the evolution of the overlaps during a simulation is shown in fig. 1.3. At the start, the network is set to the first pattern[1].

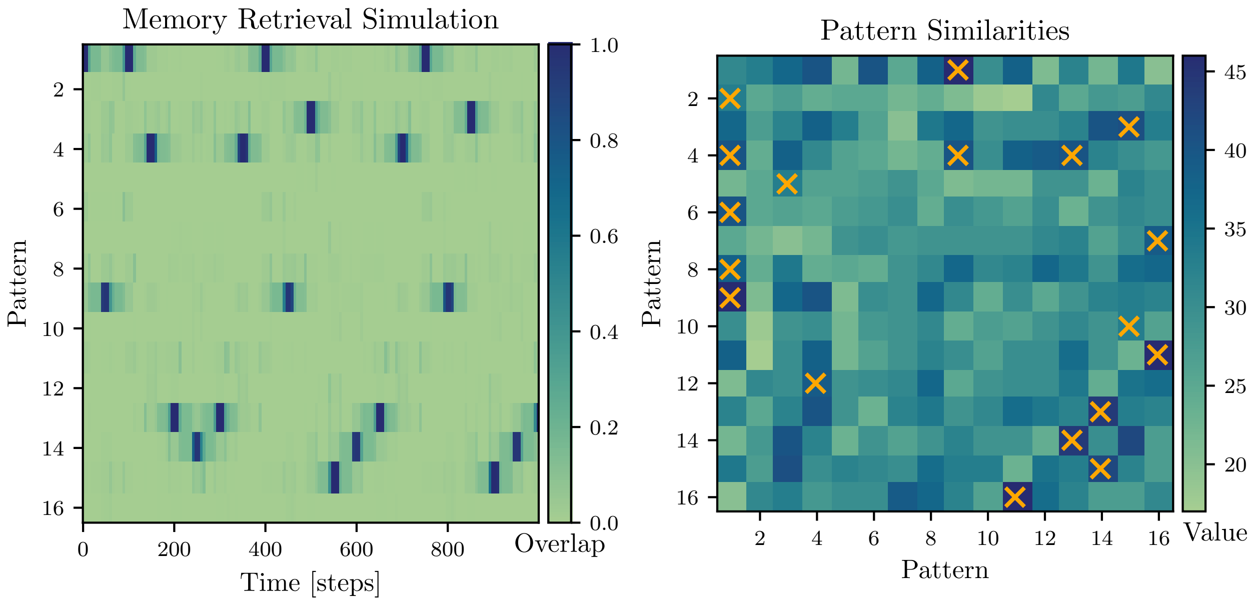


Figure 1.3 – Left: Evolution of the overlap between the network's state and each of the stored patterns. Parameters used for the simulation are: $N = 3000$, $n = 16$, $p = .1$, $\theta = .015$, $J_0^{\text{min}} = .7$, $J_0^{\text{max}} = 1.2$, $t_{\text{th}} = 45$ and $D_{\text{th}} = .045$. **Right**: Similarity matrix S between the original patterns, measured as their overlaps. The orange crosses denote the highest value(s) of each row. Entries on the diagonal have been filled with the average of the other $n - 1$ terms, to aid the visualization.

There are several things of note:

- Even if all patterns are correctly stored in the network, and can in principle be retrieved, they are not all recalled during the dynamics. In particular, after a brief transient, the network enters a loop, in which the transitions $14 \to 13 \to 4 \to 1 \to 9 \to 3 \to 15 \to 14$ are constantly repeated, and no more new items are recalled. In this specific simulation, only 7 memories out of 16 are retrieved (items 2, 5, 6, 7, 8, 10, 11, 12 and 16 are never retrieved). This is a general behavior, not specific to just this instance of simulation.
- During a transition between items, the network converges as expected to a mixture of the two items (note the values $\sim .2$ surrounding the retrieval states with overlap ~ 1). One of these two is clearly the one that was just recalled. The other, however, is not random: transitions tend to happen between patterns that are "more similar".

  To see that, consider the similarity matrix S, with entries S_{ij} equal to the overlaps between couples (i, j) of patterns:

$$S_{ij} \equiv \sum_{k=1}^{N} V_k^i V_k^j \tag{1.28}$$

[1] ∧Since patterns are randomly generated at the start of the simulation, this is effectively equivalent to starting at a random pattern.

This matrix is shown in the right part of fig. 1.3. Note how most of the transitions happen towards the item with the largest overlap.

For instance, at the beginning $1 \to 9$ is selected, and in fact 1 has the highest overlap with 9 (it is the highest entry of the first row in the similarity matrix). This is followed by $9 \to 1$, since also 1 has the largest overlap with 9 out of all patterns[2].

After that, something interesting happens: instead of $1 \to 9$ repeating, $1 \to 4$ happens instead. This is an effect of synaptic adaptation: 9 has been recently active, and so its thresholds are higher than normal. Note also that 1 has a very high overlap with 4 too (but not as large as with 9).

Thus, sometimes items with lower overlaps are chosen for a transition, because the transition towards the item with the highest overlap is precluded by synaptic adaptation.

To see if this pattern is indeed general, fig. 1.4 shows a histogram of the "ranks" of each transition, over 100 simulations (with the same parameters as in fig. 1.3). If a transition $a \to b$ happens towards the item b with the maximum similarity with a, then it has rank 1. A transition towards an item c with the second-highest overlap has rank 2, and so on. As expected, over 90% of the transitions happen towards the items with the two largest overlaps.

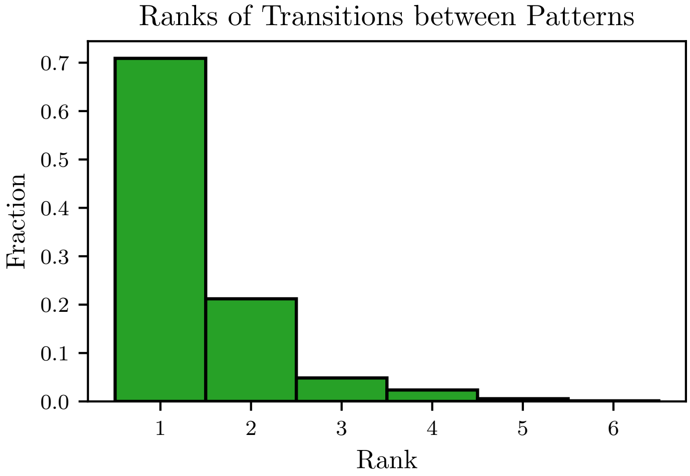


Figure 1.4 – Histogram of the ranks of transitions between memories in the neural network model of retrieval, over 100 simulations. For a transition $a \to b$, consider the overlap S_{ab} between a and b. Then, the rank of such transition is the number of distinct overlaps $\{S_{ab}\}_{k=1,\ldots,n}$ which are greater or equal than S_{ab}. Thus, if a has its highest overlap with b (i.e. the entry S_{ab} is the highest of row a), the rank of $a \to b$ is 1. Ties are assigned the same rank.

In summary, this model allows to predict the number (and sequence) of items recalled given the number of items n stored in memory. Retrieval begins with a random memory, and each recalled item serves as *cue* for recalling new items. Eventually, this leads to a loop, and no new items are retrieved, thus terminating the search.

This model, however, still requires many free parameters (p, θ, J_0^{min}, J_0^{max}, t_{th} and D_{th}), and is difficult to characterize analytically. So, it is worthwhile to "distill" it, extracting only the relevant dynamics to form a simpler model, which can be thoroughly studied and used to form predictions. This will be the subject of the following section.

[2] ∧This is not necessarily true for all couples of patterns. For instance, 2 has the largest overlap with 1, but the converse is not true. However, it happens quite often: notice how the largest entries in each row, denoted by the orange × in fig. 1.3, are mostly placed in symmetrical positions.

1.5 Graph model

As shown in fig. 1.4, during retrieval, transitions happen between memories that are strongly overlapped.

Thus, a model of retrieval can be built by starting with a similarity matrix S, where S_{ij} represents how similar pattern i is to j [2].

For now, the effect of synaptic adaptation is neglected for simplicity. Denote with $\text{rec}(t)$ the memory recalled at time step t. At the start, let $\text{rec}(t) = i$, where i is chosen at random between the n available memories. At $t+1$, the new recalled item $\text{rec}(t+1)$ is defined to be the one with the highest similarity with the previous one:

$$\text{rec}(t+1) = \underset{j \neq \text{rec}(t)}{\arg\max}\, S_{\text{rec(t)},j} \tag{1.29}$$

For a fixed matrix S, this model is deterministic. In particular, if an item is visited for a second time, all the following transitions will repeat as before, entering a loop, and no new items will be recalled.

Thus, the recall performance corresponds to the number of items visited before entering a loop.

1.5.1 Asymmetric case

For a quick computation, suppose that the entries of S_{ij} are all independent and identically distributed random variables — which cannot be true in general, since it would result in an asymmetric matrix, while a matrix of similarities must be symmetric. However, if all entries S_{ij} are i.i.d., then all transitions from an item visited just once have the same probability of happening.

Consider any path:

$$\alpha_1 \to \alpha_2 \cdots \to \alpha_k \to i$$

with all $\{\alpha_1, \dots, \alpha_k, i\}$ distinct. The probability for a transition from i to any other element $j \neq i$ is given by:

$$\mathbb{P}[i \to j] \equiv p_0 = \frac{1}{n-1}$$

Starting from any item i, there are a total of $n-1$ other items that can be visited. Since all entries in S are i.i.d., the previous transitions tell nothing about what is the maximum in $\{S_{ik}\}_{k=1,\dots,n;k\neq i}$, because i was never visited before, and so this row is being "accessed" now for the first time.

Then, the probability that the item in position j is the maximum between $n-1$ items is $(n-1)^{-1}$, whatever the items[3].

[3] ∧As long as the value of each item does not depend on its position in the i-th row of S, i.e. if the entries of the i-th row are exchangeable random variables. This clearly holds if they are i.i.d. random variables. In other

Suppose a total of m items have been visited, one of which is i. Then, since transitions are mutually exclusive events, the probability that the item recalled after i will be one that has already been visited is $(m-1)p_0$ (because i is excluded from the possible transitions).

Thus, consider now a generic sequence. Two items are always visited:

$$\alpha_1 \to \alpha_2$$

If $\alpha_2 \to \alpha_1$, which happens with probability p_0, a loop is entered. Otherwise, with probability $1-p_0$, $\alpha_2 \to \alpha_3$ happens instead, with $\alpha_3 \neq \alpha_1$. After that, a loop is entered if $\alpha_3 \to \alpha_1$ or $\alpha_3 \to \alpha_2$ (probability $2p_0$), and so on. Thus, the probability that a loop is entered after exactly k items have been retrieved is:

$$P_{\text{loop}}(k;n) = \underbrace{(1-p_0)(1-2p_0)(1-3p_0)\cdots(1-(k-3)p_0)}_{\text{Do not enter loops before } k \text{ visited items}} \underbrace{(k-2)p_0}_{\substack{\text{Enter a loop after} \\ \text{visiting the } k\text{-th item}}}$$

with $P_{\text{loop}}(1;n) = 0$. This can be rewritten as:

$$\begin{aligned} P_{\text{loop}}(k;n) &= (k-2)p_0 \prod_{\alpha=1}^{k-3}(1-\alpha p_0) \underset{p_0 \ll 1}{\approx} (k-2)p_0 \prod_{\alpha=1}^{k-3} e^{-\alpha p_0} = (k-2)p_0 \exp\Big(-p_0 \sum_{\alpha=1}^{k-3} \alpha\Big) = \\ &= (k-2)p_0 \exp\Big(-\frac{p_0}{2}(k-3)(k-2)\Big) \underset{\substack{1 \ll k \ll n \\ p_0 \approx 1/n}}{\approx} \frac{k}{n} \exp\Big(-\frac{k^2}{2n}\Big) \end{aligned} \tag{1.30}$$

Thus, the average number of recalled items is predicted to be:

$$\langle k \rangle \approx \sqrt{\frac{\pi}{2} n} \tag{1.31}$$

which exhibits a power law dependence on n with exponent $1/2$, consistent with the results from free recall experiments (see sec. 1.1.4). However, note that here n is not the number of items that are *presented*, but the average number of items that are already *stored* in memory, and that are (in principle) reachable during the retrieval search.

1.5.2 Symmetric case

While the quick computation from the previous section suggests that the model is proceeding on the right path, it makes some strong and unnecessary simplifications.

Thus, consider the following two changes:

- First, the similarity matrix S is now assumed to be symmetric ($S_{ij} = S_{ji}$), as it should be. However, besides the correlations given by the symmetry constraint, no other correlations are present.

  In other words, S is an $n \times n$ matrix constructed from the realization of $n(n-1)/2$ i.i.d. random variables filling the upper part of the matrix, with the other entries being

words, the entries of the i-th row may as well be generated from some complex joint distribution, as long as they are shuffled thereafter.

fixed by symmetry (the diagonal is irrelevant, since in any case self-transitions will be prohibited).

This is equivalent to the case of a matrix of overlaps (1.28) in the limit of $p \to 0$ ("infinitely sparse" patterns). Recall that the matrix of overlaps is defined as:

$$\mathrm{S}_{\mu\nu} = \sum_{k=1}^{N} V_k^{\mu} V_k^{\nu}$$

Entries involving different patterns altogether, such as $\mathrm{S}_{\mu\nu}$ and $\mathrm{S}_{\sigma\delta}$ with $\mu \neq \sigma, \delta$ and $\nu \neq \sigma, \delta$, are uncorrelated.

However, if a pattern is shared between two entries (as in $\mathrm{S}_{\nu\mu}$ and $\mathrm{S}_{\mu\delta}$) correlations appear. By using the symmetry condition, it is always possible to lead this case back to entries belonging to the same row, that of the shared pattern ($\mathrm{S}_{\nu\mu} = \mathrm{S}_{\mu\nu}$, which is on the same row of $\mathrm{S}_{\mu\delta}$).

Then, the Pearson's coefficient of correlation between two distinct nondiagonal entries $\mathrm{S}_{\mu\nu}$ and $\mathrm{S}_{\mu\delta}$ on the same row μ ($\nu \neq \delta$, $\mu \neq \nu$, $\mu \neq \delta$) is:

$$\rho(\mathrm{S}_{\mu\nu}, \mathrm{S}_{\mu\delta}) = \frac{\langle \mathrm{S}_{\mu\nu}\mathrm{S}_{\mu\delta}\rangle - \langle \mathrm{S}_{\mu\nu}\rangle\langle \mathrm{S}_{\mu\delta}\rangle}{\sqrt{\mathrm{Var}(\mathrm{S}_{\mu\nu})\mathrm{Var}(\mathrm{S}_{\mu\delta})}}$$

And after some computations:

$$\langle S_{\mu\nu}\rangle = \sum_{i=1}^{N} \langle V_i^{\mu} V_i^{\nu}\rangle = Np^2$$

$$\langle S_{\mu\nu}^2\rangle = \sum_{i,j=1}^{N} V_i^{\mu} V_i^{\nu} V_j^{\mu} V_j^{\nu} = \underbrace{\sum_{i=1}^{N} (V_i^{\mu})^2 (V_i^{\nu})^2}_{(j=i)} + \sum_{i=1}^{N} \sum_{j \neq i}^{N} V_i^{\mu} V_i^{\nu} V_j^{\mu} V_j^{\nu} = Np^2 + N(N-1)p^4$$

$$\mathrm{Var}(\mathrm{S}_{\mu\nu}) = \langle \mathrm{S}_{\mu\nu}^2\rangle - \langle \mathrm{S}_{\mu\nu}\rangle^2 = (Np^2 + \cancel{N^2 p^4} - Np^4) - \cancel{N^2 p^4} = Np^2(1-p^2)$$

$$\langle \mathrm{S}_{\mu\nu}\mathrm{S}_{\mu\delta}\rangle = \sum_{i,j=1}^{N} V_i^{\mu} V_i^{\nu} V_j^{\mu} V_j^{\delta} = \underbrace{\sum_{i=1}^{N} (V_i^{\mu})^2 V_i^{\nu} V_i^{\delta}}_{(i=j)} + \sum_{i=1}^{N} \sum_{j \neq i}^{N} V_i^{\mu} V_i^{\nu} V_j^{\mu} V_j^{\delta} = Np^3 + N(N-1)p^4$$

$$\rho(\mathrm{S}_{\mu\nu}, \mathrm{S}_{\mu\delta}) = \frac{Np^3 + \cancel{N^2 p^4} - Np^4 - \cancel{N^2 p^4}}{Np^2(1-p^2)} = \frac{Np^3(1-p)}{Np^2(1+p)(1-p)} = \frac{p}{1+p}$$

And so $\rho(\mathrm{S}_{\mu\nu}, \mathrm{S}_{\mu\delta}) \to 0$ when $p \to 0$, and the only correlations left are those due to symmetry.

- Synaptic adaptation is added by prohibiting transitions to the most recently visited item. In other words, any transition cannot be followed by its reversed counterpart: after $i \to j$, $j \to i$ cannot happen. The rule from (1.7) is then modified to:

$$\mathrm{rec}(t+1) = \underset{\substack{j \neq \mathrm{rec}(t) \\ j \neq \mathrm{rec}(t-1)}}{\arg\max}\, \mathrm{S}_{\mathrm{rec}(t),j} \tag{1.32}$$

Note that this is nonetheless a simplification with respect to the neural network model, where sometimes (but rarely) transitions happen even towards the item with the third

> maximum overlap, or even lower (fig. 1.4), i.e. the network avoids returning even to $\text{rec}(t-2)$ (and so $i \to j \to k$ is less likely to be followed by $k \to i$). However, as it will be shown, the rule (1.32) suffices to obtain rich dynamics and is sufficiently simple to be studied analytically.

An example of applying the new update rule (1.32) to $n = 10$ items is shown in fig. 1.5.

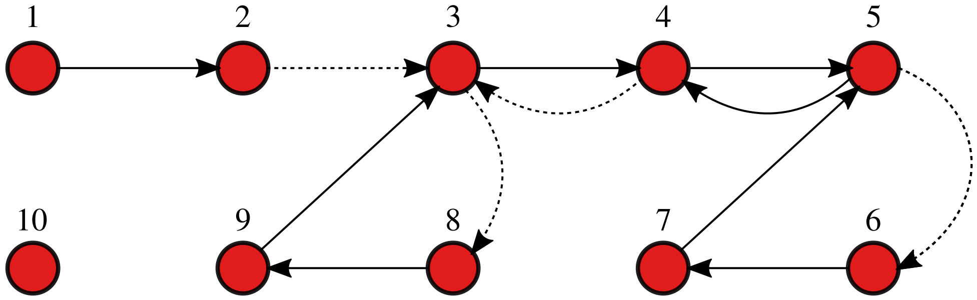


Figure 1.5 – Example of dynamics following (1.32). The transitions are: $1 \to 2 \to (3 \to 4 \to 5 \to 6 \to 7 \to 5 \to 4 \to 3 \to 8 \to 9 \to 3)$. The loop is denoted with the brackets. Transitions to items with the maximum similarity are shown as **solid** arrows ("strong" links, e.g. $1 \to 2$). If the item with the highest similarity has been visited in the previous step, a transition to the item with the second-highest similarity occurs instead (**dashed** arrows, representing "weak" links, e.g. $3 \to 8$). So, for instance, after $4 \to 3$, there cannot be $3 \to 4$ again, and so $3 \to 8$ is chosen instead.

Interestingly, a loop is not always entered when an item is visited for the second time. For instance, consider the transition $7 \to 5$ in fig. 1.5. Previously, when 5 was visited for the first time, $5 \to 6$ was chosen. However, this is not the transition that follows the highest overlap: that would be $5 \to 4$. Yet, at that time, $5 \to 4$ could not have been taken, because it would have followed $4 \to 5$, and reverse transitions cannot happen consecutively.

The second time 5 is visited, this constraint holds no more, and $5 \to 4$ is chosen. This offers a chance to "escape the loop": the next transition cannot be $4 \to 5$, and could point to a new item. This indeed happens after a bit: $5 \to 4 \to 3 \to 8$, with 8 being a new memory.

Such a situation occurs whenever two items a and b are both "the most similar to each other", i.e. $\mathsf{S}_{ab} = \mathsf{S}_{ba}$ is maximum in both row a and row b of S. So, if $a \to b \to j$, the link between $a \to b$ is "strong", i.e. it is of maximum similarity, and will be traversed again if a is revisited. The link $b \to k$ is however "weak": it is of second-highest similarity, and if b is revisited, $b \to a$ will be taken instead.

Interestingly, this setup happens quite often, due to the correlations given by S being symmetric. If S_{ab} is the maximum in row a, then $\mathsf{S}_{ba} = \mathsf{S}_{ab}$ is more likely to be the maximum also in row b.

To quantify the probability of this happening, consider that for any two rows a, b of S, there are exactly $2n-1$ distinct i.i.d. random variables, since two of the $2n$ entries are equal by symmetry. Consider them as part of a unique list $\{x_i\}_{i=1,\dots,2n-1}$, with x_1 being $\mathsf{S}_{ab} = \mathsf{S}_{ba}$, the following $n-1$ elements being the remaining entries of row a, and the final $n-1$ elements being the entries of row b (except S_{ba}, which has already been included):

$$\boldsymbol{x} = (\mathsf{S}_{ab}, \{S_{ak} \colon k \neq a\}, \{S_{bk} \colon k \neq a\}) \in \mathbb{R}^{2n-1}$$

Then:

$$
\begin{aligned}
&\mathbb{P}[\mathrm{S}_{ab} = \max_k \mathrm{S}_{bk} | \mathrm{S}_{ab} = \max_k \mathrm{S}_{ak}] = \\
&= \mathbb{P}[x_1 > \max\{x_i \colon i = n+1, \dots, 2n-1\} | x_1 = \max\{x_i \colon i = 1, \dots, n\}] = \\
&= \frac{\mathbb{P}[x_1 = \max\{x_i \colon i = 1, \dots, 2n-1\}]}{\mathbb{P}[x_1 = \max\{x_i \colon i = 1, \dots, n\}]} = \frac{1/(2n-1)}{1/n} = \frac{n}{2n-1} \underset{n \gg 1}{\approx} \frac{1}{2}
\end{aligned}
\tag{1.33}
$$

This also the unconditioned probability of a specific link being “strong” or “weak”.

In summary, a loop is not surely entered just if an item is visited two times. To enter a loop, a transition must be exactly repeated.

For instance, in the example from fig. 1.5, the loop is not entered when 5, 4 and 3 are visited the second time. Instead, the loop begins when the transition $3 \to 4$ is repeated following the same order. This makes the context equal to that when $3 \to 4$ was first chosen, meaning that all the following items are repeating an already known sequence.

Let the probability that any transition enters a loop be p^*. Following the same reasoning from (1.30), the probability of entering a loop after k distinct elements have been visited is given by:

$$
P_{\text{loop}}(k; n) \approx k p^* \exp\Big(-\frac{p^* k^2}{2}\Big) \tag{1.34}
$$

The probability p^* is estimated as:

$$
p^* \approx p_0(1 - p_1) \tag{1.35}
$$

where:

- p_0 is the probability of revisiting an “old” item for the second time.
- $1 - p_1$ is the probability of *immediately* entering a loop after revisiting the old item. That is, the probability that the transition chosen after reaching again the “old” item was already made.

Note that this is just an approximation. First, some effects that are negligible for longer lists are being ignored. For instance, if a transition happens towards the *first* visited item, then the following transition will be repeated for sure, not with probability $1 - p_1$. However, this becomes more unlikely as n increases, and so it can be neglected.

Another issue is the adopted definition of $1 - p_1$. To get an exact value for p^*, $1 - p_1$ should be replaced with the probability $(1 - p_1')$ that, after the “old” item is revisited, a new item *will* be visited (i.e. “truly escaping a potential loop”). This is approximated by the probability of not entering the loop immediately (“escaping the first threat of a potential loop”), since the majority of the resulting paths will go on to explore a new item.

The difference is subtle and is best illustrated by considering again the example in fig. 1.5. p_0 is the probability of the transition $7 \to 5$, which goes back to an “old” item. Then, p_1 is the probability of *not* taking $5 \to 6$ again after that, while p_1' is the probability that, after revisiting 5, a new item will be visited (e.g. 8). p_1' is slightly lower than p_1, because not all paths that avoid entering an "immediate" loop go on to explore new items. In the case of fig. 1.5, this was the case, since eventually one reaches 3 → 8, and 8 is a new item. However, suppose that 3 → 2 happened instead, followed by 2 → 4. The next transition, i.e. 4 → 5, would be repeated, meaning that a loop has been entered. By approximating $p^* \approx p_0(1-p_1)$, these situations are simply neglected. Note, however, that again they should become rarer as n increases, simply because there are more "new" items that can be explored.

Having made these remarks, one can proceed in estimating both p_0 and p_1.

Transitions to old items (p_0)

Consider a generic path (fig. 1.6):

$$(\alpha_1 \to \cdots \to \alpha_s) \to i \to \underbrace{(\beta_1 \to \cdots \to \beta_t)}_{t\geq 1} \to k \tag{1.36}$$

where all items are distinct. The objective is to compute the probability p_0 that the next transition will be $k \to n$, i.e. towards a *specific* "old" item.

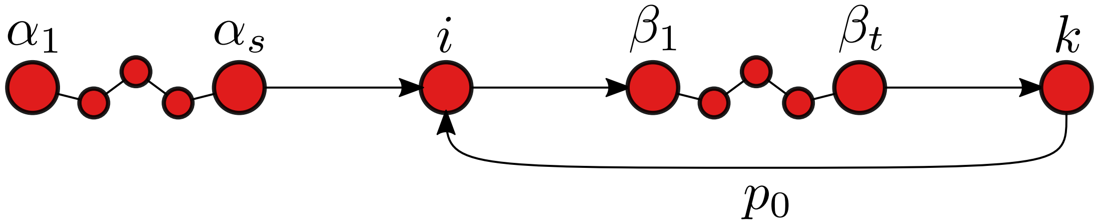


Figure 1.6 – Graphical representation of the path (1.36).

Since S is symmetric, its entries are correlated, and past transitions tell something about the likelihood of transitions that have yet to happen.

In this case, since all items are distinct, the transition $i \to k$ has *not* happened. This means that S_{ik} is surely not the maximum[4] in the i-th row of S.

Then, denoting the a-th row of S as the vector $\boldsymbol{S_a}$, p_0 can be estimated as follows:

$$\begin{aligned} p_0 = \mathbb{P}(\mathrm{S}_{ki} = \max \boldsymbol{S_k} | \mathrm{S}_{ki} < \max \boldsymbol{S_i}) &= \frac{\mathbb{P}[\mathrm{S}_{ki} = \max \boldsymbol{S_k}, \mathrm{S}_{ki} < \max \boldsymbol{S_n}]}{\mathbb{P}[\mathrm{S}_{ki} < \max \boldsymbol{S_n}]} = \\ = \frac{\mathbb{P}[\mathrm{S}_{ki} = \max \boldsymbol{S_k}]\mathbb{P}[\mathrm{S}_{ki} < \max \boldsymbol{S_i} | \mathrm{S}_{ki} = \max \boldsymbol{S_k}]}{\mathbb{P}[\mathrm{S}_{ki} \neq \max \boldsymbol{S_i}]} &= \\ = \frac{\mathbb{P}[\mathrm{S}_{ki} = \max \boldsymbol{S_k}]\mathbb{P}[\max \boldsymbol{S_k} < \max \boldsymbol{S_i}]}{\mathbb{P}[\mathrm{S}_{ki} \neq \max \boldsymbol{S_i}]} \approx \frac{(1/n)(1/2)}{1-1/n} &\underset{n\gg 1}{\approx} \frac{1}{2n} \end{aligned} \tag{1.37}$$

Note that $i \to k$ could happen even as a "weak" link, which requires only $\mathrm{S}_{ki} =$ second max $\boldsymbol{S_k}$. However, for $n \gg 1$, this leads to the same result as above.

[4] ∧If $i \to \beta_1$ was a "weak" link, which happens with $p \approx 1/2$, as shown in (1.33), then also $\mathrm{S}_{ik} <$ second $\max_j \mathrm{S}_{ij}$. However, this does not add much more information, since both the max and the second max are expected to be very close, especially for $n \gg 1$. So, this fact is neglected for simplicity.

Transitions escaping loops (p_1)

Now suppose that $k \to i$ has happened, i.e. the previous sequence (1.36) now looks like (fig. 1.7):

$$(\alpha_1 \to \cdots \to \alpha_s) \to i \to \underbrace{(\beta_1 \to \cdots \to \beta_t)}_{t\geq 1} \to k \to i \tag{1.38}$$

There are only two candidates for the next transition:

- If $i \to \beta_1$ was a "strong" link, then this transition will happen again upon revisiting i. This immediately enters a loop.
- If $i \to \beta_1$ was instead a "weak" link, then revisiting i will lead to a "backward transition" $i \to \alpha_s$ instead. The loop is avoided (for now).

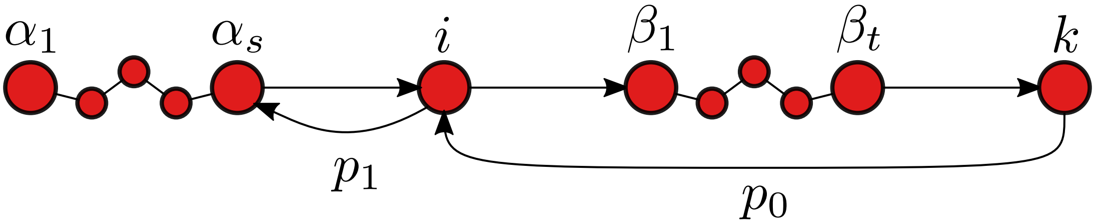


Figure 1.7 – Graphical representation of the path (1.38).

Since p_1 is the probability of *not* entering the loop immediately, it is equal to that of $i \to \beta_1$ being a "weak" link, i.e. $\mathrm{S}_{i\alpha_s} = \max \boldsymbol{S}_i$ and $\mathrm{S}_{i\beta_1} = \text{second}\max \boldsymbol{S}_i$.

Normally, this would be $1/2$, following (1.33). However, in this case, there are three transitions involving i that have happened, which give some more information about p_1.

Specifically, $\alpha_s \to i$, $i \to \beta_1$ and $k \to i$ have happened. This means that, according to (1.32):

$$\begin{aligned} \mathrm{S}_{\alpha_s i} &= \max_{\mu \neq \alpha_{s-1}} \mathrm{S}_{\alpha_s \mu} \\ \mathrm{S}_{i\beta_1} &= \max_{\mu \neq \alpha_s} \mathrm{S}_{i\mu} \\ \mathrm{S}_{ki} &= \max_{\mu \neq \beta_t} \mathrm{S}_{k\mu} \end{aligned} \tag{1.39}$$

where the diagonal S_{ii} is set to some value ≤ 0 so that it can be neglected in the maxima.

Then, $\mathrm{S}_{\alpha_s i} = \mathrm{S}_{i\alpha_s}$, and $\mathrm{S}_{ki} = \mathrm{S}_{ik}$ by symmetry. So, these three values are all on the same i-th row, and they are informative about the next transition from i. Note also that, since $i \to \beta_1$ has happened, but not $i \to k$, then $\mathrm{S}_{i\beta_1} > \mathrm{S}_{ik}$.

Finally, $i \to \beta_1$ is a "weak" link if and only if $\mathrm{S}_{i\alpha_s} > \mathrm{S}_{i\beta_1}$. In that case, there is a "strong" link between $i \to \alpha_s$, which could not have been taken when i was first visited, because α_s had been the very previous element.

Thus, p_1 is given by:

$$p_1 = \mathbb{P}[\underbrace{\mathrm{S}_{i\alpha_s}}_{A} > \underbrace{\mathrm{S}_{i\beta_1}}_{B} \mid \underbrace{\mathrm{S}_{i\beta_1}}_{B} > \underbrace{\mathrm{S}_{ik}}_{C}, \text{and all (1.39)}]$$

Since $S_{i\alpha_s}$, $S_{i\beta_1}$ and S_{ik} are maxima of different rows of S, with at most one element in common between each of them (due to symmetry), for $n \gg 1$, their $3! = 6$ orderings have all equal probability. However, there are only 3 of them that satisfy the condition $B > C$: $A > B > C$, $B > A > C$ and $B > C > A$. Of these, only one is accepted, meaning that:

$$p_1 \underset{n \gg 1}{\approx} \frac{1}{3} \tag{1.40}$$

Number of retrieved items

Finally, the estimates for p_0 and p_1 found in (1.37) and (1.40) lead to the following value for p^* (1.35):

$$p^* \approx p_0(1 - p_1) \approx \frac{1}{2n}\frac{2}{3} = \frac{1}{3n}$$

Substituting that in (1.34) gives:

$$P_{\text{loop}}(k; n) \approx \frac{k}{3n} \exp\Big(-\frac{k^2}{6n}\Big)$$

and the average number of retrieved items from a storage of size n is:

$$\langle k \rangle = \sqrt{\frac{3}{2}\pi n} \tag{1.41}$$

In summary, the above result follows from the analysis of a simplified version of the neural network model from sec. 1.4, where stored patterns are “infinitely sparse” ($p \to 0$) and many ($n \gg 1$), where transitions always happen towards the item sharing the highest similarity with the previous one, and where synaptic adaptation precludes transitions of the type $a \to b \to a$.

When S is instead constructed from overlaps of patterns with finite sparsity p, then the resulting $\langle k \rangle$ obtained by averaging many simulations is significantly lower than the estimate for $p \to 0$ from (1.41), as shown in fig. 1.8.

On the other hand, simulations with a symmetric S constructed from i.i.d. random variables are consistent with the result from (1.41).

Remarkably, this model does not require any free parameters. The only issue is that it needs as input the number n of memories that are already stored in memory, and that can (in principle) be retrieved. As it will be shown in the next section, this can be estimated by a test of recognition memory.

1.5.3 Experimental evidence

Naim et al. have experimentally tested the predictions of (1.41) through the following setup [1], using the online platform of Amazon Mechanical Turk®:

- Each participant was shown a list of length $L \in \{8, 16, 32, 64, 128, 256, 512\}$ constructed from distinct random English words with frequency per million greater than 10 [1, Suppl. Mat.].

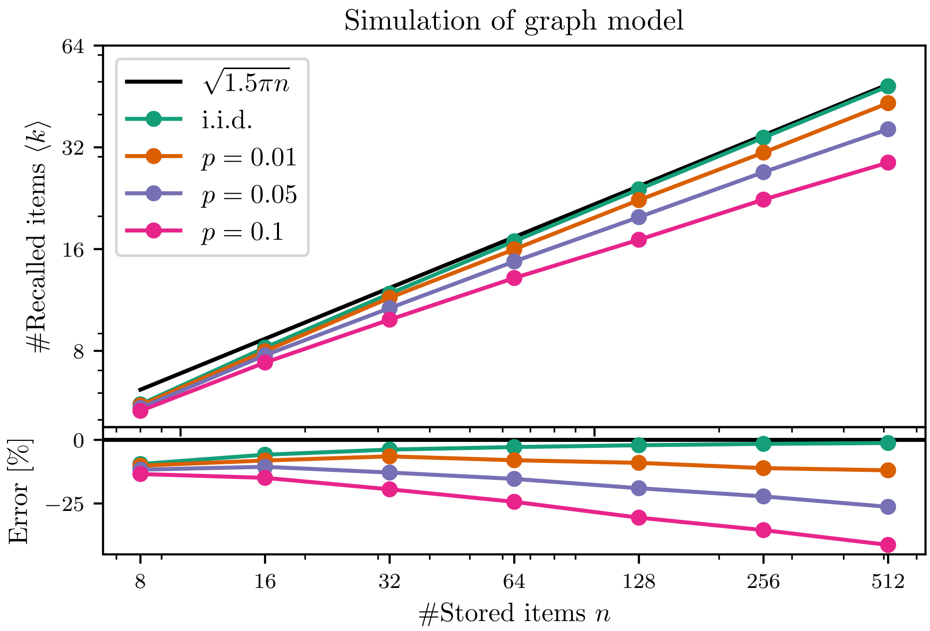


Figure 1.8 – Mean number of retrieved items obtained by simulating (1.32) and averaging over 10 000 samples (error bars are negligible, and are not shown). The $n \times n$ matrix S is constructed from the overlaps of binary biased patterns of finite sparsity p for the orange, purple and pink curves. For the green curve, $n(n-1)/2$ values are sampled uniformly from $[0, 1)$, and are then used to construct a symmetric $n \times n$ matrix. So, in this last case, the only correlations are those given by symmetry. Finally, the black curve reports the prediction from (1.41), which is valid for $n \gg 1$ and $p \to 0$.

- Then each participant was asked to recall words from the presented list in any order, and type them in a text field. Only the word being written was visible on screen.

  In the following analysis, obvious errors were corrected, while intrusions (i.e. words that were not present in the shown list) were ignored. At the end, the number of recalled items R averaged across participants was computed for each list length.

- After one week, the same participants were shown another list of the same length L, but containing different words, at the same rate of presentation from before.

- Then, they performed a total of 5 forced-choice recognition trials. In each trial, two words were shown on screen: one was from the presented list, the other was unrelated. Participants were asked to always choose the word they “recognized” as being in the shown list.

  During the analysis, only the first trial for each participant was considered. Then, the fraction of correct answers c across all participants was computed, and from that, the average number of items stored in memory M was estimated following (1.1):

$$M = L \cdot (2c - 1)$$

Naim et al. observed that, while each graph of M and R over the list lengths L depends on details of the experimental setup (especially the presentation rate), the graphs of R over M all collapse to a curve very close to the prediction (1.41), if n is set to M and $\langle k \rangle$ is used as an estimate of R, as shown in fig. 1.9. The same scaling appeared when short sentences (facts) were used as items instead of words.

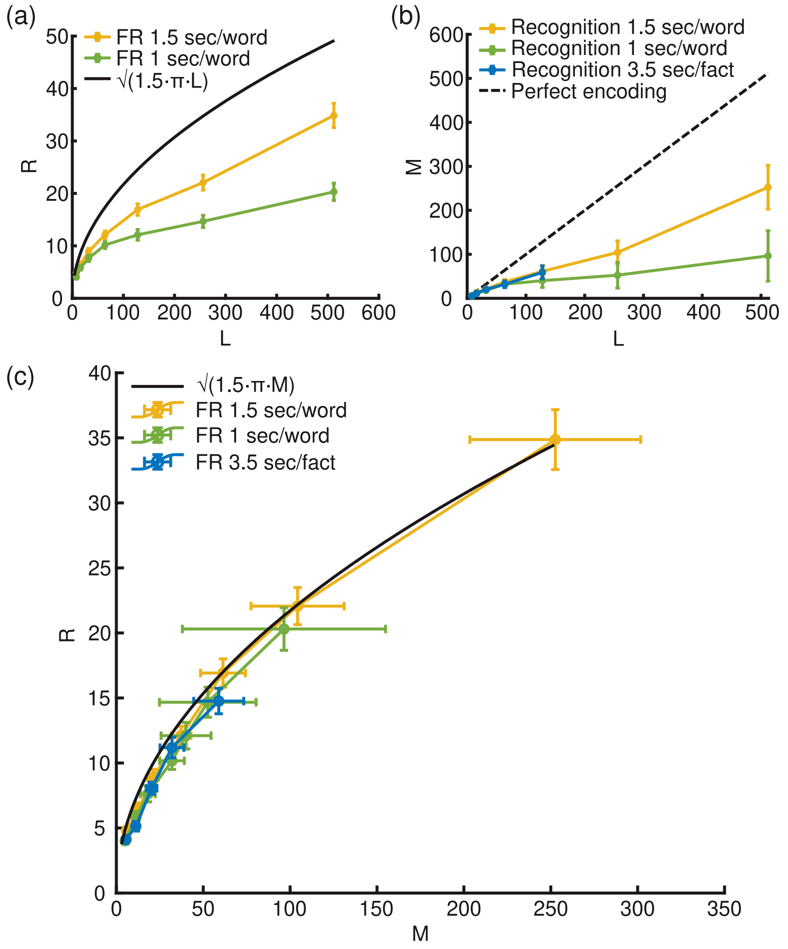


Figure 1.9 – Both R over L (a) and M over L follow power laws depending on the presentation rate and the nature of items (words or facts). However, when R is plotted over M, all curves collapse together, and closely follow the predictions from the graph model. This plot shows the experimental results of Naim et al., and it is taken from [1, Figure 2] with permission from the authors.

Remarkably, this happens without any kind of tuning, since (1.41) has no free parameters to be fitted. Thus, the conclusion is that there seems to be a "fundamental" pattern in human recall that can be modelled following the first principles stated in sec. 1.3.

This result is very interesting: the graph model of recall starts with very strong assumptions and is sufficiently simple to be studied analytically. Nonetheless, it is able to capture the relation between items that are recalled and items that can only be recognized, two high level tasks of the human brain requiring billions of densely interconnected neurons to be accomplished.

Several questions naturally arise:

- What are the other predictions that this model can make, and can they be experimentally verified?
- How general is the graph model? Can it be applied even to other kinds of recall processes (e.g., cued recall)?
- The result from (1.41) hinges on assuming specific statistics for the similarity matrix S. In particular, such a matrix is assumed to be constructed from the realization of i.i.d. random variables. Can this be experimentally verified, perhaps by directly measuring the overlaps between neural populations corresponding to different items?

1.6 Models' shortcomings

Ultimately, in the limit $p \to 0$, the graph memory model is very basic. It does not address any of the effects studied for free recall (e.g., primacy, recency, contiguity, see sec. 1.1.2).

In a sense, it considers only a common feature of recall processes (power law scaling), without saying anything of the details.

For now, the model has been tested only on lists of random words. While Ch. 2 will examine the case of semantically related words, it would be interesting to see if the same "fundamental law" holds for nonverbal items, such as pictures or musical motifs. This poses some issues: for instance, how can a subject prove to have recalled a picture? By drawing it? Or would a sufficiently detailed verbal description suffice? The latter possibility has been examined in [9]. In that case, it has been observed that sometimes random descriptions, generated by subjects that have not seen any picture, can still match, by pure chance, some specific picture that has been shown — which is of course a problem when determining if someone is recalling the correct stimulus, and not something else and totally unrelated.

However, there is a more fundamental issue that this model is completely neglecting: the items that have been shown are not the only ones present in the subjects' minds. Human memory does not contain just the n words that were presented in a list and are still recognizable given some cues. It rather contains a whole vocabulary, with associated experiences and emotions. Yet, during the recall phase of the experiment, participants are able to "filter out" all the other words, because they are instructed to remember just a particular set of words that was just shown. Sometimes, this process fails, and unrelated words are indeed recalled, forming the so-called "intrusions" [39].

None of this is considered in the models of sec. 1.4 and sec. 1.5, which in fact cannot explain intrusions.

Moreover, there is evidence that encoding happens during retrieval, and vice versa [28]. In other words, the associations between items that drive transitions change over time: the entries of S should not be fixed. However, it is difficult to explain how they change without adding a plethora of parameters.

There is also the issue of recognition, which is based on some variable of "familiarity" that allows participants to know if an item has been shown or not. The model assumes that a recognizable item can, in principle, be recalled, if a good cue is given. However, it is not clear if this is true, and it would be very difficult to test: suppose an item is recognized, and then a cue is given to recall that item, resulting in a successful retrieval. Does this happen because the item was indeed retrievable, or because the item being presented in the recognition test has reminded the subject of it, and so *now* they recall it, but they couldn't before?

Moreover, sometimes recognition can fail. For instance, a subject may have had a thought during the experiment, and then feel familiar with some word that was not shown, but that was really present in the mind during the experiment [28]. Terms related to words that are shown also tend to be mistaken as shown [40], and sometimes even items that are not recognized are actually still present in memory, and are then recalled [41].

Thus, recognition is, at best, an indication that items are available in memory, not a definitive proof.

Additionally, the time dimension is completely neglected by these models, which predict words recalled at regular times. However, experiments suggest that recall significantly slows down after the first few items [42].

Finally, there are the "technical" simplifications. For instance: real neurons can fire at different rates, they are not just either "not firing" or "firing" as supposed in sec. 1.4. They are not fully connected, and the links are not symmetric. Additionally, the same neuron can either make excitatory or inhibitory connections, not both at the same time (Dale's Law [43]).

Additionally, the mechanisms for synaptic adaptation and oscillation of global inhibition are quite *ad hoc*, and alternative processes may lead to the same result. However, as suggested by the authors of [42], oscillations in neural activity seem to be connected with recall processes [44] [45].

Several of these shortcomings are examined in the next section. Ultimately, however, the deeper questions, such as the role of intrusions, the role of context in recall, and the significance of recognition memory cannot be addressed at the current level of knowledge.

1.7 Model variations and other predictions

The model from sec. 1.4 can be extended to remove several assumptions, as it has been done in [42]. In particular:

- Discrete neurons are replaced with continuous ones that output a firing rate $r_i(t) \in \mathbb{R}$. The evolution of the network is then modelled by a set of differential equations. However, memories are still regarded as binary: they specify (sparse) patterns of neurons with firing rate above a certain threshold r_{thresh}.
- The mechanism of transition is the same as in sec. 1.4, with an oscillating global inhibition field φ.
- Synaptic adaptation is replaced by neuronal noise. In this way, when the network transitions from a mixture to a memory, it has a high possibility of not revisiting the previous active memory due to noise. However, this is not certain, as in the case of sufficiently strong synaptic adaptation. At times, the network will revisit the same memory for multiple times. This allows modelling the Inter-Retrieval Times (IRTs), i.e. the intervals elapsed between each recalled word and the next, which agree well with experiments.
- Off-diagonal terms can be added to the synaptic matrix T to model the contiguity effect, i.e. that words appearing closer during presentation are recalled in closer positions during recall.

However, these modifications add even more parameters and make the neural network model even more complex: for a reasonably sized network, numerical simulations require ~ 1000 differential equations to be integrated. Additionally, the model was tested only on free recall of short lists (16 words), with participants being presented with many lists (16) during each session. This adds some undesired effects: subjects know the lists' length and can construct strategies to improve their performance, as noted in [46].

Similarly, the graph model (sec. 1.5) can be extended to fit all the interesting effects of free recall (e.g., primacy, recency, contiguity) by adding other terms in the definition of the similarity matrix S [4]. This shows that it is at least as flexible as other recent models of free recall.

However, the added variables are still phenomenological in nature: they are introduced to fit effects *a posteriori*, they cannot be measured outside a recall experiment.

On a different note, the graph model possesses a variable which can be measured independently: the sparsity p of the neural representations. In the previous section, the limit $p \to 0$ was examined. However, if $p \ll 1$ is nonzero, the model leads to some interesting results. For instance, it predicts that items possessing, by chance, larger encoding patterns (i.e. containing more active neurons than the average Np), have a higher probability of recall [4]. Moreover, such "easy" words tend to appear earlier during recall, and then inhibit the retrieval of more "difficult" words, i.e. those with smaller neuronal sizes. This is because a pattern with more active units will tend to have, on average, larger overlaps with other patterns. Starting from a random word, transitions will quickly lead to one such pattern. After that, only patterns of similar size will be explored, until a loop is entered. These results are both confirmed experimentally. Moreover, pattern sizes may be measured directly by a brain scan, but such an experiment has not yet been performed.

The graph model (sec. 1.5) can also explain why, depending on how a list is constructed, words with lower or higher syllabic length tend to be recalled more [3]. If the performance is compared between lists containing words of the same length, then it is highest for the lists with the shortest words. However, if the presented list has words of any length, then longer words will be recalled more. This is explained by assuming that longer words, due to their higher variability, are associated to neural encodings with on average Np active neurons (as all other words), but with a higher variance of activation. In this way, longer words can have neuronal sizes bigger than shorter words, resulting in them being easier to recall and "masking off" other more "difficult" words.

Finally, the idea of recall terminating when the dynamics reach a loop is backed by some evidence: for instance, the probability of recall termination is higher following repetitions or intrusions rather than correct responses [47].

Chapter 2

Experiment

2.1 Preliminaries

The graph model of free recall (sec. 1.1.2) makes accurate predictions with no need of parameter tuning, which is compelling evidence for it being on the right track.

Effectively, the entire dynamics of the model are driven by the statistics of the similarity matrix S, which specifies how much neural encodings of different items overlap with each other.

Such a matrix is then a natural target for new experiments seeking to validate or reject the model.

For instance, one possibility would be to directly measure the entries of S, which are identified with overlaps of neural populations, through brain imaging (e.g., fMRI).

However, before committing so many resources to a complex experiment, it is proper to first obtain independent confirmation of the previous results, while also exploring indirect ways to characterize the entries of S.

Thus, without a way to directly access neural encodings, similarities between items can be estimated from the items themselves. For words, perhaps the most immediate possibility is that of semantic relatedness. It makes intuitive sense that overlapping populations of neurons should activate for items with similar meaning, because they react to shared features.

Here, "semantic relatedness" is a fuzzy concept indicating a measure of the "likeness in meaning" of words. Humans have an intuitive understanding of how "close" different words are. For instance, it is clear that `apple` and `banana` are more "similar" in meaning than `car` and `star`, even if the latter are more textually similar. In general, two words are *related* if they share any kind of lexical or functional association [48] (e.g. `house-rent`), and are said to be *semantically similar* only if they are synonyms (e.g. `happy` and `cheerful`) or share an "is-a" (hypernym-hyponym) relationship (e.g. `apple-fruit`).

Surprisingly, relatedness can be reliably quantified. One way to do that is by building *onthologies*, such as WordNet [49], which are large curated graphs connecting words with lots of functional relations. This allows to define the similarity distance between words as the length of the shortest path connecting them in the graph, so that related words are those separated by the lowest distance.

Another approach is through statistics. In linguistic, the "distributional hypothesis" states that words with similar meaning tend to appear in similar contexts [50]. Thus, by analyzing the co-occurrence statistics of words in large corpora of text, one can measure their relatedness. Algorithms such as Latent Semantic Analysis (LSA) [51] map words to a vector space of dense embeddings, in such a way that words that often appear in similar contexts (and thus are similar according to the distributional hypothesis) correspond to vectors that are closer in the space of embeddings.

Experiments suggest that semantic relatedness as measured by LSA influences the order of words during free recall: items with close embeddings tend to be retrieved in succession, and with less time in between them [52]. The same happens, in the framework of S. Romani, S. Recanatesi et al. for items with high neural overlaps [42, fig. 5]. This suggests a correlation between the entries of S and semantic relatedness of items, which is the target of the present chapter. In other words, items that mean similar things should have larger neural overlaps.

The idea is then to perform a free recall experiment on lists of words sharing strong semantic bonds. If S does indeed depend on word relatedness, then the performance of recall should decrease, because larger correlations in the entries of S increase the likelihood of loops (as can be seen in the fig. 1.8 for $p \neq 0$). Specifically, for the same number M of words that can be recognized (which is assumed to correspond to the number of items retained in memory), the number of recalled words R should be lower than that obtained with independent entries in S, i.e. $R < \sqrt{1.5\pi M}$. This is consistent with past experiments, suggesting that LTM may be impaired by semantic relatedness [53].

Moreover, consider lists composed of words that can be cleanly divided into two well-separated "categories" (e.g. `animals` vs. `vehicles`). Then, if attention is paid equally to each word, both sets should have a similar number of retained words. However, in the graph model (sec. 1.5), transitions happen only towards items with strong overlaps, which in this case will likely belong to the same category. Thus, depending on which words are recalled at the start of recall, there will be an asymmetry, with more words recalled from a category than the other, even if, in principle, both should be equally recalled.

The amount of asymmetry will depend on the size of semantic correlations in the entries of S. If the two categories are so well separated that all items belonging to different clusters have no overlap, then only items from a single category will be recalled. More realistically, participants will tend to recall a few words from the same category, then "jump" to the other category and recall another few words, and so on, with the first appearing category being better recalled than the other.

In fact, free recall experiments on categorized lists have shown that such *clustering effect* is indeed present [54] [55] [56].

However, no experiment as of now has compared recall and recognition performance on categorized lists, meaning that no data is available to immediately test these hypotheses. Therefore, the next section will discuss the details of a new proposed experiment to measure both the reduction in recall performance and the asymmetry of retrieval for lists of words categorized into two well-separated groups.

2.2 Methods

The proposed experiment closely follows the procedure from [1, Suppl. Mat.], with the main difference being that only lists of length $L = 64$ are presented.

This is because of two practical reasons:

- Adding more list lengths requires finding many more participants, which in turn would make the experiment exceedingly long. Due to time constraints, this work is limited to a "pilot experiment", to eventually inform a later investigation at a larger scale.
- Building categorized lists for a recall/recognition experiment requires finding at least L words for each category. This means that L cannot be too high: words strongly related to a category are very limited in number, so building longer lists would necessarily require choosing words that are less and less related. $L = 64$ is found to be the highest value for which lists of only two categories (the simplest case to analyze) can be built effectively, and the scaling predicted in (1.41) is particularly evident for higher L.

The experiment is run online[1], on a specifically built web app. It is targeted to native Italian speakers, and so both the interface and the presented words are in Italian.

Upon first connecting to the web app, each participant is randomly sorted in either the **main group** or the **control group**. Subjects in the main group are shown categorized lists (32 words per category), while the others are shown lists of random words mirroring those of Naim et al. experiment [1]. All details on the construction of the lists are reported in sec. 2.2.1.

All lists contain exactly $L = 64$ items, with no intralist repetitions, and with the items randomly shuffled at the start of each trial. Before starting the experiment, the participants are informed that it consists of two tasks involving memory, which require undisturbed attention, and that will last less than 10 minutes each. They are told to perform the two tasks on two consecutive days, preferably at the same hour and in the same place. Basic info (age, gender, and e-mail) are collected to ensure data integrity.

The two tasks are as follows:

- **Free Recall (A)**. At first the explanation from fig. 2.1 is shown. When the participant clicks on "Comincia l'Esperimento", after $\sim$ 1 s the list's presentation starts. Each word is shown for 1 s, with 500 ms of blank screen between words, such that the total presentation rate is 1.5 s/word.

  Within 1 s of the final word shown, a single-line text field for user input appears on the screen, requiring the participant to type all words they can recall. The form is cleared upon pressing `space` or `enter`, meaning that subjects only see the word they are currently typing.

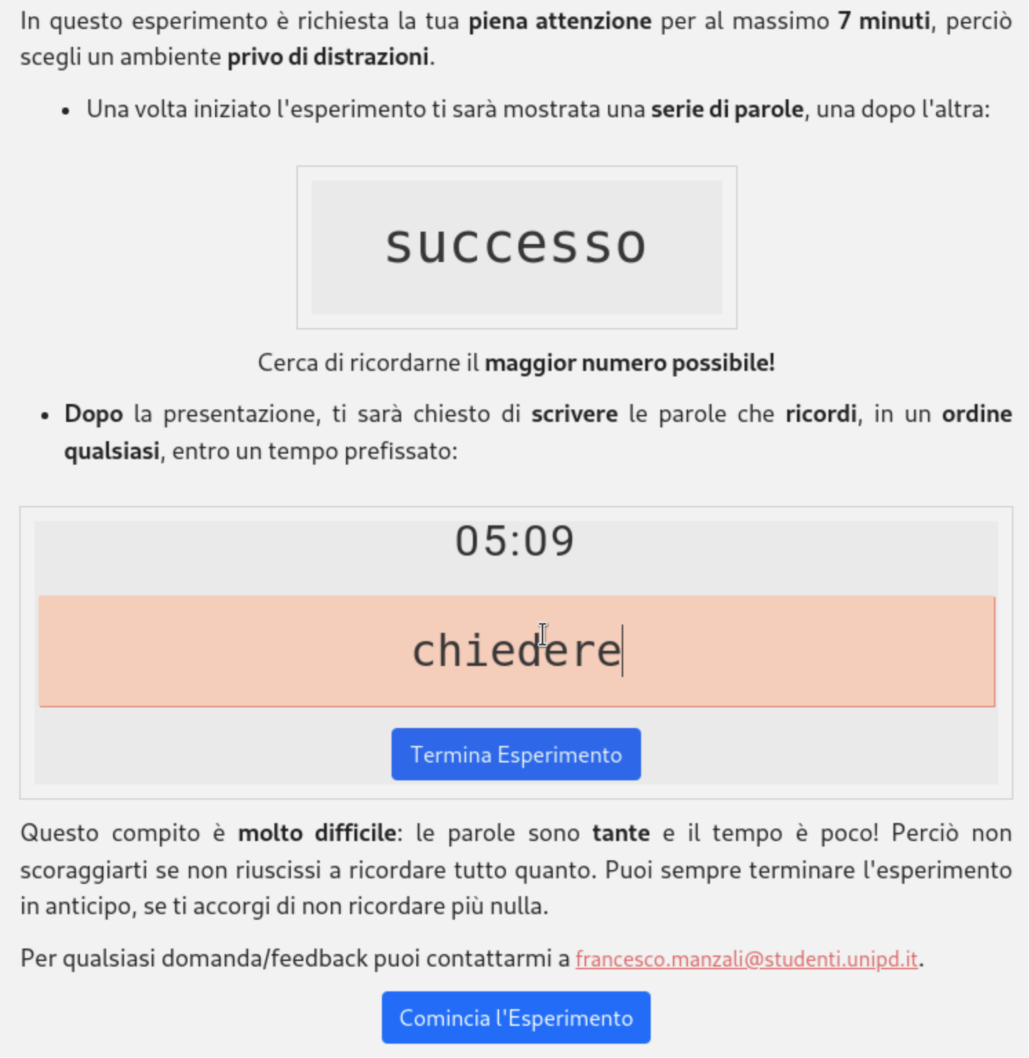
  A timer of 6 minutes is shown over the input form. When time runs out, or when the participant decides to terminate the experiment early, the text field is removed and a thank-you message is shown, prompting the user to exit from the web app.

[1] ∧The address, at the time of writing, is `https://goldshish.it/parole`. The web-app is coded with React.js, and works both on desktop and mobile.

- **Recognition (B)**. At first the explanation (fig. 2.2) is shown. When the participant clicks on "Comincia l'Esperimento", after ~ 1 s the list's presentation starts, following the same setup from the free recall task. Note that this list is always different from the one presented in (A). In particular, if during (A) a categorized list was shown, then the list used for (B) always contains words from entirely different categories.

  Within 1 s of the final word shown, the recognition phase begins. A pair of words appears on the screen, one under the other. One word is taken from the previously shown list, while the other is an unrelated distractor, chosen to have a similar lexical frequency as measured by the `wordfreq` package [57]. In this way, participants cannot simply reason about the relative rareness of words in language. Otherwise, if all words shown were of common use, and a recognition pair involved terms such as `tulip` and `gobbledygook`, it would be clear, by rarity alone, that the second word is the distractor.

  Moreover, in the case of a categorized list, distractors belong to the same categories of



In questo esperimento è richiesta la tua **piena attenzione** per al massimo **7 minuti**, perciò scegli un ambiente **privo di distrazioni**.

- Una volta iniziato l'esperimento ti sarà mostrata una **serie di parole**, una dopo l'altra:

successo

Cerca di ricordarne il **maggior numero possibile!**

- **Dopo** la presentazione, ti sarà chiesto di **scrivere** le parole che **ricordi**, in un **ordine qualsiasi**, entro un tempo prefissato:

05:09

chiedere

Questo compito è **molto difficile**: le parole sono **tante** e il tempo è poco! Perciò non scoraggiarti se non riuscissi a ricordare tutto quanto. Puoi sempre terminare l'esperimento in anticipo, se ti accorgi di non ricordare più nulla.

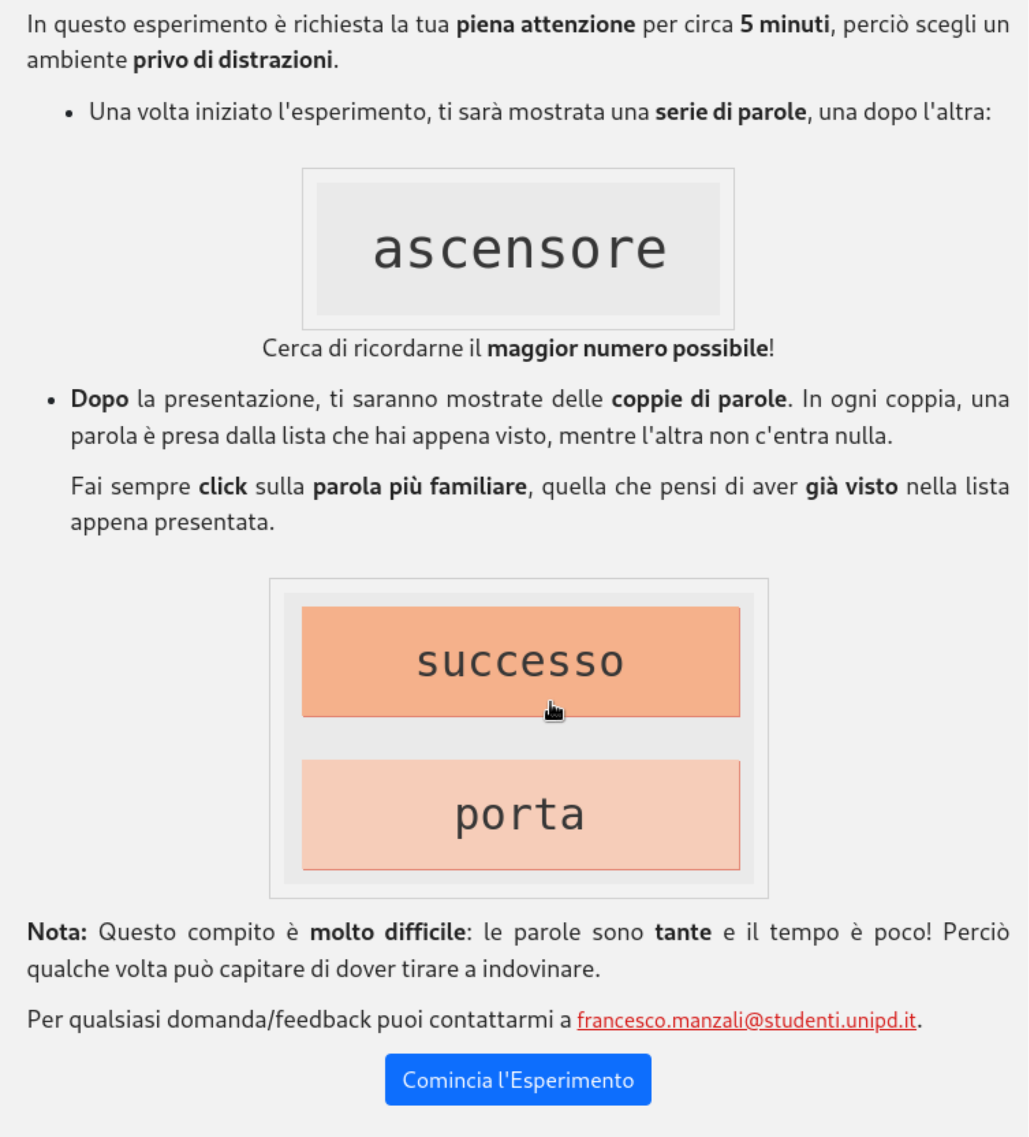
Per qualsiasi domanda/feedback puoi contattarmi a francesco.manzali@studenti.unipd.it.

Figure 2.1 – Screenshot of the initial explanation for the free recall task (A). The translation is as follows: "*In this experiment your full attention is required for at most 7 minutes, so choose an environment free from distractions. Once the experiment has begun, you will be shown a sequence of words, one after the other. Try to remember as many words as possible! After the presentation, you will be asked to write all the words you remember, in any order, within a set time. This task is very hard: the words are many and the time is short! So don't get discouraged if you don't manage to recall all of them. You can always end the experiment early if you realize you don't remember anything else.*"

the shown list[2]. Otherwise, it would be easy to distinguish them without the need to remember all presented words.

Participants simply click on the word they recognize as belonging to the presented list. A total of $L = 64$ recognition pairs are shown one after the other, the next appearing immediately after the user has expressed a choice for the previous one. No time limit is enforced, with each pair generally taking only a few seconds to be resolved.

After the last choice has been made, a thank-you page is shown, and the participant is prompted to exit from the web app.



In questo esperimento è richiesta la tua **piena attenzione** per circa **5 minuti**, perciò scegli un ambiente **privo di distrazioni**.

- Una volta iniziato l'esperimento, ti sarà mostrata una **serie di parole**, una dopo l'altra:

Cerca di ricordarne il **maggior numero possibile**!

- **Dopo** la presentazione, ti saranno mostrate delle **coppie di parole**. In ogni coppia, una parola è presa dalla lista che hai appena visto, mentre l'altra non c'entra nulla.

  Fai sempre **click** sulla **parola più familiare**, quella che pensi di aver **già visto** nella lista appena presentata.

Nota: Questo compito è **molto difficile**: le parole sono **tante** e il tempo è poco! Perciò qualche volta può capitare di dover tirare a indovinare.

Per qualsiasi domanda/feedback puoi contattarmi a francesco.manzali@studenti.unipd.it.

Figure 2.2 – Screenshot of the initial explanation for the recognition task (B). The translation is as follows: "*In this experiment your full attention is required for at most 5 minutes, so choose an environment free from distractions. Once the experiment has begun, you will be shown a sequence of words, one after the other. Try to remember as many words as possible! After the presentation, you will be shown pairs of words. In each pair, one word is taken from the list that you have just seen, while the other is unrelated. Click always on the more familiar word, the one you think you have already seen in the previously presented list. This task is very hard: the words are many and the time is short! Thus, it may sometimes happen that you have to guess.*"

[2] ∧This is also why one needs L words for each category. A presented list contains 32 of them, for each of the two categories. The other 32 are needed as distractors during the recognition task.

The two tasks are accessible from the same web address. The personal information reported by the participants is used to correctly show the second task to someone that has already completed the first. Note that no information on the lists' length is ever given, and counters such as "x more words remaining" are shown neither during presentation nor the recognition task.

The order of the tasks is fixed: A always precedes B. This is done mainly for simplicity, since they happen on different days and with different lists. However, some effects of order may be present: for instance, during the second task, participants may remember roughly how many words were shown in the previous task, and thus better anticipate its complexity, adopting different strategies.

To properly measure such effects, one would need to randomize the order of tasks over two groups of participants. This analysis is left for future work.

Apart from the use of categorized lists, the experimental details closely follow those of Naim et al. [1, Suppl. Mat.]. A significant difference is in the number of recognition pairs. The original study asked just 5 pairs to each participant, with only the first one considered for the analysis, possibly to avoid any delay between presentation and recognition, or effects of other encoding/retrieval processes happening during a prolonged recognition.

However, in this study, the number of recognition pairs is equal to the number of items in the presented list (L). This allows estimating the average number of retained items for each individual participant, not only for the group of subjects as a whole.

2.2.1 Dataset

Categorized Lists

A categorized list consists of 64 words, of which exactly 32 belong to a category (a), and the other 32 to another category (b), with the following requirements:

- Words in a category should be strongly related, so that just a few words of the same category suffice to identify it. For instance, `gilet`, `foulard` and `glove` can be recognized as articles of clothing. Since 64 words are required for each category, categories should be sufficiently "general" in their meaning, encompassing lots of terms.
- The two categories that appear in the same list should be well separated in meaning from each other: if a word from (b) is presented along with few words of (a), it should be easily identifiable as intrusive (e.g. in `jeep`, `caravan`, `sandal`, `railroad`). Moreover, thinking about words from a category should not prompt words from the other category. For instance, `wool` is related to `clothes`, but also to `sheep`, which is an `animal`. So, if a list contains words of `animals` and `clothes`, it should not contain `wool`.
- Words of (a) and (b) that appear in the list should have similar statistical characteristics, in terms of lexical frequency and syllabic length. This is because different attention is paid to common or rare words [58], and recall is higher for rare or very frequent words than for terms of intermediate frequency [59]. Analogously, in lists with words of different sizes, longer words are more easily recalled [3].
- Common use words that are not ambiguous are preferred.

An initial set of general categories is taken from [60, Experiment 1], where participants were asked to freely write a list of random concrete objects, which were then grouped by independent judges, keeping only the categories that appeared the most and did not dependent on the subjects' context. Such a procedure is much less discretionary than simply choosing a few categories at random, since there is evidence that such groups may reflect common mental models of the world.

Of the 12 categories taken from [60, Table 1], two (`furniture` and `furnishings/fittings`) are merged because of their similarity, one (`kitchenware`) is removed because of insufficient generality, and two (`weather` and `colors`) are added. At the end, the categories considered for constructing lists are the following 12:

animals, body parts, clothes, colors, furniture/furnishings/fittings, housing buildings, plants, stationery, weather, vehicles, foods, hobbies/sports

For each of these categories, at least 64 related terms are written by hand, starting from the $\sim$ 12 found in [60, Experiment 2], and then using the web and Italian dictionaries. Unfortunately, there is no efficient way to automate this task: procedural generation through ontologies or embeddings leads to many words that are either too rare, too abstract, too ambiguous, or simply different forms of the same word (singular/plural, feminine/masculine).

After the required amount of words has been gathered, their embeddings are computed using the pretrained model from ConceptNet Numberbatch [61]. In this way, each word is associated to a dense vector of $d = 300$ entries. The semantic relatedness between two words x and y with embeddings $\boldsymbol{v}$ and $\boldsymbol{u}$ can now be measured as the cosine of the angle θ between the two vectors:

$$\text{Semantic relatedness}(x, y) \equiv \cos(\theta) = \frac{\boldsymbol{v} \cdot \boldsymbol{u}}{\|\boldsymbol{v}\|\|\boldsymbol{u}\|} = \frac{\sum_{i=1}^{d} v_i u_i}{\sqrt{\sum_{i=1}^{d} v_i^2}\sqrt{\sum_{i=1}^{d} u_i^2}}$$

Two words with equal meaning have $\cos(\theta) \approx 1$, because their embeddings "point in the same direction", and so $\theta \approx 0$. Words that are opposites have $\cos(\theta) \approx -1$, while unrelated words have $\cos(\theta) \approx 0$.

Similarly, the semantic distance between words is defined as:

$$\text{Semantic distance}(x, y) \equiv 1 - \cos(\theta)$$

An embedding is assigned to each category by averaging the embeddings of the terms they contain, i.e. by computing its centroid. This procedure is validated by inspecting the word that is closest to these centroids. In all cases it is either a descriptor of the category itself, or a term related to it (tab. 2.1).

Using the centroids, the semantic distances between each pair of categories are computed (fig. 2.3).

It is immediately clear that not all categories can appear with each other. For instance, `furniture/furnishings` and `housing buildings` clearly contain related terms, as for `plants-colors` and `stationery-colors`. This analysis marks also associations that are

Category	Centroid	Translation
animals	animale	animal
body parts	anatomia	anatomy
clothes	giubba	jacket
colors	blu	blue
foods	paste	pastries
furniture & furnishings	canterano	chest of drawers
hobbies & sports	sport	sport
housing buildings	palazzina	building
plants	clivia	clivia [a type of lily]
stationery	matite	pencils
vehicles	veicolo	vehicle
weather	burrasche	storms

Table 2.1 – Terms with the closest embedding to the centroid of each category.

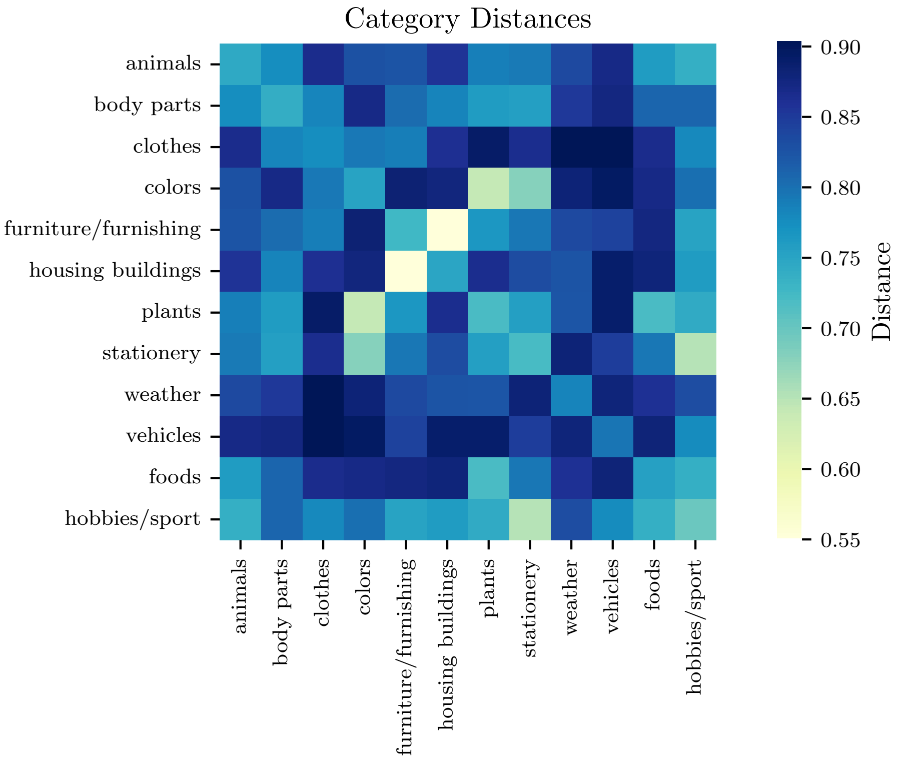


Figure 2.3 – Semantic distances between categories, computed as $1 - \cos(\theta)$ of the centroid of their embeddings. Terms on the diagonal are filled with the average of the other entries in the same row.

not immediately obvious, such as `stationery-hobbies` (many hobbies, such as `painting`, `drawing`, `writing`, require stationery equipment).

For building categorized lists, only pairs of categories with semantic distance higher than the average value of distances (0.8) are considered.

Then, to validate that a pair of categories is well separated, all embeddings of their words are embedded in a 2d space with the PaCMAP algorithm [62], a tool to reduce the dimensionality of sets of vectors such that both the relevant local and global structures are preserved. Then, KMeans [63] is used to cluster the embedded points into two groups in an unsupervised way. A pair is considered well separated only if this step correctly recovers the known division in the two categories.

In practice, all 39 pairs of categories with semantic distances higher than 0.8 successfully pass this test.

However, many of them still contain words of different categories that "remind of" each other. To test this, a dataset of word associations from Small World of Words [64] is used. Such a dataset has been constructed by asking participants to choose 5 words related to a given cue, and then collecting the mental links (e.g. `wool` → `sheep`) that are reported by most people.

As expected, most of the considered pairs contain words of different categories that are commonly linked. This happens for general associations (e.g. `frigorifero`→ `fame`, transl. `fridge` → `hunger`, is a link between the categories of `furniture` and `food`), or when a word has several senses: for instance `calcio` in Italian means both `football` (a `sport`) or `calcium`, an element present in `bones` (`body parts`).

Thus, the 11 pairs with < 10 associative links between categories are selected, and words are manually swapped until the number of inter-category links drops to 0. After this step, all previous analysis is repeated to confirm the results, leading to the following final 11 pairs of categories:

clothes-animals, furniture/furnishings-animals, housing buildings-animals, plants-housing buildings, weather-stationery, vehicles-body parts, vehicles-clothes, vehicles-housing buildings, vehicles-plants, foods-clothes, foods-vehicles

Note that few of the originally chosen categories (e.g. `colors`) do not appear in these pairs, because they are not sufficiently well separated from all others.

As a final step, one needs to select 32 words from each category of each pair to form the final categorized lists.

This is done while uniformizing as much as possible the distribution of lexical frequencies (as measured by `wordfreq` [57], using the largest corpora available) and syllabic lengths.

In essence, words of each category are divided into bidimensional bins, each containing words of the same syllabic length, and within a narrow range of (log)frequency[3].

Then, for each word of a category (a) selected from a bin, another word from category (b) is selected from the same bin, until a total of 32 words per category is achieved. If there are no more words of (b) within the selected bin, a word within the same range of frequencies, but any syllabic length, is selected instead. In practice, there is no necessity to ever lift this last constraint.

At the end, the final result is 11 pairs of words which have very similar joint probability distributions of frequency/syllabic length, the exact same marginal distribution of frequency, and very similar marginal distributions of syllabic length (fig. 2.4).

[3] ∧Specifically, let $f_{\max}$ and $f_{\min}$ be respectively the frequencies of the most common and rarest word between all categories. The (log)frequency bins are obtained by dividing the interval $[\log_{10} f_{\min}, \log_{10} f_{\max}]$ in 20 equal parts. Logarithmic bins are necessary because word frequencies follow Zipf's law, which is a power law distribution [65].

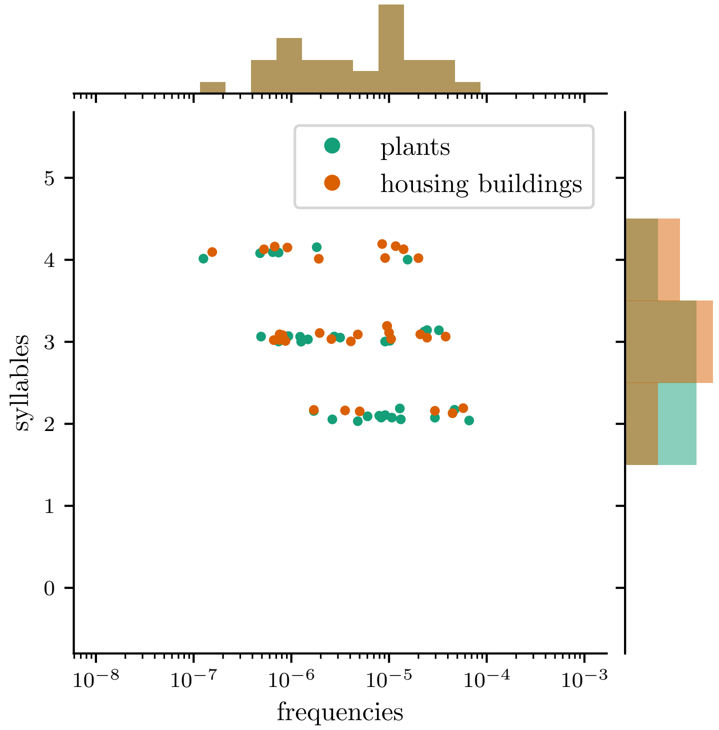


Figure 2.4 – Joint distribution of word frequencies and syllabic lengths for the pair `plants-housing buildings`. Points are uniformly shifted along the y axis to better visualize them.

Random Lists

Random lists are constructed from words taken from ItalWordNet [66], after removing slurs and non-Italian words with common usage (e.g. `barbecue`), and taking only words with frequency greater than 10^{-5}. This is done to replicate as closely as possible the setup from [1].

Another possibility would be to mimic the frequency distribution of words in the categorized lists. However, in this case, due to the constraint of choosing 64 terms for each category, rarer words are used, with frequency as low as $\sim 10^{-7}$. This is fine for a categorized list, because the categories act themselves as cues that aid recall, but would make random lists very difficult to remember.

Nonetheless, to avoid making the random lists too different from the categorized ones, the same syllabic length distribution is followed. Namely, for each of the 11 generated categorized lists, a random list is constructed choosing words with the same number of syllables of the words from the corresponding categorized list.

2.3 Analysis

A total of 59 people participated in the experiment. However, despite instructions and reminders, only 28 managed to complete both parts (recall and recognition). Of those, 2 participants prematurely clicked on the button that terminates the free recall experiment after typing the first remembered word. Thus, only 26 subjects have been included in the study: 14 received a *random* list, and the other 12 a *categorized* list[4].

Of these 26 participants, 15 identified themselves as females, and 11 as males. Most, but not all, are university students or graduates. Mean age is 24 ± 4 (min 17, max 39). The time between the two tasks ranges from 20h to 2 days and 18h, with an average of 1 day and 6h consistent with the provided prescriptions.

For each subject, the two main observables of interest are:

- The number R of words reported in the free recall experiment with a list of size $L = 64$. Obvious typos are corrected, while intrusions and repetitions are removed.
- The average number M of words retained in memory from a list of size $L = 64$, of the same type (random/categorized) of the one from the free recall part. This is computed from the number S of correctly recognized words in L two-alternative forced-choice recognition trials, correcting for random guesses (see (1.1)):

$$M = 2S - L$$

The results, expressed as sample mean and standard error of the mean, are shown in fig. 2.5 and in tab. 2.2.

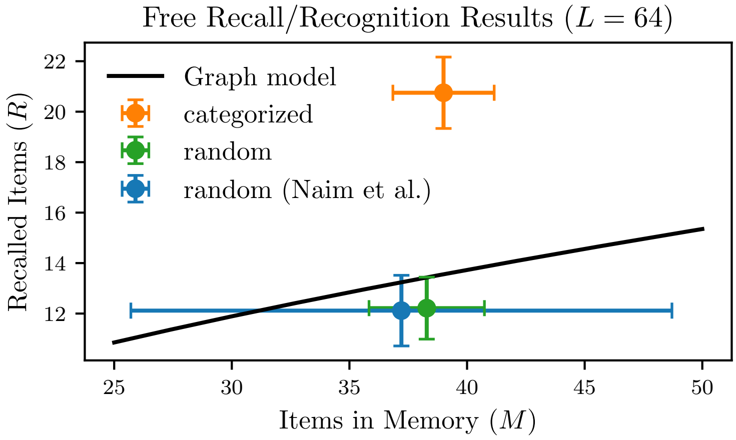


Figure 2.5 – Recalled words (R) and retained words (M) for random and categorized lists, compared with the predictions from the graph model ($R = \sqrt{1.5\pi M}$, black line), and with the datum for $L = 64$ from Naim et al. paper [1] (blue point).

	random	**categorized**
M	38.3 ± 2.5	39.0 ± 2.2
R	12.2 ± 1.2	20.8 ± 1.4

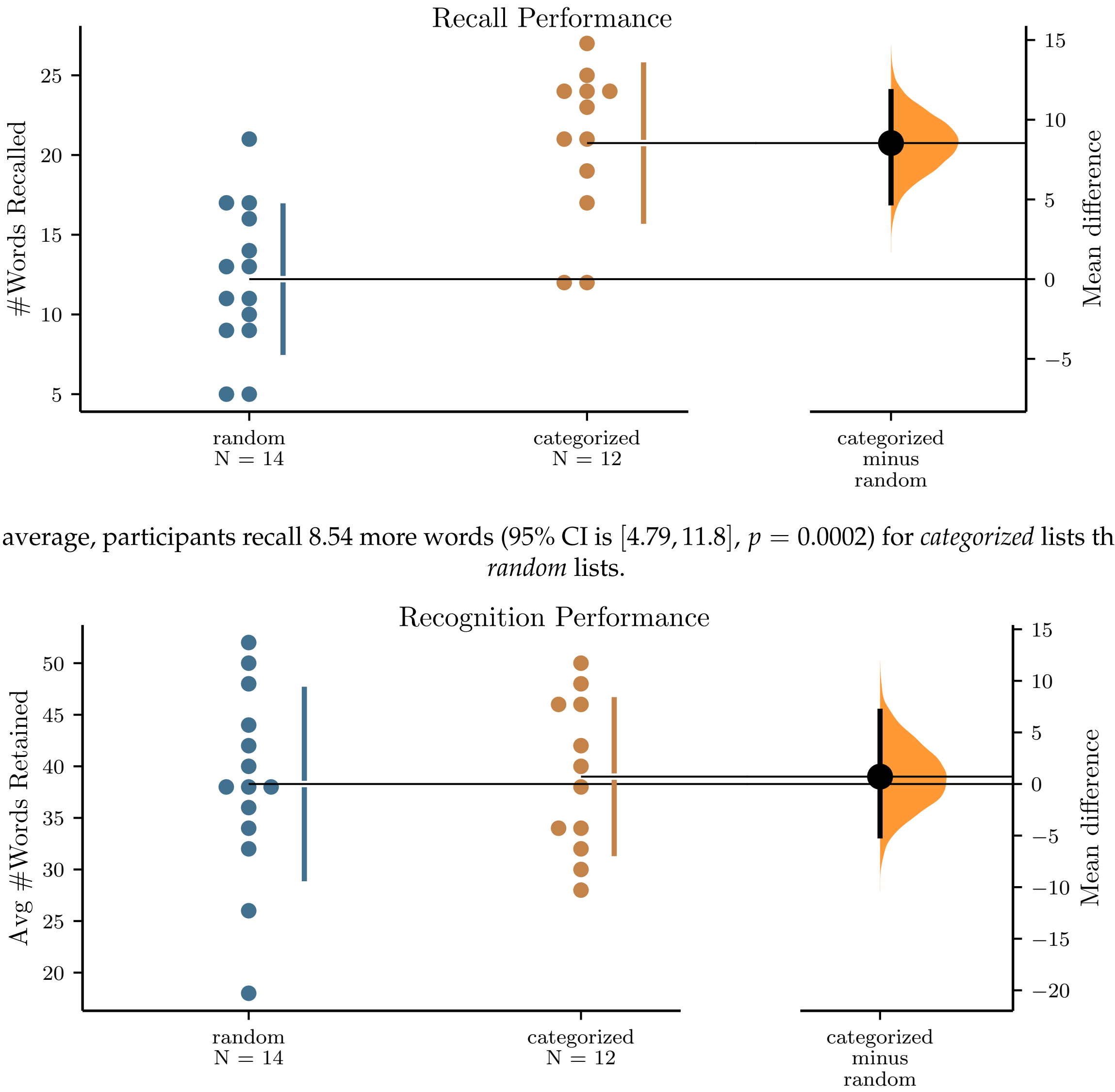
Table 2.2 – Experimental results for recalled words (R) and retained words (M).

The graph model agrees with the performance on lists of random words, but not so much for categorized lists.

[4] ∧The unbalance is because, due to random chance, more people from the *categorized* group quit the experiment after the first part.

2.3.1 Performance: Random vs Categorized

At a closer inspection, people tend to recall more words when categories are present (fig. 2.6a), but the number of retained words is mostly the same (fig. 2.6b).



(a) On average, participants recall 8.54 more words (95% CI is $[4.79, 11.8]$, $p = 0.0002$) for *categorized* lists than for *random* lists.

(b) Participants retain as many words for *categorized* lists as for *random* lists. The difference of averages between categorized and random is 0.71 (95% CI is $[-5.02, 7.05]$), but it is not statistically significant ($p = .8$).

Figure 2.6 – Gardner-Altman estimation plots [67] for the effect of categorized lists on free recall and recognition. The plots are made with the Python package `dabest` [68], and show all the data points, along with their mean and standard deviation (the vertical colored bars), and the bootstrap distribution of the difference between their means (5000 samples), with its 95% bias-corrected and accelerated (BCa) confidence interval [69]. p-values are computed with a two-sided unpaired permutation t-test (5000 reshuffles).

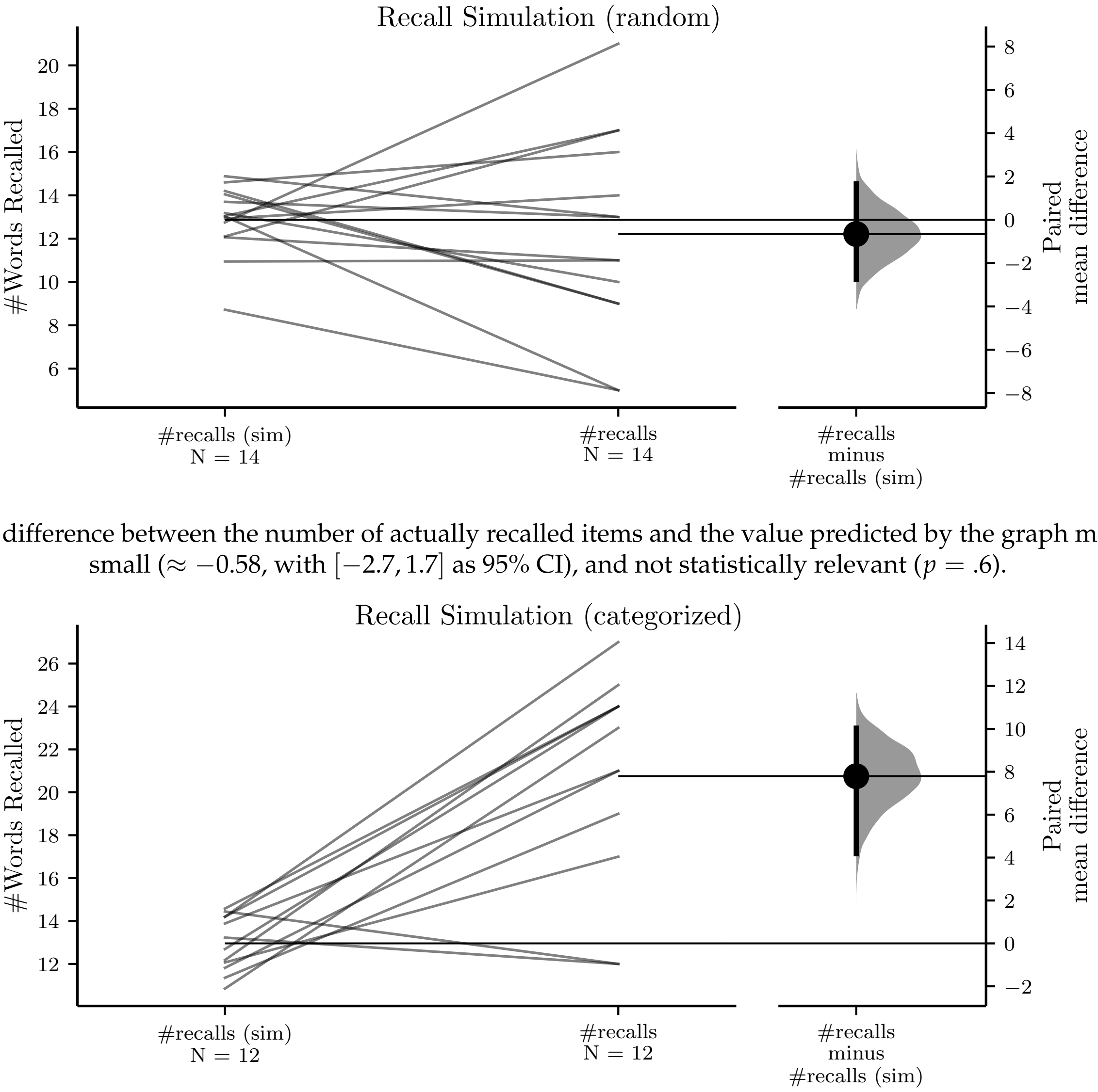
This is particularly interesting because the recall task always happens before the recognition one, and participants do not know that the presented list will be categorized before presentation. If subjects "adapted" to memorizing categorical words, then they would perform better on the recognition task, not the recall one.

Moreover, there is no evidence of participants typing random category-related words in the hope of guessing some of them: the average number of intrusions is 0.7, and is approximately the same for *random* and *categorized* lists. No more than 3 intrusions happen in the same trial.

2.3.2 Comparison with the Graph Model

The experimental results are directly compared with the predictions of the graph model as follows:

1. For each trial on a random/categorized list the average number M of retained words is computed.
2. The graph model is run on an $M \times M$ similarity matrix constructed from i.i.d. random values uniformly distributed in $[0, 1)$, leading to a sequence of recalled items.
3. The last step is repeated 1000 times to compute the predicted average number R_{sim} of recalled items if M items are stored in memory. This is then compared with the number R of recalled items measured experimentally.



(a) The difference between the number of actually recalled items and the value predicted by the graph model is small (≈ -0.58, with $[-2.7, 1.7]$ as 95% CI), and not statistically relevant ($p = .6$).

(b) Participants recall more items for categorized lists than what the graph model predicts (≈ 7.6 more, with 95% CI of $[4.0, 9.9]$, $p = .004$).

Figure 2.7 – Gardner-Altman estimation plots for the discrepancy between graph model and experimental results. In this case, data points are paired, because the same participant does the recall task (used for #recalls) and the recognition task (from which one estimates M, and then #recalls (sim)). Pairings are represented as gray lines. p-values are computed with a two-sided paired permutation t-test.

This analysis confirms that the graph model correctly predicts the performance on *random* lists (fig. 2.7a), but not that on *categorized* lists (fig. 2.7b).

However, in the case of *random* lists, the model can reliably predict the number of recalled words even for individual participants. This is a stronger result than that of [1], where, due to the small number of recognition trials, M was measured just as a group average, and not for each subject.

2.3.3 Graph Model with Correlations

Perhaps, the graph model does not work for *categorized* lists just because the similarity matrix S is constructed from i.i.d. entries, while in this case it should include also semantic correlations.

As observed before (fig. 1.8), adding correlations in S tends to reduce performance, and this conclusion holds even for semantic correlations.

For instance, consider the (symmetric) matrix of semantic similarities W between items of a *categorized* list (fig. 2.8).

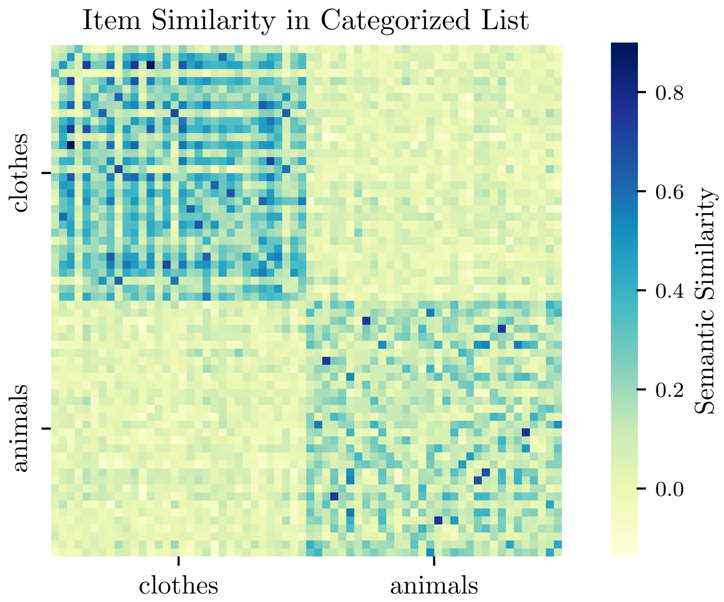


Figure 2.8 – Semantic similarities ($\cos\theta$ of embeddings) between items of a *categorized* list used in the experiment, including terms for `clothes` and `animals`. As expected, a block structure is evident: items of the same category are more similar, and items of different categories are less similar.

Entries in W are between $[-1,+1]$, but are mostly positive. Then, consider a similarity matrix S built from i.i.d. values uniformly distributed in $[0,1]$, and add to it the correlations from W:

$$\mathrm{S}_{\mathrm{corr}} = \mathrm{S} + \alpha \mathrm{W}$$

where $\alpha \geq 0$ determines the relative strength of the semantic correlations between items.

Then, to measure the effect of W, one simulates the dynamics over many realizations of $\mathrm{S}_{\mathrm{corr}}$ at a given α. At each trial, S is resampled, and W is constructed from the semantic similarities of one of the 11 *categorized* lists considered in the experiment, chosen at random.

The recall performance is shown in fig. 2.9 as a function of α. Note how, as α increases, the average number of recalled words decreases.

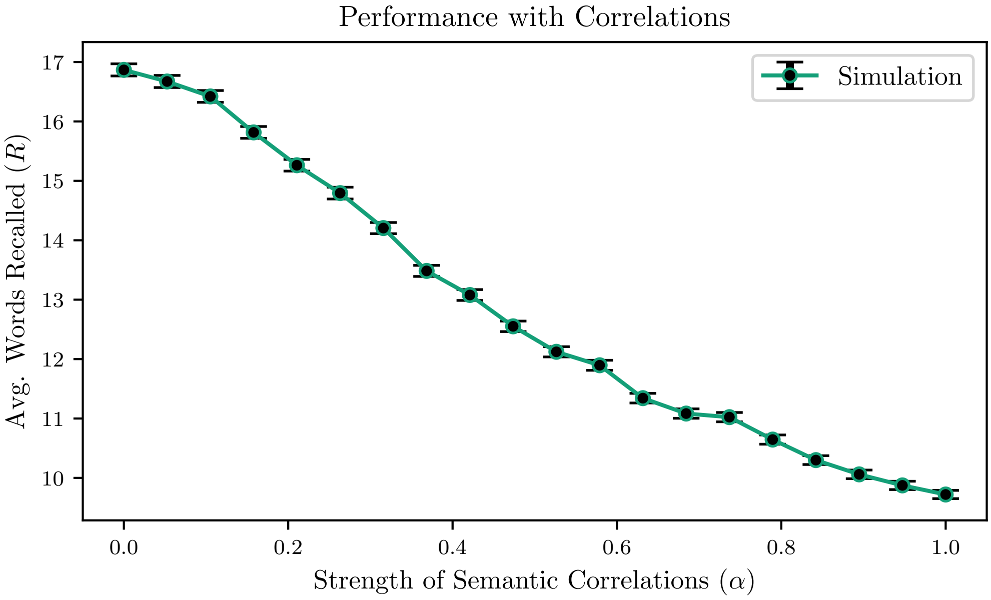


Figure 2.9 – Recall performance as predicted by the graph model in the presence of varying levels of semantic correlations.

This happens because for higher values of α transitions between items of different categories become impossible. Fig. 2.10 shows how items are connected by their similarity, and how, as α increases, they gather in separate groups.

This reduces the overall "connectivity" of the transitions graph $\mathcal{G}$. In fact, consider the partition of $\mathcal{G}$ into strongly-connected components $\mathcal{S}_i$, i.e. sets of nodes such that $\forall a, b \in \mathcal{S}_i$, there exists a path linking $a \to b$ and $b \to a$.

When a recall sequence enters a set $\mathcal{S}_i$, it can either form a loop between some or all of its elements, or leave it forever. Thus, the size of loops is constrained by the size of such strongly connected components.

As α increases, $\mathcal{G}$ is divided between more and more strongly connected components that are individually smaller (fig. 2.11), meaning that shorter loops become more likely.

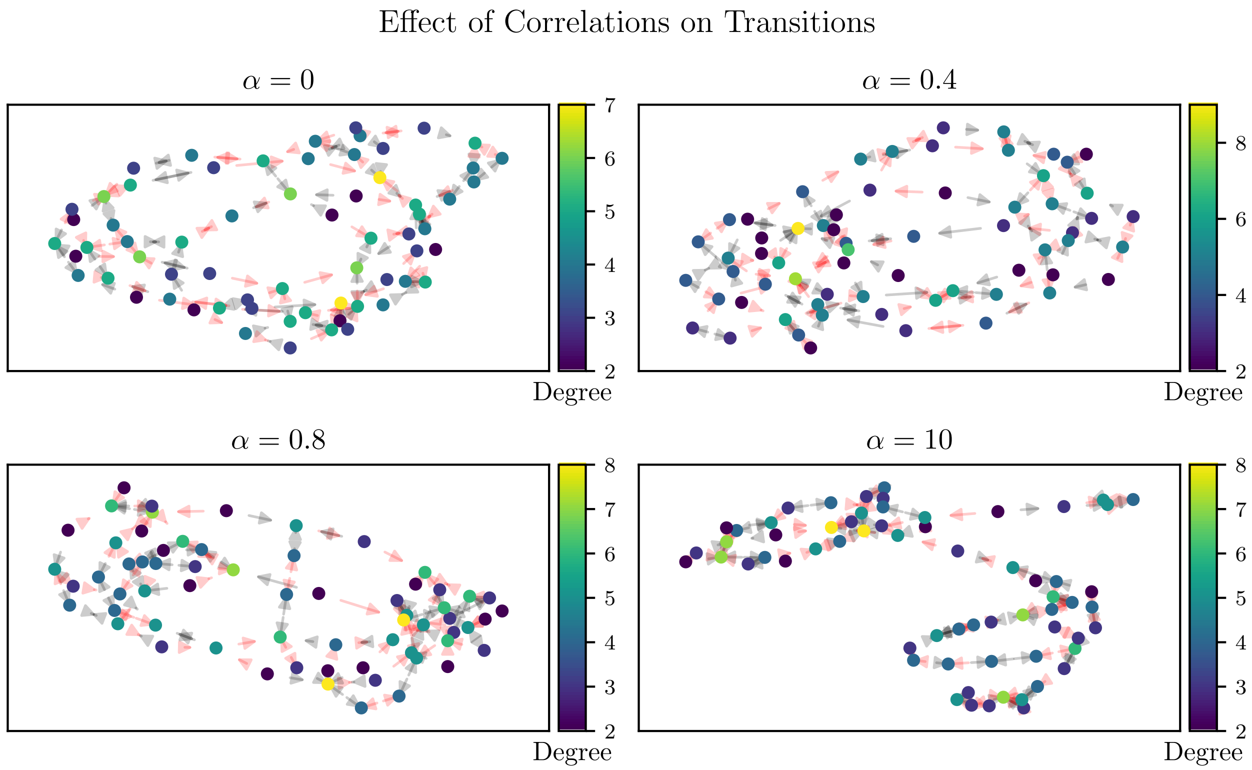


Figure 2.10 – Graph representation of all possible transitions between 64 items, each shown as a dot. Gray arrows point towards the items with the largest similarity, while red arrows to those with the second-highest similarity. The color of a node represents the number of transitions pointing to that node.

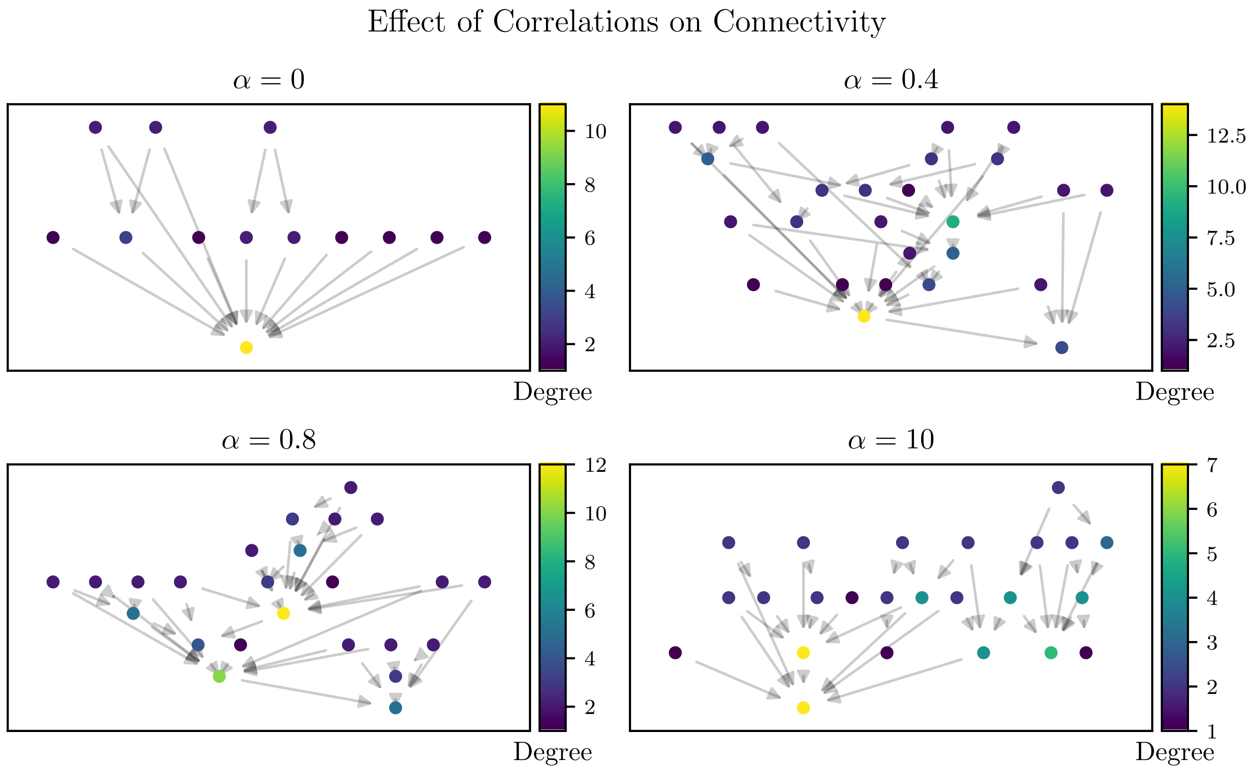


Figure 2.11 – Condensation of the graphs from fig. 2.10, where each dot represents a strongly-connected component of the graphs, colored by the number of nodes it contains. When $\alpha = 0$, there is only one large component where loops may happen — all the others have size 2, and cannot support loops because 2-loops are prohibited. For higher α, instead, there are more and more components that can support loops, resulting in shorter cycles overall.

2.3.4 Graph Model with Segregated Categories

A simple trick to adapt the graph model to *categorized* lists would be to apply it to the two categories as if they were two separately presented lists.

Perhaps, the graph model only applies to words in the same context, i.e. items that are accessed by the same original cue. In the case of a random list, this would include all the "words from the presented list" (thus excluding all other words that participants know, but that were not presented). Possibly, the categorized case originates two separate contexts, one for each category, and a retrieval process happens independently for each of them. This would mean that a separate sequence of recalled items is generated for each category, using an independent similarity matrix which, as it now involves only intracategory transitions, can be well approximated by i.i.d. entries.

However, such a tentative, but arbitrary, explanation is not even corroborated by evidence. In fact, from the recognition trials, one can estimate the average number M_1 and M_2 of retained items for each category. Then, the graph model is simulated as in sec. 2.3.2 to compute the average recalls R_1 and R_2 for each context, which are then summed to give the prediction for R.

The results, as shown in fig. 2.12, do not agree with the experimental evidence. Due to the limited sample size, however, this test is close to the significance threshold ($p < 0.05$).

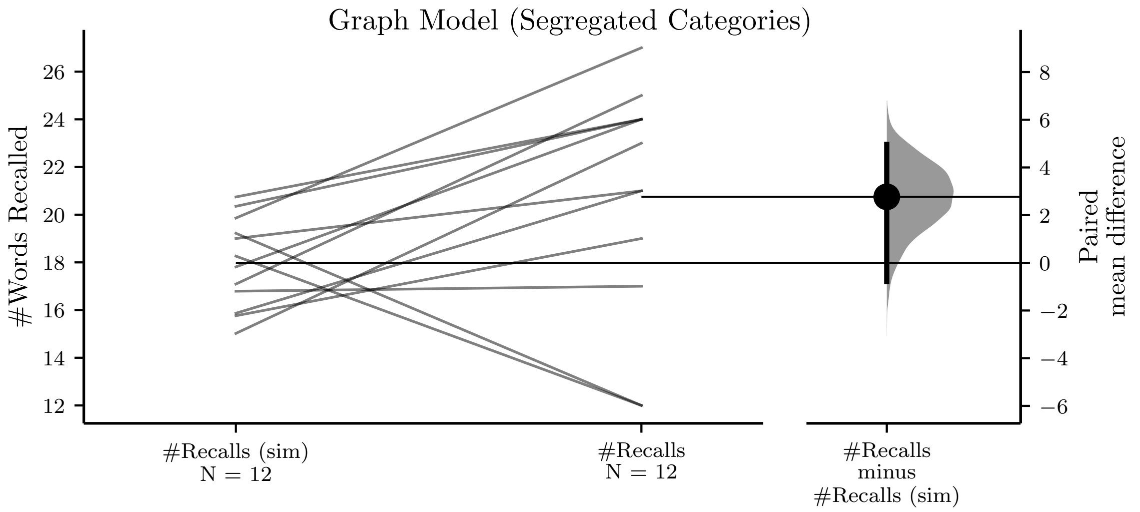


Figure 2.12 – Comparison with the predictions of a graph model simulated separately over the retained words of each category. Difference in means is 2.75 (95% CI is $[-0.9, 5.0]$, $p = 0.1$).

2.3.5 Clustering in Recall

Data suggests that, however, categories do interact in memory. In fact, participants tend to recall items in "clusters", i.e. groups of $2 \div 8$ items of the same category.

This can be evaluated using standard metrics of *semantic clustering* for free recall data, for instance those used in the California Verbal Learning Test [70].

Firstly, the *observed* amount of clusters in a sequence is defined as the number of adjacent items belonging to the same category. For instance, if the two categories used in the lists are denoted with A and B, a sequence (A, B, A, A, A, B, B) has exactly 3 clusters.

Clustering is significant when the number of observed clusters is higher than what would be expected from pure chance, which can be computed with combinatorics.

Consider a sequence of R items, n of which are of class A, and $R-n$ of the other classes $\neg A$. With these constraints, there are a total of $R!/[n!(R-n)!]$ possible arrangements.

Then, consider all lists with at least a pair (A, A) at a specific position i. The presence of the pair fixes 2 of the R items, and the other ($n-2$ of A, and $R-n$ of $\neg A$) can be arranged in a total of $(L-2)!/[(L-n)!(n-2)!]$ possible ways. Note that a pair can appear at $R-1$ positions in a sequence of R items, meaning that the total number of clusters in all the possible arrangements of R items with n belonging to class A is:

$$\text{\#Clusters in all arrangements} = \frac{(R-1)!}{(R-n)!(n-2)!}$$

Dividing by the number of possible arrangements gives the average number of clusters involving a single class A that should be expected in a randomly shuffled sequence of length R with n items belonging to A:

$$\text{Avg. Clusters of a Class} = \frac{(R-1)!}{(R-n)!(n-2)!}\frac{n!(R-n)!}{R!} = \frac{c(c-1)}{R}$$

Then, summing over all classes leads to the expected number of clusters:

$$\text{Expected \#Clusters} = \sum_{i=1}^{C} \frac{n_i(n_i-1)}{R} \tag{2.1}$$

where n_i is the number of items belonging to the i-th class, C is the number of classes, and R is the number of items in the sequence.

The semantic clustering index used in the CVLT is defined as the ratio between the observed number of clusters in a sequence of R recalls (including also intrusions and repetitions) and the expected value for a random arrangement of R items, computed with (2.1). A value close to 1 means that the observed clusters could be explained by random chance, while a result significantly higher denotes a tendency of subjects to organize items during retrieval.

For the performed experiment, the CVLT semantic clustering index, averaged over 12 trials, amounts to 1.26 ± 0.07 (95% CI is $[1.12, 1.40]$), which suggests a significant amount of clustering.

However, the newer CVLT-II uses a different index, based on a formula from Frender and Doubilet [71]. In this metric, the number of clusters expected solely by chance is given by:

$$\text{Expected \#Clusters (List based)} \equiv (R-1)\frac{m-1}{N_L-1} \tag{2.2}$$

where R is the number of recalled words, this time *without* intrusions and repetitions, $N_L = 64$ is the length of the *presented* list and $m = 32$ is the size of each category (all assumed to be equal). The rationale, as explained in [70], is as follows. Given a recalled item in position i, the next ($i+1$) is of the same category (and thus forms a cluster) with probability $(m-1)/(N_L-1)$, which is independent of the position i. So, the expected number of clusters is obtained by summing the probability of having a cluster at *any* of the $R-1$ possible positions, leading to (2.2). Note that, contrary to (2.1), this formula shifts the focus to the *presented* list.

Moreover, the semantic clustering index used in the CVLT-II is defined as the difference (not the ratio) between the observed number of clusters in a trial and the expected one (2.2). A value close to 0 denotes a random arrangement, while higher values are indicative of organization.

Experimentally, this "list-based" semantic clustering index averages over 12 trials to 4.2 ± 1.0 (95% CI is $[2.30, 5.88]$), which again suggests that participants tend to follow categorical structures when reporting items.

2.3.6 Categorical Asymmetry

Moreover, the categories are not recalled equally. To measure this asymmetry, consider the category of each word correctly recognized/recalled, denoting the two categories with 0 and 1. For instance, for $R = 5$ recalls, this would result in a binary vector $\boldsymbol{c}$ such as:

$$\boldsymbol{c} = (0, 1, 1, 0, 1)$$

If the participants have no bias towards one category or the other, then the number of 1s and 0s in $\boldsymbol{c}$ should be roughly equal.

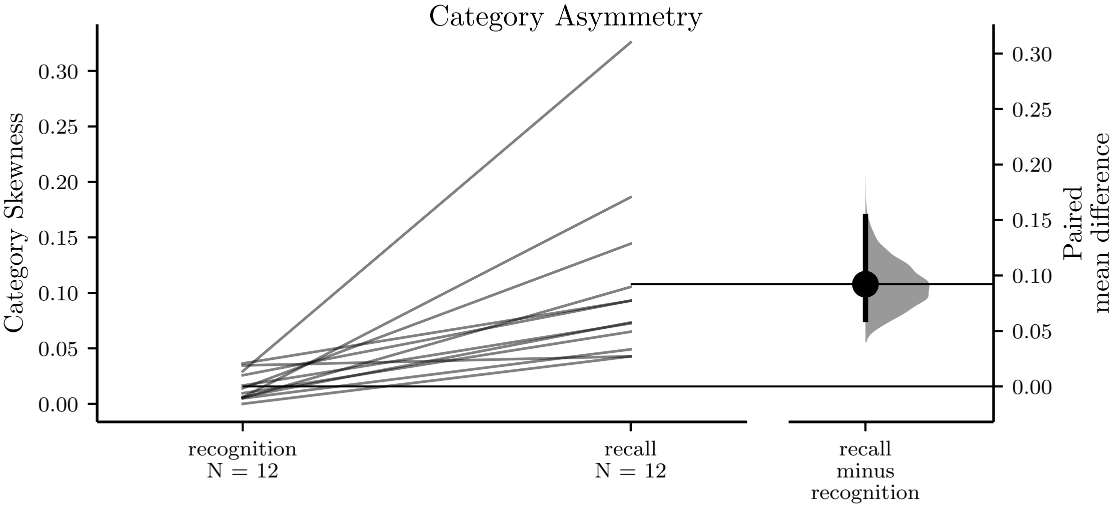


Figure 2.13 – Absolute skewness $|\tilde{\mu}_3|$ of the distribution of categories in recognized/recalled words. The difference in mean is 0.1 (95% CI is $[0.06, 0.15]$, $p < 0.0001$).

If items are considered independent (for simplicity), then the number of 0s follows a binomial distribution with success probability p, which is close to 0.5 if no categorical bias is present. Then, the skewness $\tilde{\mu}_3$ of the binomial distribution can be used to estimate the asymmetry between categories:

$$\tilde{\mu}_3 = \frac{q - p}{\sqrt{npq}}$$

where $q = 1 - p$, and $n = R, M$ is the number of items recalled/retained.

Practically, it can be shown that while $|\tilde{\mu}_3| \approx 0$ for the recognition task, it is significantly higher during recall (fig. 2.13). In other words, even if words from both categories are equally retained, participants tend to recall more words from one category than the other. In most cases (8 out of 12), the category with most recalls is the one corresponding to the first recalled word.

This result suggests that the graph model cannot simply be applied to two categories as if they were distinct lists[5]. Interestingly, such asymmetry is predicted by the graph model when semantic similarities are added: in that case, transitions are more likely to happen towards items of the same category, and so the first category to be visited is likely to be "explored more" than the other. Nonetheless, such a model fails to explain the increase in recall performance.

[5] ^In this case, moreover, it would be difficult to find a good threshold between a *categorized* list and a list of random words.

2.4 Serial Position Curves

The primacy and recency effects are not considered in the graph model, but they are observed in the data.

Fig. 2.14 shows the serial position curves for all participants who performed the free recall task. Note that the words that are presented first are better recalled (primacy effect) and the same happens for the words shown last (recency effect). However, both effects are short-lived, since they depend on the limited STM buffer, which can store on average 7 items [8]. An estimate of their entity is obtained by summing the excess probability of recalling the first/last 7 words, relative to the average retrieval probability for the middle of the list. The result is that, on average, the positional effects are stronger for categorized lists than for random ones, resulting in roughly one more word recalled. However, the strength of primacy and recency effects cannot explain alone the much higher difference in performance found in fig. 2.6.

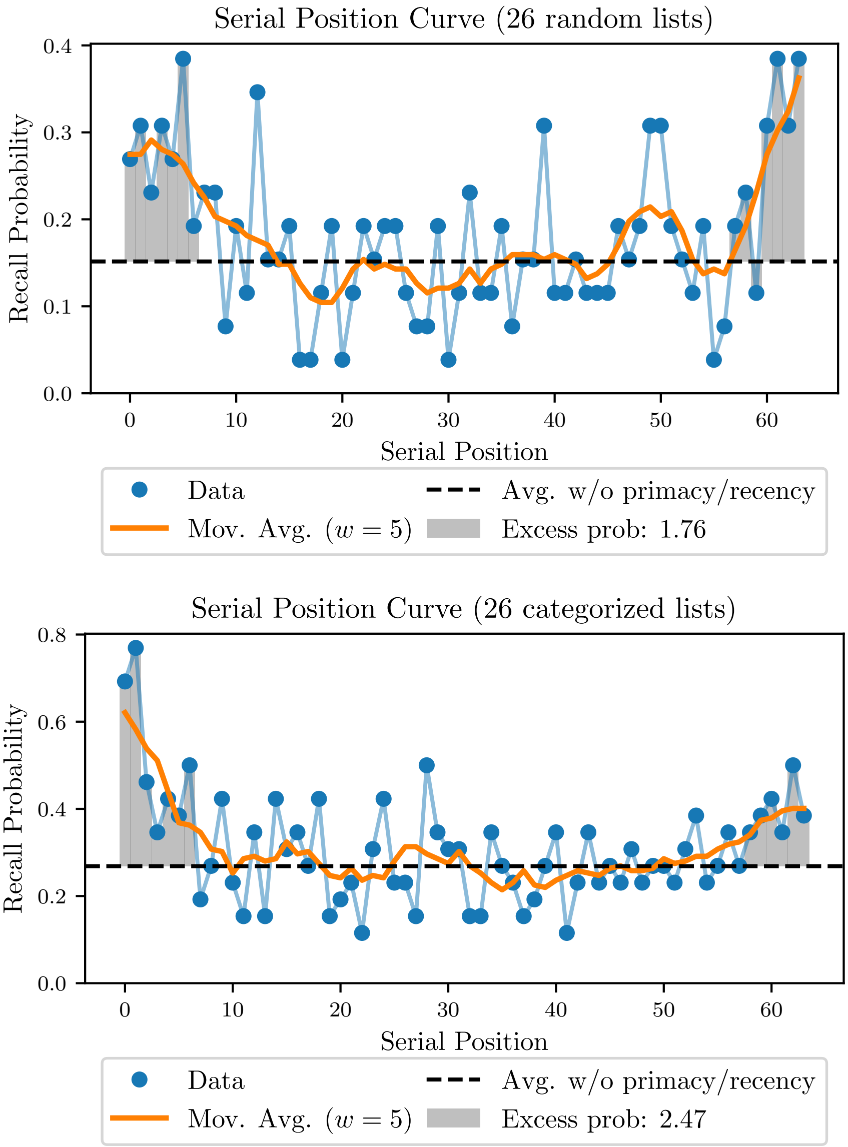


Figure 2.14 – Serial position curves for random/categorized lists. The excess probability is the sum of the probabilities of recalling the first/last 7 items, relative to the average probability of retrieving items in the middle of the list. Data is very noisy due to the limited number of trials. In fact, several positions only appear once in the entire dataset.

2.5 Discussion

An experiment consisting of two memory tasks (free recall and recognition), mirroring the setup from [1], is performed both on *random* and *categorized* lists.

The graph model (sec. 1.5), with no modifications and S from i.i.d. elements, agrees well with the data from the *random* case, thus reproducing the results from fig. 1.9 for the lists with $L = 64$ items.

However, in the case of *categorized* lists, participants tend to recall more words than expected. This cannot be explained by simply adding a semantic similarity term to the matrix S of the graph model.

If words from different categories are treated as two completely separated lists, the graph model gets closer to the data, but it does not fit them well.

Moreover, participants usually recall significantly more words from one category than the other. This result is consistent with a graph model with semantic correlations, but not with a model acting on separated categories.

Thus, neither the original graph model from [1], nor its most natural generalizations to the *categorized* case can fit all obtained data at once.

This conclusion is however not definitive for the following reasons:

- Sample sizes for the experiment are very small. All p-values and confidence intervals were computed with a bootstrap procedure, which should be taken with a grain of salt when data are few.
- Psychology experiments are complex: there are many variables influencing the behavior and performance of participants, and it is not certain that all of them have been accounted for. In particular, it would be interesting to ask the subjects about the strategy they used to memorize the words and complete the experiment. Perhaps, any performance exceeding the one suggested by the “fundamental law of memory recall” [1] is due to some higher level plan that overrides some limits of the basic retrieval processes.

Chapter 3

Conclusions

Human memory is incredibly complex: even in the simplest of settings, when participants freely recall words from a shown list, complex effects (primacy, recency, contiguity) arise from the interplay of many cognitive processes.

However, retrieval tasks also exhibit striking similarities. Particularly, the number of recalled items scales as a power law of the number of presented items, with the coefficients depending on the details of the experiment.

This observation has led researchers to explore new physics-inspired models of recall, based on two "first principles" [5] that bridge biology and cognition: memories are stored as sparse patterns of neural activation, and each of them acts as a *cue* to retrieve new memories during free recall.

A first implementation of these principles can be found in a *neural network model* [2], based on Hopfield networks, in which a cyclic global inhibition signal induces transitions between attractor states, simulating a sequence of recalls. This was then "distilled" in an extremely simple *graph model*, in which memory patterns have independent (but symmetric) similarities with each other, and each transition happens towards the item with the highest similarity that is different from the previously recalled one. In this deterministic model, there is no guarantee that all items stored in memory are effectively recalled, and the number of successful retrievals increases as a power law of the number of retrievable items.

Experimentally, both the number of recalled items R and of recallable items M (estimated as the number of items that can be *recognized* between unrelated alternatives) scale on the number of presented items L depending on the details of the setup (e.g. the rate of presentation). However, when R is plotted against M, curves corresponding to different setups all collapse to the prediction from the *graph model* [1].

This is a remarkable finding, highlighting the existence of a "fundamental law" of memory recall of random words, since the model has no free parameters. As shown in sec. 1.4 and 1.5, such "fundamental law" can be derived from first principles.

However, if the words to be recalled are not chosen randomly, but as belonging to either of two distinct categories, the recall performance increases, while the recognition performance stays the same. As a result, a statistical significant deviation from the proposed "fundamental law" is observed (sec. 2.3.2), albeit for a small sample size.

This outcome cannot be accounted by a difference in other memory effects (such as primacy or recency), nor it can be modelled by adding semantic correlations in the *graph model*, as doing this would predict a lower recall performance, i.e. the opposite of what is observed.

Even if the *graph model* is applied to the words of each category as if they were from entirely different lists, the predictions still do not agree with the experiment. Moreover, categories definitely interact during recall: participants can recognize equally well items from different categories, but recall more items from the category of the first remembered word.

Thus, this work shows how the *graph model* does not reliably predict performance in the case of *categorized lists*, and simple variations cannot fix this estimate. At the same time, however, independent confirmation is given to the predictions of the *graph model* on *random lists* thanks to novel data from a variety of subjects, collected through a newly developed open web app platform.

Our results suggest that the "fundamental law of memory recall" may actually be a lower bound to retrieval processes. When additional structure is available, the brain may exploit it to strategically improve its performance. For instance, words belonging to specific categories may be "compressed": one needs less "mental space" to process two related words than two words sharing no link, which could lead to more words recalled for the *categorized* case. In fact, participants tend to cluster words belonging to the same category during retrieval (sec. 2.3.5), which is indicative of the use of a strategy. Actively organizing words has been shown to improve recall [72], but impair recognition. So, if participants use a strategy only when it improves performance, this could explain why recall is better for *categorized* lists, while the number of retained words remains close to that of *random* lists.

However, note that the data examined in this study is limited only to lists of $L = 64$ words, which are effectively the longest lists that can be produced with just two groups of related words. In particular, this means that the scaling of recall performance for *categorized* lists has not been measured, and it is left to a future study over different lengths. Note that shorter lists can be generated by simply sampling from the set of terms that were collected in sec. 2.2.1.

Another interesting test would be to generate "mixed" lists, by taking a proportion of random and categorized terms. This would allow quantifying how recall performance varies depending on the strength of semantic correlations. Perhaps, such fine-grained data could suggest a way to generalize the *graph model* to the case of related items.

Finally, in future studies, attention should be paid to the recall strategies enacted by participants during the experiment, which may be collected by directly asking them for feedback at the end of the recall task.

In fact, progress on the study of memory depends not only on understanding pure retrieval dynamics, but also the more advanced control mechanisms that the human brain can activate to augment its performance.

Bibliography

[1] Michelangelo Naim et al. “Fundamental Law of Memory Recall”. *Physical Review Letters* 124.1 (Jan. 2020), p. 018101.

[2] Sandro Romani et al. “Scaling Laws of Associative Memory Retrieval”. *Neural Computation* 25.10 (Oct. 2013), pp. 2523–2544.

[3] Mikhail Katkov, Sandro Romani, and Misha Tsodyks. “Word length effect in free recall of randomly assembled word lists”. *Frontiers in Computational Neuroscience* 8 (2014), p. 129.

[4] Mikhail Katkov, Sandro Romani, and Misha Tsodyks. “Effects of long-term representations on free recall of unrelated words”. *Learning & Memory* 22.2 (Feb. 2015), pp. 101–108.

[5] M. Katkov, S. Romani, and M. Tsodyks. “Memory Retrieval from First Principles”. en. *Neuron* 94.5 (June 2017), pp. 1027–1032.

[6] Alan Baddeley, Michael W. Eysenck, and Michael C. Anderson. *Memory*. 3rd ed. London: Routledge, Mar. 2020.

[7] G. Sperling. “A model for visual memory tasks”. eng. *Human Factors* 5 (Feb. 1963), pp. 19–31.

[8] George A. Miller. “The magical number seven, plus or minus two: Some limits on our capacity for processing information”. *Psychological Review* 63.2 (1956). Place: US Publisher: American Psychological Association, pp. 81–97.

[9] Lionel Standing. “Learning 10000 pictures”. en. *Quarterly Journal of Experimental Psychology* 25.2 (May 1973). Publisher: SAGE Publications, pp. 207–222.

[10] John R. Anderson. “A spreading activation theory of memory”. *Journal of Verbal Learning & Verbal Behavior* 22.3 (1983). Place: Netherlands Publisher: Elsevier Science, pp. 261–295.

[11] M. Karl Healey, Patrick Crutchley, and Michael J. Kahana. “Individual differences in memory search and their relation to intelligence”. eng. *Journal of Experimental Psychology. General* 143.4 (Aug. 2014), pp. 1553–1569.

[12] Murdock and B. Bennet. “The serial position effect of free recall” (1962).

[13] N. C. Waugh and T. R. Anders. “Free recall of very slowly presented items” (1969).

[14] M. J. Kahana. “Associative retrieval processes in free recall”. eng. *Memory & Cognition* 24.1 (Jan. 1996), pp. 103–109.

[15] Endel Tulving and Zena Pearlstone. “Availability versus accessibility of information in memory for words”. *Journal of Verbal Learning & Verbal Behavior* 5.4 (1966). Place: Netherlands Publisher: Elsevier Science, pp. 381–391.

[16] Roger Brown and David McNeill. “The “tip of the tongue” phenomenon”. en. *Journal of Verbal Learning and Verbal Behavior* 5.4 (Aug. 1966), pp. 325–337.

[17] D. J. Murray. “Semantically cued retrieval of words from long-term memory”. en. *Bulletin of the Psychonomic Society* 5.2 (Feb. 1975), pp. 134–136.

[18] D. J. Murray. “Graphemically cued retrieval of words from long-term memory”. *Journal of Experimental Psychology: Human Learning and Memory* 1.1 (1975). Place: US Publisher: American Psychological Association, pp. 65–70.

[19] D. J. Murray, Carol Pye, and W. E. Hockley. “Standing’s power function in long-term memory”. en. *Psychological Research* 38.4 (Dec. 1976), pp. 319–331.

[20] M. E. Hasselmo and B. P. Wyble. “Free recall and recognition in a network model of the hippocampus: simulating effects of scopolamine on human memory function”. eng. *Behavioural Brain Research* 89.1-2 (Dec. 1997), pp. 1–34.

[21] Jeroen G. W. Raaijmakers and Richard M. Shiffrin. “SAM: A Theory of Probabilistic Search of Associative Memory”. en. *Psychology of Learning and Motivation*. Ed. by Gordon H. Bower. Vol. 14. Academic Press, Jan. 1980, pp. 207–262.

[22] Eugen Tarnow. “There is no capacity limited buffer in the Murdock (1962) free recall data”. *Cognitive Neurodynamics* 4.4 (Dec. 2010), pp. 395–397.

[23] Marc W. Howard and Michael J. Kahana. “A Distributed Representation of Temporal Context”. en. *Journal of Mathematical Psychology* 46.3 (June 2002), pp. 269–299.

[24] Sean M. Polyn, Kenneth A. Norman, and Michael J. Kahana. “A context maintenance and retrieval model of organizational processes in free recall”. *Psychological review* 116.1 (Jan. 2009), pp. 129–156.

[25] Gordon D. A. Brown, Ian Neath, and Nick Chater. “A temporal ratio model of memory”. eng. *Psychological Review* 114.3 (July 2007), pp. 539–576.

[26] G. D. Brown, T. Preece, and C. Hulme. “Oscillator-based memory for serial order”. eng. *Psychological Review* 107.1 (Jan. 2000), pp. 127–181.

[27] Eugen Tarnow. *How to Kill a Computer Model of Short Term Memory in Psychological Review Part I: Separate Fittings of Experimental Data*.

[28] Douglas L. Hintzman. “Research Strategy in the Study of Memory: Fads, Fallacies, and the Search for the “Coordinates of Truth””. en. *Perspectives on Psychological Science* 6.3 (May 2011). Publisher: SAGE Publications Inc, pp. 253–271.

[29] J. J. Hopfield. “Neural networks and physical systems with emergent collective computational abilities”. en. *Proceedings of the National Academy of Sciences* 79.8 (Apr. 1982). Publisher: National Academy of Sciences Section: Research Article, pp. 2554–2558.

[30] R. McEliece et al. “The capacity of the Hopfield associative memory”. *IEEE Transactions on Information Theory* 33.4 (July 1987). Conference Name: IEEE Transactions on Information Theory, pp. 461–482.

[31] Kwan F. Cheung, Les E. Atlas, and Robert J. Marks. “Synchronous vs asynchronous behavior of Hopfield’s CAM neural net”. EN. *Applied Optics* 26.22 (Nov. 1987). Publisher: Optical Society of America, pp. 4808–4813.

[32] M.v. Feigelman and L.b. Ioffe. “The augmented models of associative memory asymmetric interaction and hierarchy of patterns”. *International Journal of Modern Physics B* 01.01 (Apr. 1987). Publisher: World Scientific Publishing Co., pp. 51–68.

[33] D. O. Hebb. *The organization of behavior; a neuropsychological theory*. The organization of behavior; a neuropsychological theory. Pages: xix, 335. Oxford, England: Wiley, 1949.

[34] Daniel J Amit, Hanoch Gutfreund, and H Sompolinsky. “Statistical mechanics of neural networks near saturation”. en. *Annals of Physics* 173.1 (Jan. 1987), pp. 30–67.

[35] Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. “Spin-glass models of neural networks”. *Physical Review A* 32.2 (Aug. 1985). Publisher: American Physical Society, pp. 1007–1018.

[36] R. Quian Quiroga and G. Kreiman. “Measuring sparseness in the brain: Comment on”. *Psychological review* 117.1 (Jan. 2010), pp. 291–297.

[37] M. V. Tsodyks and M. V. Feigel\textquotesingleman. "The Enhanced Storage Capacity in Neural Networks with Low Activity Level". en. *Europhysics Letters (EPL)* 6.2 (May 1988). Publisher: IOP Publishing, pp. 101–105.

[38] E. Gardner. "The space of interactions in neural network models". en. *Journal of Physics A: Mathematical and General* 21.1 (Jan. 1988). Publisher: IOP Publishing, pp. 257–270.

[39] Franklin Zaromb et al. "Temporal associations and prior-list intrusions in free recall." *Journal of experimental psychology. Learning, memory, and cognition* (2006).

[40] Timothy A. Salthouse and Karen L. Siedlecki. "An individual difference analysis of false recognition". *The American journal of psychology* 120.3 (2007), pp. 429–458.

[41] Paul Muter. "Recognition failure of recallable words in semantic memory". en. *Memory & Cognition* 6.1 (Jan. 1978), pp. 9–12.

[42] Stefano Recanatesi et al. "Neural Network Model of Memory Retrieval". English. *Frontiers in Computational Neuroscience* 9 (2015). Publisher: Frontiers.

[43] Henry Dale. "Pharmacology and Nerve-endings (Walter Ernest Dixon Memorial Lecture)". *Proceedings of the Royal Society of Medicine* 28.3 (Jan. 1935), pp. 319–332.

[44] Michael J. Kahana. "The cognitive correlates of human brain oscillations". eng. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* 26.6 (Feb. 2006), pp. 1669–1672.

[45] Daria Osipova et al. "Theta and Gamma Oscillations Predict Encoding and Retrieval of Declarative Memory". en. *Journal of Neuroscience* 26.28 (July 2006). Publisher: Society for Neuroscience Section: Articles, pp. 7523–7531.

[46] Sandro Romani, Mikhail Katkov, and Misha Tsodyks. "Practice makes perfect in memory recall". *Learning & Memory* 23.4 (Apr. 2016), pp. 169–173.

[47] Jonathan F. Miller, C. Weidemann, and M. Kahana. "Recall termination in free recall". *Memory & cognition* (2012).

[48] Jorge Gracia and Eduardo Mena. "Web-Based Measure of Semantic Relatedness". en. *Web Information Systems Engineering - WISE 2008*. Ed. by James Bailey et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 136–150.

[49] George A. Miller. "WordNet: A Lexical Database for English". *Communications of the Acm* 38 (1995), pp. 39–41.

[50] Zellig S. Harris. "Distributional structure". *Word* 10 (1954), pp. 146–162.

[51] T. Landauer and S. Dumais. "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge." (1997).

[52] Marc W Howard and Michael J Kahana. "When Does Semantic Similarity Help Episodic Retrieval?" en. *Journal of Memory and Language* 46.1 (Jan. 2002). Number: 1, pp. 85–98.

[53] A. D. Baddeley. "The influence of acoustic and semantic similarity on long-term memory for word sequences". *Quarterly Journal of Experimental Psychology* 18.4 (Nov. 1966), pp. 302–309.

[54] W. A. Bousfield. "The occurrence of clustering in the recall of randomly arranged associates". *Journal of General Psychology* 49 (1953). Place: US Publisher: Heldref Publications, pp. 229–240.

[55] John A. Robinson. "Category Clustering in Free Recall". *The Journal of Psychology* 62.2 (Mar. 1966), pp. 279–285.

[56] F. Gobet et al. "Chunking mechanisms in human learning". eng. *Trends in Cognitive Sciences* 5.6 (June 2001), pp. 236–243.

[57] Robyn Speer et al. *LuminosoInsight/wordfreq: v2.2*. Oct. 2018.

[58] Marc Brysbaert, Paweł Mandera, and Emmanuel Keuleers. “The Word Frequency Effect in Word Processing: An Updated Review”. en. *Current Directions in Psychological Science* 27.1 (Feb. 2018). Publisher: SAGE Publications Inc, pp. 45–50.

[59] Lynn J. Lohnas and Michael J. Kahana. “Parametric effects of word frequency in memory for mixed frequency lists”. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 39.6 (2013). Place: US Publisher: American Psychological Association, pp. 1943–1946.

[60] Maria Montefinese et al. “Semantic memory: A feature-based analysis and new norms for Italian”. *Behavior research methods* 45 (Oct. 2012).

[61] Robyn Speer, Joshua Chin, and Catherine Havasi. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”. 2017, pp. 4444–4451.

[62] Yingfan Wang et al. “Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization”. *arXiv:2012.04456 [cs, stat]* (Aug. 2021). arXiv: 2012.04456.

[63] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* 5.1 (Jan. 1967). Publisher: University of California Press, pp. 281–298.

[64] Simon De Deyne et al. “The “Small World of Words” English word association norms for over 12,000 cue words”. en. *Behavior Research Methods* 51.3 (June 2019). Number: 3, pp. 987–1006.

[65] Álvaro Corral, Gemma Boleda, and Ramon Ferrer-i-Cancho. “Zipf’s Law for Word Frequencies: Word Forms versus Lemmas in Long Texts”. en. *PLOS ONE* 10.7 (2015). Publisher: Public Library of Science, e0129031.

[66] Adriana Roventini, Rita Marinelli, and Francesca Bertagna. “ItalWordNet v.2”. ita. *http://www.ilc.cnr.it/it/content/italwordnet* (Dec. 2016). Accepted: 2016-12-02T20:52:23Z Publisher: Istituto di Linguistica Computazionale “A. Zampolli” - Consiglio Nazionale delle Ricerche (ILC-CNR).

[67] M J Gardner and D G Altman. “Confidence intervals rather than P values: estimation rather than hypothesis testing.” *British Medical Journal (Clinical research ed.)* 292.6522 (Mar. 1986), pp. 746–750.

[68] Joses Ho et al. “Moving beyond P values: data analysis with estimation graphics”. en. *Nature Methods* 16.7 (July 2019), pp. 565–566.

[69] Bradley Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. en. Google-Books-ID: gLlpIUxRntoC. CRC Press, May 1994.

[70] John L. Stricker et al. “New semantic and serial clustering indices for the California Verbal Learning Test-Second Edition: background, rationale, and formulae”. eng. *Journal of the International Neuropsychological Society: JINS* 8.3 (Mar. 2002), pp. 425–435.

[71] Robert Frender and Peter Doubilet. “More on measures of category clustering in free recall-although probably not the last word”. *Psychological Bulletin* 81.1 (1974). Place: US Publisher: American Psychological Association, pp. 64–66.

[72] Scott A. Guerin and Michael B. Miller. “Semantic organization of study materials has opposite effects on recognition and recall”. eng. *Psychonomic Bulletin & Review* 15.2 (Apr. 2008), pp. 302–308.