POLITECNICO DI TORINO

Master's Degree in Physics of Complex Systems

Politecnico
di Torino

Master's Degree Thesis

Minimal parsimonious chunking of written language: investigating the storage-computation trade-off as a driving principle in chunk formation

Supervisors

Prof. ALESSANDRO PELIZZOLA

Prof. DAVIDE CREPALDI

Dr. ROMAIN BRASSELET

Candidate

FRANCESCA DI GIOVANNI

April 2021

Abstract

Visual word identification is the process that allows the brain to recognize a familiar and meaningful word from an ordered collection of letters. Chunking seems to play an important role in this process: rather than jumping from single letters to words, the brain seems to group letters in smaller units. Some recent studies suggest to oust morphemes, the smallest meaning-bearing units in language, from their role as building blocks in chunking: they should instead be replaced by letter chunks which do not necessarily have an explicit connection with semantics, but which could be explained by statistical regularities in letter co-occurence. However, the exact principles according to which these chunks emerge in skilled readers are still unclear.

The algorithm developed in the thesis tries to answer this question, looking for the set of chunks that optimizes the trade-off between the storage of many different units and the computational effort needed to process completely new words. This optimization problem is formally translated in the minimization of a one-parameter function featuring two competing terms, the number of stored chunks and the average number of chunks per word. The parameter in the objective function allows us to adjust the relative weight of the two players in this competition, and potentially mirrors psychologically meaningful phenomena, such as the progressive mastering of literacy. Since we used a massive database to learn the chunks, many computational tricks are introduced to accelerate the algorithm, which otherwise would not be able to compute a solution in a finite time. A natural improvement of the algorithm is then to assign different weights to the chunks. We considered here the concept of "chunk productivity", which we defined as the number of times that a chunk is used to identify a word – although this concept turned out to be fairly elusive. Finally, in order to evaluate the algorithm's performance against real psychological data, we tested its ability to account for priming, the time saving in the identification of a target word (e.g., deal) that is brought about by a related one (the prime, e.g., dealer). The core idea is that priming is larger when the prime is chunked onto its target (dealer=deal+er). This naïve reader, whose only goal is to find the best compromise between storage and computation without any semantic

or morphological information, surprisingly proves able to select many interesting affixes and chunks. Nevertheless, it only partially accounts for human performance. This can be considered a further hint supporting the hypothesis that chunks could partially emerge in a language-independent mechanism, which could take into account, among other factors, the computation and storage trade-off.

Table of Contents

List of Tables

List of Figures

Chapter 1

Setting the purpose

Chapter 1 reviews the most significant evidence in favour of a chunking approach applied to visual word recognition, explaining which are the consequences of considering reading as a visual endeavour, and how statistical learning comes into play. From the discussion, it will be natural to ask which are the exact mechanisms behind the formation of higher-order orthographic units, the chunks, which group together particularly cohesive bunches of letters, and this sets the purpose of the thesis.

1.1 A neuronal-recycling code for written words

Written language is a relatively **recent** invention: Woods marks its birth about 5,5 thousand years ago [1], while anatomically modern humans have existed for about one hundred thousand years. It is therefore natural to believe that reading and writing are **not** part of our biological endowment: written language is in fact still not universally spread among human communities, and it cannot be acquired spontaneously and without an explicit teaching. Nevertheless, it is impressive how **efficiently** and effortlessly we process information by reading: for example, it has been estimated that the average silent reading rate for adults in English is about 250 word per minute [2]. Finding an explanation behind this apparent contradiction could represent a significant step in the direction of understanding general human learning capacities.

According to several authors from different fields [e.g., 3], who all share the "standard social science model" even if differently declined, reading, as well as the other human

cultural skills, must have arisen from the emergence of a flexible domain-general learning capacity: thanks to its plasticity, the human brain would be a sort of 'tabula rasa', capable of absorbing essentially any form of culture, without any limitation due to its biological architecture . A more convincing theory, the *Neuronal Recycling Hypothesis* [4], suggests to take into account the **constraints** that our prior evolution and brain organization impose on novel mental capacities, due to the much shorter **timescale** in which they develop: human genome evolution cannot have been influenced by as recent and culturally variable activities as reading. There is indeed strong evidence that the mechanism behind the acquisition of new cultural capacities is the *neuronal recycling* process, by which the novel tasks invade a pre-existing brain system, and reorient a fraction of its neural resources to a different use. In particular, many experiments [5] showed that several left-hemispheric regions are reproducibly activated during word reading; one region among the others, located in the left occipito-temporal sulcus, appears to be involved in written word recognition but is not shared with spoken language processing (at least when subjects listen passively to spoken words [6]), and the same one is also activated by other categories of **visual** objects such as faces or objects. This region has been defined *visual word form area* (VWFA): the name does not imply that it is entirely dedicated to reading, but just that it is the cortical sector recycled by visual word recognition processes.

1.2 Orthographic processing is not a predominantly linguistic skill

Visual word recognition is just the first step of the reading process, that includes accessing the sounds (phonology) and the meanings (semantics) of words. The complexity of the process, that has no clear boundaries between the different phases, and that is supposed to rely on mappings between orthographic, semantic and phonological codes [7], has led the majority of earlier research on visual word recognition to focus mainly on how letter-level information maps onto higher-level linguistic properties (phonological and semantic) [8], downplaying the status of printed words as visual objects. Orthographic processing was considered as an extension of already established linguistic skills in spoken language processing, since most children start learning to read when they have

already developed a sophisticated system for the recognition and production of speech. The mapping between individual letters or graphemes, letter clusters, and phonemes, the elementary units of spoken language, was considered to be a necessary step in processing written words, as well as letter-to-meaning associations. In general, it seemed that written language acquisition would make mainly use of the regularities in the mappings between different levels of linguistic information.

The recycling hypothesis has instead recently prompted research to focus on the status of written words as **visual inputs**. Many experiments have indeed confirmed the idea that orthographic processing can operate in the absence of prior linguistic knowledge, without phonological and semantic hints: not only baboons [9], but also pigeons, whose visual system is very different from the one of primates, proved to be able to process orthographic regularities [10]; it was indeed demonstrated that pigeons trained to discriminate words from non-words are able to extract the orthographic properties that define words and to use them to distinguish new unseen English words from comparable letter strings. We can then conclude that reading processing in its earliest stages of visual word recognition can be actually performed by domain-general visual mechanisms, since the process is present also in non-linguistic animals.

1.3 Hierarchical coding of letter strings: where visual neuroscience meets linguistic morphology

Research in the field of visual neuroscience has largely demonstrated that the brain recognizes objects via a feedforward hierarchy of computational layers, whose units become sensitive to increasingly larger and more complex objects [11, 12, 13]. Following the hypothesis that written word recognition could share the same mechanisms that lie behind the recognition of any combinations of visual objects and features, we can expect to have a similar hierarchical organization for orthographic processing: using fMRI, Vinckier et al. [14] showed that the visual word-form system presents indeed a gradient of increased sensitivity to larger and higher-level components of words. And this leads to theorising the existence of intermediate objects between letters and words, precisely on the basis of the nature of written words as visual objects processed by the visual word form area.

The same idea emerges in an apparently different literature, belonging to the field of linguistic morphology, that studies the internal structure of words.

The *multiple level hypothesis*, that embodies the notion that the orthographic code has multiple levels of representation arranged in a hierarchical order, has always been the core assumption of most models of word processing [15, among the others]. However, there is still no consensus on which orthographic units characterise the different levels of the orthographic code. At the base of the hierarchy, we find the building blocks of orthographic representations: the most obvious hypothesis is that these are represented simply by the individual letters, but various visual word recognition models consider instead as elementary codes the **graphemes**. These are the smallest graphic units that translate sounds into written language, and they arise from the mapping of the phonological code, normally already learnt by the readers, in the orthographic code [8]. The highest level of the hierarchy represents the interface with semantic representation; also in this case, it is not clear which units are involved: do different inflections of a word (*cat-cats*; *fall-fell*) share the same representation? And what about the derived words, where both the meaning and the grammatical class could change (e.g., *dark* (adjective) and *darkness* (noun); *angel* and *angelic* (that has a broader meaning))? Between the two extremes of the hierarchy, there should exist a level where **morphemes** play a role [16]. Every word is made up of one or more morphemes, that are the smallest meaning-bearing units in a language [17]. They can be subdivided into free morphemes, if they can stand alone (e.g., *dog*, *play*) and bound morphemes, if they have to be used with a free morpheme to form a word (*-s* in *dogs*, *-er* in *player*). Several word recognition experiments have actually shown the existence of morphological effects in the processing of morphologically complex words, but it is not clear which mechanisms underlie the decomposition. The dominant belief for a long time has been that the importance of morphemes in processing is linked to their nature of units carrying a meaning: they represent indeed "islands of regularities" in the human language [18], which is otherwise characterized by arbitrariness in meaning-form mapping, since they allow us to create new words, or to grasp the meaning of unknown words (if a worker is someone who works, a "decider" will be someone who decides). As a consequence, morphological decomposition was thought to occur only for complex words related in meaning to their stems (e.g., *worker=work+er*, but *corner* should not be decomposed

into *corn+er*). Again, the units characterizing the orthographic code are seen as the product of a mapping, this time from the semantic code, which, like the phonological one, is generally already well established when an individual approaches reading.

Therefore, both the fact that the orthographic code emerges when phonological and semantic codes are already part of the knowledge of the developing readers, and more generally the relatively recent birth of written language, have always led to overlook the possibility that the units of different orthographic levels could be created and organised in an linguistic-independent process, based on the regularities of the visual input per se.

We will see that some recent masked priming experiments invited instead to consider this possibility, starting from the evidence that morphemes prove to be relevant in a meaning-independent decomposition, defined as *morpho-orthographic segmentation*.

1.4 A new experimental paradigm leads to theorizing a morpho-ortographic segmentation

Written word priming studies represent an experimental paradigm aimed at investigating the structure of the mental lexicon and the way written words are processed by the readers [see 19, 20, for a review]. In general, participants are asked to perform a task involving the reading of a word, the target, and researches measure their response time (the time required to execute the task) when the target is preceded by the presentation of another word, the prime: if the presentation of a certain prime reduces the reaction time (RT, in milliseconds) with respect to a baseline, determined by presenting an unrelated prime, this means that the processing of the prime helps to process the target and complete the task, and that there is some kind of relationship between the prime and the target. The tasks can be different, depending on the objective of the study and on the stages of word processing the study intends to investigate: when reading aloud the target, participants are forced to map the orthographic code into the phonological one; a semantic categorization of the inputs involves instead the access to the word meaning; identifying the presence in a string of a target letter, or deciding if it is a real word or a non-word (lexical decision task) are often used to isolate the orthographic processing. Also the prime-target pairs are chosen specifically to investigate different aspects of the organisation of the mental lexicon, i.e. how we mentally organize and

represent words, or of the way in which we access it: reading the word *cat* helps us reading the target *dog* more than *car*? Is the target *dialog* processed faster if it is preceded by the presentation of the prime *dial*, because of their orthographic overlap, compared to the presentation of an unrelated word *lately*?

The presentation time of the prime is also a critical variable to be taken into account: if the prime remains visible for a long time, participants might be led to start thinking how to solve the task more effectively, introducing higher-level, strategic factors that do not belong to spontaneous processing. To avoid this effect, masked priming experiments can be used, where a lower-case prime is presented for a very short time interval (e.g., 50 ms), followed and/or preceded by pattern masks (e.g., #####), and by an upper-case target. As a result of both the masking and the short prime presentation time, in most trials the participants show no conscious experience of the prime, that is they do not report having seen it, and this allows to exclude the influence of strategy-based processes, while the priming effect is still present.

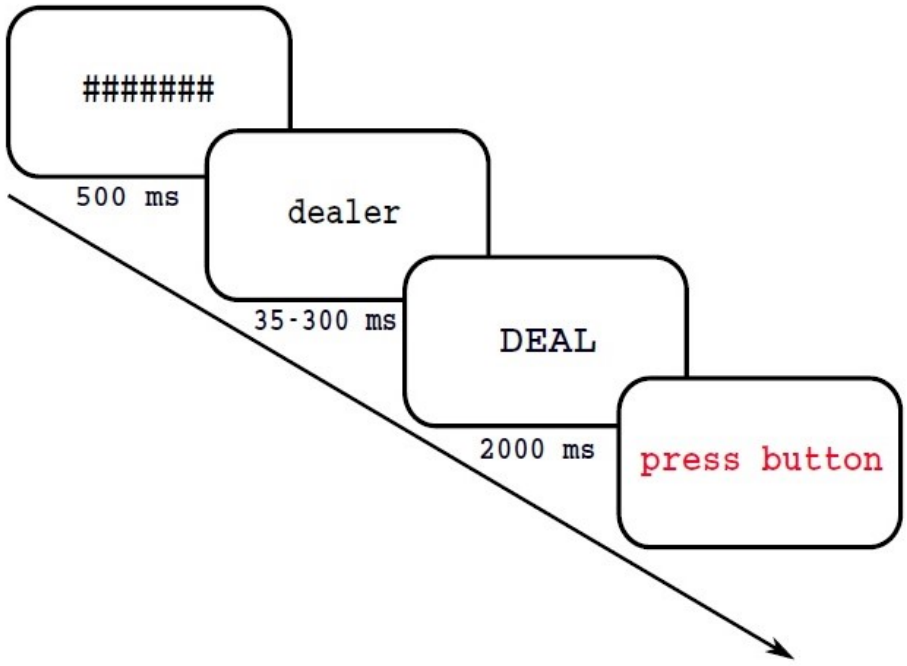

Figure 1.1: Priming experiment scheme

Masked priming experiments are one of the most common experimental paradigm used to explore how morphological relationships between words are processed by the readers; in the related condition, the prime is a morphologically complex word (e.g. cleaner), and the target contains the stem (e.g., CLEAN). The proposed task is often a lexical decision task: participants have to decide as quickly as possible if the target is a word or a non-word, pressing one of two buttons. Here, a reduction in the reaction time would suggest that the recognition of complex words involves an effective morphological decomposition: the morphological analysis of words would not be only a theoretical linguistic approach to study complex words, but the actual strategy readers use to process them. This facilitation has been confirmed by many experiments for over 40 years (e.g., [21]), and it cannot be explained by summed effects of semantic priming (the kind of priming observed for the pairs like cello-VIOLIN) and orthographic priming (e.g., electrode-ELECT) [22].

Precisely in the context of morphological masked priming, effects are observed which the semantically-based theories of morphological decomposition cannot explain: semantic information may not play a role in the morphological decomposition. The first studies highlighting the unexpected phenomenon are the ones by Longtin et al. [23] and by Rastle et al. [24]. Considering the latter, the targets, simple words, where preceded by masked primes in three different conditions [see 3.10.3 for more examples]:

1. in the **transparent** condition, prime and target are morphologically and semantically related (e.g., dealer-DEAL)

2. in the **opaque** condition, prime shares with the target a morphological relationship that is only apparent, since they are not semantically related (e.g., corner-CORN: here, *-er* is not a suffix and *corn* is not a root)

3. in the **form** condition, prime and target only share an orthographic overlap (e.g., scandal-SCAN).

Surprisingly enough, results showed that priming effects were significant and nearly equivalent for primes and targets in the transparent and opaque condition, and statistically distinguishable from priming effects for pairs in non-morphological form condition. These findings suggest that readers tend to identify morphological structure also in

words that only have an orthographic appearance of being complex, but whose meaning is completely unrelated to the meaning of the pseudo-morphemes they include.

These two experiments were the forerunners for a series of subsequent studies [25, 26], which confirmed that word processing involves this form of decomposition, defined as *morpho-orthographic chunking*. The main features of this processing can be summarised as follows [20]:

- it is applied to all morphologically structured stimuli, even when the morpheme does not carry its particular meaning (e. g., corner-CORN), and even for pseudowords (e. g., darkism-DARK) [27].
- it cannot be due to a simple orthographic overlap, since a priming effect is not observed for pairs like canalast-CANAL.
- it arises early in visual word recognition, since the phenomenon is not observed anymore if prime duration is increased.

1.5 Morphemes as letter chunks: investigating the statistical principles involved in morpho-orthographic chunking

If it is true that the masked priming experiments show that morphemes certainly play a role in the decomposition of written words in the early stages of visual word recognition [16], the features of the morpho-orthographic process identified before seem to indicate that this phenomenon occurs not because of the nature of morphemes as meaning-bearing units, but maybe because they represent elements of regularity in the written language in its status as visual input. In fact, precisely because they play a fundamental role in the from-to-meaning mapping, i.e. because they carry the same meaning or play the same role, at least in a broad sense, in all the words in which they appear, morphemes naturally represent recurrent letter chunks, which could be easy to identify from a statistical point of view [28], to the point of being recognized even when they do not play any morphological role (as in the corner-CORN priming effect). Indeed, it is not a novelty that the human brain is inclined to identify regularities in the environment [see 29, for a review], in order to decrease uncertainty, organize the

information load, and anticipate events [30], and the same sensitivity to orthographic regularities seems likely to be one of principles that guides the processing of written words [31]. As such, we can expect other letter chunks besides morphemes to perform the same function as intermediate units in the decomposition, albeit they are not directly involved in form-to-meaning mapping. Similarly, despite their semantic origin, morphemes themselves could be identified based on visual-orthographic information alone, with a language-agnostic mechanism that captures regularities in the written words.

A clever way to test this hypothesis is to resort to artificial languages: this allows to completely remove the linguistic knowledge (semantic, phonological or syntactic information) of the participants, in order to study the possibility that morphemes emerge even without it. The study conducted by Lelonkiewicz et al. [32] exploited this experimental paradigm: using pseudoletter strings characterised by the presence of the same clusters of characters across different strings, they showed that, after a brief familiarisation with the strings, participants were more likely to classify as words belonging to the pseudolanguage those containing one of these chunks.

However, it remains to be understood which statistical principles guide morpho-orthographic chunking. Since on the one hand the problem is new in these terms (previous computational models explained morphological processing as the result of another type of statistical regularities, those in **form-to-meaning mapping** [33]), and on the other hand it has only recently generated new interest in its connection with all the other fields of human learning based on the identification of statistical regularities (all the computational models that learn the form-to-meaning mapping for morphologically complex words have to be fed by **pre-segmented** input [34]), the cognitive literature still has no answer.

The chunking mechanism seems to characterise many different processes of perception, learning and cognition [35]: understanding its principles is therefore of interest in a much wider range of fields than just the literature on reading, and at the same time written language, due to its nature as an artificial product, can be more easily controlled and tested. An analogous problem is indeed faced in the study of speech segmentation: how do infants become able to segment a continuous speech stream into words without clear physical boundaries? Experimental results [e.g., 36] showed that adults were able

to segment an artificial language, perceived in a continuous flow, without any prosodic or phonological cue: therefore, also in this case, the exploitation of statistical regularities seems to be fundamental in the identification of words. The principles considered in the elaboration of computational models of speech segmentation are essentially based on the frequency of letter sequences: if high, they are grouped together to form a single unit; if low, they are more likely to correspond to word boundaries [37]. The same idea can be applied to morpho-orthographic segmentation.

1.6 The formation of chunks in developing readers

Another field of research that can allow us to understand the principles that drive chunk formation is the study of **reading acquisition** in children. Compared to other human capacities, such as the more general visual object recognition, the advantage in the study of reading-related phenomena is precisely the fact that it is not an innate ability and it is actually learned quite late in human development, only by explicit teaching, with a proficiency that increases with exercise and practice. It may therefore not be difficult to follow the evolution of children's reading skills in order to understand how chunks are formed, how they evolve, and which functions they play in word processing. In particular, proposals have been made that, exactly as for the recognition of other visual objects, also in the case of reading the construction of higher-level orthographic units on the basis of statistical principles could be the mechanism that makes developing readers efficient, much faster and almost error-free. The same chunking mechanism that children use to create larger and more complex visual shapes (e.g., objects, faces), starting from elementary simple shapes (e.g., lines) and taking advantage of pattern regularities, could be used to form higher-order chunks of letters during reading acquisition: the slow and laborious conversion of many single units (letters or graphemes), typical of children approaching reading, would become a quick and efficient process based on larger units (larger n-grams, morphemes, words), that could be formed inferring the distribution of letter patterns that generally give written languages a great redundancy (only a small percentage of the possible letter combinations are actual entries in the lexicon [38]). The statistical learning principle could therefore be the pre-existing cognitive mechanism of the visual system that reading takes over and that could explain our efficiency as

readers, although reading cannot be considered part of our biological endowment.

A confirmation of this hypothesis is found in the study of Developmental Dyslexia: the difficulty and slowness in reading for dyslexic patients could be caused by the absence of whole-word representations, which force them to rely on smaller units, paying the price of a greater computational effort. In the study conducted by Burani et al. [39], it is shown that both dyslexics and younger readers are facilitated in reading aloud derived words, made up of roots and suffixes, with respect to simple words (reading aloud involves, in addition to written word processing, also a mapping from orthographic to phonological code), as opposed to skilled and adult readers, for whom instead the facilitation due to the presence of morphemes occurs only when they are included in pseudo-words: this suggests that intermediate units - in this case the morphemes - are exploited in word processing when there are no higher-level units, i.e. a whole-word representation - so in children and dyslexics even in the case of real words, in proficient readers only for invented and therefore unknown words. A regression to letter-by-letter processing is also observed after brain damage in previously proficient readers [e.g. 40]: the Letter-by-letter (LBL) dyslexia is indeed characterized by a large increase in word processing time as a function of the number of letters the words contain, an effect that is not observed in intact adult readers, and that precisely suggests that word recognition for LBL dyslexic readers proceeds by the slow, sequential identification of individual letters. In a more specific context, that can be however reinterpreted in the statistical learning landscape, it was demonstrated that children start using morphemes as functional units in the course of reading development, although it remains almost completely unclear how and when they become sensitive to morphology. A study by Hasenäcker et al. [41] on German children in elementary school, from Grade 2 to Grade 6, reveals that this happens at a very early age, with differences depending on vocabulary knowledge and the type of morphological relationship. According to the hypothesis we support, the sensitivity to morphemes may actually involve chunks of a more general nature, but the study can be seen as a confirmation of the idea that developing readers tend to form longer chunks extracting statistical information from the reading material, and that in this way they improve their proficiency.

1.7 Investigating the role of the storage-computation trade-off in chunk formation

It is precisely in this theoretical background that the idea behind this thesis was born, from a proposal by Romain Brasselet, a mathematical neuroscientist working at Language, Learning and Reading Lab led by Davide Crepaldi: he suggested investigating whether and to what extent the formation of chunks from individual letters could be driven by an attempt to optimise a storage-computation trade-off, the tension between the tendency to store many different units, in order to minimise the number of their combinations needed to process a word, and the opposite inclination to minimise storage, paying the price of an increased computational effort. Is it realistic to think that the reading brain could form chunks from letters in an attempt to solve this trade-off? Could it be that it creates independent representations for particularly cohesive groups of letters in order to facilitate their recognition, and save on computational effort, while paying a cost in terms of storage? What would be the nature and size of the chunks in this case?

To find an answer to these questions, we tried to design the simplest possible formal model capable of embodying the storage-computation trade-off applied to the context of written word processing, by asking it to meet certain criteria that we considered important to include:

- the corpus on which the optimisation is based must be as realistic as possible: this allows us to have chunks that are meaningful from a linguistic point of view, and a more immediate comparison with experimental data; unlike other computational parsing models [e.g., 37], we choose not to use simplified artificial languages, and this will make it necessary to introduce computational tricks to deal with a super exponential number of possible combinations of optimal chunk sets;

- chunks of different lengths can compete with each other and co-exist in the set that resolves the tension; we do not place restrictions on the number of letters that can be chunked together (on the contrary, for example, in the MDLChunker model [42] the choice was to consider only binary chunks), but since we will consider each word as a separate input, the chunks will be at most as long as a whole word;

- unlike other models that aim to reproduce classical morphological decomposition as faithfully as possible - for example, the Naive Discriminative Reader proposed by Baayen et al. [33], a computational model in which morphological processing arises from form-to-meaning mapping-, we choose not to include any extra-orthographic information that could guide the formation of chunks, in order to understand whether morpho-orthographic segmentation can actually have an origin related to the statistical regularities of English language;
- finally, the fact that we choose a realistic corpus and thus obtain plausible chunks from a linguistic point of view allows us to propose a more reliable way of testing the computational model: the same morphological masked priming experiments used to investigate morpho-orthographic decomposition in human readers can in fact be used to compare the performance of the algorithm, without having to resort to simplified artificial languages, in the construction of which we could introduce arbitrary distributional patterns, and without having to ask participants to explicitly form bunch of symbols, a task that would lead them to reason strategically and not spontaneously.

Chapter 2

Minimal parsimonious chunking of written language: a first version of the algorithm

2.1 Objective function minimization

2.1.1 The problem

Investigating chunking as one of the mechanisms behind visual word recognition processes (see chapter 1), it is natural to ask which are the principles that lead to the segmentation of words, that is to the creation of chunks of letters. The project starts from the idea of trying to solve the tension between the storage of different chunks and the computational effort needed to process new words:

- storing a huge number of chunks would imply a great use of memory but a null computational effort;
- on the other hand, reducing the amounts of stored singletons (up to the only alphabetic letters) would mean lowering the memory load, requiring nonetheless to combine them more extensively (to the point that each word would have to be built from scratch).

2.1.2 The objective function

The challenge of optimizing the trade-off between storage and computation is formally translated into the problem of finding the set of chunks that minimizes a one-parameter

function L featuring two competing terms:

$$L = N + \alpha \cdot \bar{n}$$

where

- N is the number of stored chunks
- $\bar{n} = \sum_{i=0}^{N_{tot}} freq_i \cdot nchunk_i$ is the average number of chunks per word in a chosen corpus [see 2.1.3], with
  - $freq_i$ normalized frequency of the ith word in the corpus, i.e. the number of times a word appears in the corpus divided by the total number of words
  - $nchunk_i$ (minimum) number of chunks needed to process the ith word
  - N_{tot} total number of words in the corpus

In minimizing L, the first term N pushes for storing the least possible amount of chunks, while the second one, proportional to $\bar{n}$, is for having a vast set of chunks, in such a way that every word could be processed with the minimum number of steps (one chunk per word in the optimal situation). We call $morph$ the best set of chunks, the one that minimizes L.

The algorithm that tries to solve this optimization problem is implented in MATLAB version R2020b [43].

2.1.3 The choice of the corpus

In order to compute the average number of chunks per word $\bar{n}$, we used the SUBTLEX-UK corpus [44], a word frequency database for British English based on subtitles of British television programmes, which provides for each word appearing in the corpus the number of times it has been counted in all subtitles. Although it may seem a contradictory choice in a context of written word recognition, research in many different languages have actually shown that word frequencies based on television and film subtitles are better predictors of word processing times than the ones based on written sources [e.g., 45], and the same effect is observed for British English: the SUBTLEK-UK frequency database is more accurate in predicting lexical decision times than the British

National Corpus., a 100-million-word collection of samples of mostly written language [46].

Among the other choices, words forming an hyphenated expression are dehyphenated and considered independently, and contractions are **not** represented in the extended form, but if they are generated by more than one word, these are separated into different entries (e.g., *don't* $\rightarrow$ *do* + *n't*, *'ve* $\rightarrow$ *ve*, *'ll* $\rightarrow$ *ll*, *a'ight* $\rightarrow$ *a'ight*). The contracted expressions and many other word type whose spelling would not be accepted by a UK or a US word spell checker, including proper names, are classified as *X* in a special column of one of the file where the SUBTLEX-UK data are available (otherwise, the column presents the notation 'UK' or 'US'). This classification allows us to easily remove them from the corpus. Contractions in particular would not be easy to deal with, because the role the apostrophe plays here is controversial: on the one hand, it visually marks a separation in the word, on the other, it should a priori be treated like any other letter, because of their common nature as symbols. However, the choice made in SUBTLEX-UK to count the number of times an expression appears in the corpus by separating the different words involved in any contraction presupposes a kind of decomposition which is not necessarily the one done by the readers in the early reading stages (e.g., why exclude *don't* $\rightarrow$ *don*+*'*+*t*?), and it arbitrarily discards the role of the apostrophe.

2.1.4 How to include the positional-dependent constraints for chunks

Another issue we had to face is how to integrate in the algorithm the positional constraints that are demonstrated to characterise morphological decomposition: in the experiment by Crepaldi et al. [28], it emerges that the identification of morphemes is strictly related to the position they typically occupy in words. What is observed is that suffixes (by definition, morphemes that follow the stems) would not be identified as units if they appeared at the beginning of a word: for example, the presence of the suffix -*ful* at the beginning of a non-word *fulfgas* does **not** make it more difficult to reject with respect to the orthographic control *filgas*; that is, the brain is not keen on categorising *fulfgas* as a real word despite the presence of a familiar unit. The difficulty of rejection of a nonword, known as **morpheme interefence**, occurs instead

with morphologically complex nonwords like *gasful*, where the suffix follows the stem.

The solution we have adopted to keep into account the positional constraints consists in adding a symbol '_' at the beginning and at the end of each word of the corpus. Considering the previous example, the fact that *-ful* as a suffix appears mainly at the end of words will make it likely that a chunk *ful_* will be included in the best set $morph$: this will then be recognised in *_gasful_*, but not in *_fulfgas_*, correctly reproducing human behaviour. A similar argument can be made for the prefixes, that precede the stems.

However, such a choice shows its limits when for example a word has several suffixes one after the other: taking *_peacefulness_*, the suffix *-ful* here has the same role as in *_peaceful_*, but, since it does not end the word, the algorithm will not recognize the suffix *ful_*. Nevertheless, these words are rather rare in the corpus, which is a corpus based on spoken sources, especially if we set the maximum word length M to a certain value (we have seen that setting $M = 12$ already gives us significant results).

2.1.5 The α parameter

Of course, the best set of chunks highly depends on the value of the parameter α, as it can be clearly seen in the two extreme cases:

- when α is zero, $L = N$, and minimizing L is minimizing the storage, regardless of the huge amount of computation that this choice will require;
- in the opposite case, when α tends to infinity, the first term becomes negligible and L is reduced to the computational term, whose minimization forces the storage of all possible words.

The use of this parameter therefore allows us to adjust the relative weights of the two players in this competition, choosing which one to favour from time to time, and potentially mirrors psychologically meaningful phenomena: we can indeed imagine that an increasing value of α reflects a progressive improvement in reading proficiency. According to what has been theoretically suggested and experimentally proven in the field of reading acquisition [see 1.6], sensitivity to morphemes increases during reading development: we can expect that at the beginning each word would be processed letter

by letter, with a considerable computational cost that is reflected in the difficulty and slowness in written language processing; with the progressive mastering of literacy, the developing readers may start to form clusters of letters to be memorized, because they are recurrent and particularly cohesive, in order to recognize them as units and reduce the computational effort. Of course, we expect that beyond a certain threshold value, increasing α no longer makes sense: it is unrealistic for skilled readers to process every single word as an independent unit, otherwise no priming phenomenon would ever be observed.

2.2 How to do it in practice: the introduction of some computational tricks

2.2.1 The best set of chunks

The problem of solving the storage-computation trade-off is therefore turned into finding the set of chunks, over all the possible ones, that leads to the minimization of the objective function as we have defined it. But it's evident that the number of candidates is so huge that the minimization over this set can't be done in a reasonable time unless one introduces some tricks to fasten the algorithm. In fact, we will consider all chunks formed by a number of letters from 1 to M, and the candidates sets are all possible sets formed by these chunks.

We can compute exactly the number of candidates among which we have to search for the minimum: our alphabet Σ is formed by all the letters of the alphabet and the underscore '_', the only special character we keep in the corpus,

$$\Sigma = \{a, b, ..., z, \_\}$$

so it contains a number $S = |\Sigma| = 27$ of elements.

$$\Sigma^\star = \{(c_1, ..., c_s) : c_i \in \Sigma, s \in \{1, ..., S\}\}$$

is the set of all possible chunks in the alphabet Σ. The number of elements in $\Sigma^\star$ is given by

$$|\Sigma^\star| = S + S^2 + S^3 ... + S^M = \sum_{i=1}^{M} S^i = \frac{1 - S^{M+1}}{1 - S} \simeq 156 \cdot 10^{15}$$

choosing $M = 12$. And finally, every set of chunks can contain or not each element of $\Sigma^\star$ (2 possible choices for every element, to include it or not), and this corresponds to a super exponential number of possible candidates, equal to $2^{|\Sigma^\star|}$.

2.2.2 A trimmed reservoir

The first necessary cut is done on the set of all possible chunks $\Sigma^\star$ called $reservoir$, that in principle should contain all possible combinations of letters from 2 to M, where M is the number of letters of the longest words in the corpus. Some of these chunks are indeed so unusual that they cannot contribute in any way to lowering the objective function, and they can be safely excluded a priori. The condition that a chunk has to satisfy in order to be accepted in the best set is that

$$\Delta L < 0$$

where $\Delta L = L' - L$, and L is the value of the original objective function, L' the value of the modified function after the introduction of the new chunk; now, $L' = N' + \alpha \cdot \bar{n}'$, with $N' = N + 1$ the new number of stored chunks, so the condition for each chunk simply becomes:

$$\begin{aligned} L' - L &< 0 \\ N' + \alpha \cdot \bar{n}' - N - \alpha \cdot \bar{n} &< 0 \\ \alpha(\bar{n} - \bar{n}') &> 1 \end{aligned}$$

Expliciting the expression for the average number of chunks, we get:

$$\alpha \cdot \sum_{i=0}^{N_{tot}} freq_i \cdot (nchunk_i - nchunk_i') > 1$$

but the number of chunks needed to process the ith word $nchunk_i$ has changed only for the words that contain the chunk ($nchunk_i - nchunk_i' = 0$ otherwise), so we can reduce the summation to these words:

$$\alpha \cdot \sum_{j \in C} freq_j \cdot (nchunk_j - nchunk_j') > 1$$

where C is the set of words that contain the chunk.

Now, we would like to find an upper bound for the left side of this inequality, in order to get the necessary condition a chunk has to satisfy to become part of the best set of chunks $morph$; that is, we have to find the maximum value that the difference $\bar{n} - \bar{n}'$ can take on, so that, if the inequality is not even fulfilled in this case, it cannot be verified in any case, and the chunk has to be excluded from the possible candidates. This means supposing to be in the extreme condition where $morph$ initially contains only the letters, and where the new chunk is used in **all** the words in the corpus which contain it (a chunk can be present in a word and not be used in the decomposition, if it does not contribute to the shortest path, see 2.2.4), and in this case we can see that the condition simply becomes:

$$\alpha \cdot \sum_{j \in C} freq_j \cdot (k-1) > 1$$
$$\alpha \cdot (k-1) \cdot \sum_{j \in C} freq_j > 1$$
$$\alpha \cdot freq_{chunk} \cdot (k-1) > 1$$

where k is the length of the chunk, and we have used: $\sum_{j \in C} freq_j = freq_{chunk}$, defining $freq_{chunk}$ the sum of the frequency of all the words in which a chunk appears.

This becomes clearer if we consider an example: taking the word ***_meaning_***, the introduction of the chunk *ing_*, whose length is $k = 4$, in the set of $morph$ that contains only the alphabetic letters + the space, would change the number of chunks used to decompose it from $nchunk = 9$ to $nchunk' = 6$, and so $\Delta nchunk = 3 = k - 1$.

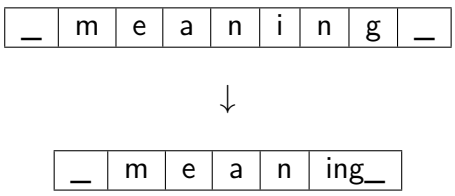

Figure 2.1: Example: how the number of chunks per word changes with the introduction of one chunk

We have found in this way a necessary (yet not sufficient), α-dependent condition,

that allows us to make a clean cut of the initial possible candidate chunks, fastening significantly the algorithm.

2.2.3 How to propose different sets of chunks

Once we have cleaned the $reservoir$, that now contains only the most reasonable chunks, always depending on the chosen value of α, we start from a $morph$ set made of the 26 letters of the alphabet + the space: in this case $N = 27$, and $\bar{n}$ simply coincides with the average length of the words. Then, we define some actions that allow us to propose different sets of chunks to choose the best one among: the $morph$ set is updated only when the new set lowers the current value of L. Combining the different actions and adding some random components, we hopefully reach a minimum in L and find the best set of chunks. The actions that the algorithm applies on morph are the following ones:

ADDITION The chunks contained in the $reservoir$ are randomly sorted, then the algorithm tries to add each of them to the best set $morph$: N becomes $N' = N+1$ and the average number of chunks per word is recomputed in $\bar{n}'$, but only for those words that include the proposed chunk (this is again a trick that fastens the algorithm, and it will be used also for every other action). If the value of the resulting objective function L' is less than L, then the new chunk is added to the best set $morph$, and N, $\bar{n}$ and L are updated to the new values of N', $\bar{n}'$ and L'.

DELETION This time the algorithm acts on the current best set $morph$, trying to remove from the set each of the chunks, once randomly sorted. Again, whenever a chunk is removed, L is recomputed and it will be updated only if $L' < L$.

EXTENSION Again starting from $morph$, one of the chunks is randomly selected and the algorithm tries to substitute it for an extended form, adding some letters (from 1 to 3) before or after it: in order to reduce the huge number of possible candidates, it is always checked if the extension belongs to the $trimmed\ reservoir$ and if it is not already member of $morph$, and the new L' is computed only in this case.

CUT Similarly to the previous action, this time it is proposed to replace a chunk in $morph$ with a cut version of it (without the initial or the final letter), always after

having checked the same conditions as in the extension case; then L is recomputed with the proposed $morph$ set and updated if $L' < L$.

SPLIT Finally, this action tries to split the selected chunk of $morph$ in two smaller chunks, and if both of them belong to the $reservoir$ and are not members of $morph$, and lastly if $L' < L$, they substitute the original one in $morph$.

Since the acceptance or not of a chunk in the best set $morph$ might depend on the order in which it is selected for evaluation, the order of the chunks in the $reservoir$ is randomly changed before every action, and the 5 actions are repeated a number p of times: some consistency checks [see 2.3.1] show that for $p = 5$, the algorithm is likely to obtain the set that correctly gives the global minimum of L, because, repeating many times the computation, the chunks in the best set have an overlap of at least 90%.

2.2.4 Decomposing each word using chunks: moving on to a graph

The next challenging point we had to face concerns the way in which, having a set of chunks, the algorithm could use them to decompose each word in the corpus, trying to minimize the number of chunks used to process (that is, to segment) each word. This procedure is then used after each of the actions described above to compute the new number of chunks per word $nchunk'$, for that words which contains the modified (added, removed, cut...) chunks of $morph$, and then the average value $\bar{n}'$ for all the words in the corpus.

The idea that allows us to fasten the process considerably is to build a graph for each word in the corpus, with a number of nodes equal to $l + 1$, where l is the length of the word. Considering as an example the word *_meaning_*, the associated graph will be the following 2.2:

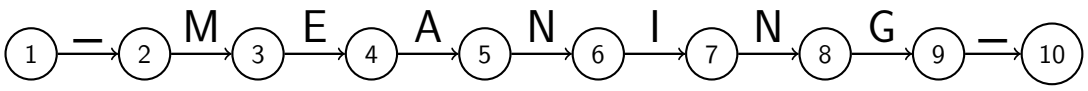

Figure 2.2: Graph representation of the decomposition process of the word *_meaning_* using the initial set $morph = \{a, b, ..., z, \_\}$

At the beginning, when $morph$= $\{a, b, ..., z, \_\}$ contains only the letters and the

space, the only edges present in the word graph are those connecting each letter of the word to the following one: the only path that the algorithm can follow to process the whole word is the one that goes through all the vertices one after the other, and this trivially coincides with the shortest possible path. The decomposition letter-by-letter is always a possibility. The adjacency matrix A associated with the graph in this case will have all elements equal to 0, except for the superdiagonal (the edge from node 1, before the first space, to node 2, after the space, corresponds to the element $A_{1,2} = 1$).

If through the actions described in the previous paragraph 2.2.3 some new useful chunks are introduced in $morph$, for example *ing_* and *ing*, with $morph = \{a, b, ..., z, \_, ing, ing\_\}$, their introduction is translated into the creation of the corresponding edges in the graph, which connect the node preceding the first letter of the chunk, to the node following the final one. In our specific example 2.3, adding the chunk *ing* corresponds to adding an edge connecting node 6 to node 9, while the chunk *ing_* introduces from the same node 6 to node 10. The corresponding elements in the adjacency matrix A are updated to 1: $A_{6,9} = 1, A_{6,10} = 1$.

Clearly, longer chunks reduce the number of steps required in the processing, and lower the value of the optimization function L. In this first version, all edges have the same weight, set equal to 1.

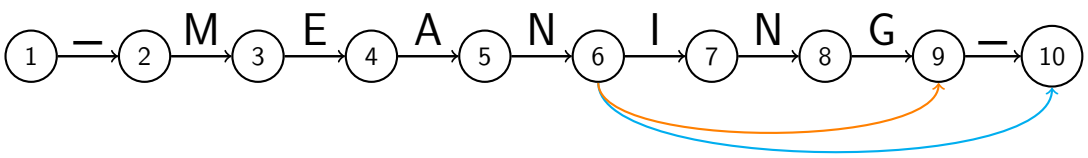

Figure 2.3: Graph representation of the decomposition process of the word *_meaning_* using the set $morph = \{a, b, ..., z, \_, ing, ing\_\}$

2.2.5 Finding the shortest path: Dijkstra algorithm

For the way in which we have formulated the problem, finding a way to decompose each word in the "energetically" cheapest manner correspond to finding the shortest path in the graph representing each word, from the initial node to the final one, using the edges that are created by the introduction of chunks in $morph$. In order to do

that, we used *Dijkastra algorithm*: this dynamic programming algorithm returns both the total weight of the shortest path – that in our case, since all the weights are set at 1, corresponds to the total number of steps needed in the decomposition, i.e. the number of chunks per word $nchunk_i$ – and the path it follows, from which we can easily compute the chunks it uses in the shortest path [47].

If implemented in a naïve way, the computational time required by *Dijkastra algorithm* is $\mathcal{O}(|V|^2)$, where V is the number of nodes in the graph, so in our case we would have a number of operation of the order $\mathcal{O}((l+1)^2)$ for each word of the corpus, with $\max l = M = 12$, and the number of different words with a number of letters less or equal than M in the corpus is 41062 (counting each word just one time, neglecting its frequency). The number of operation can be reduced using a *priority queue* structure implemented with an *heap*: in this case, the algorithm becomes almost linear in the number of edges $|E|$, with a computational time of the order $\mathcal{O}((|E|+|V|)log|V|)$. for our problem, using a *heap* is time saving since the adjacency matrices are generally sparse, that is the number of edges, corresponding to the useful stored chunks for decomposing each word, is much smaller than the number of all potential edges ($|V|^2$).

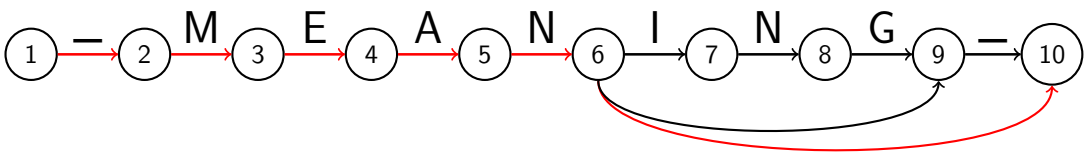

Figure 2.4: Graph representation of the shortest path decomposition of the word *_meaning_* using the set $morph = \{a, b, ..., z, \_, ing, ing\_\}$

2.3 Analysis of the results

2.3.1 Consistency checks

Before analysing the results, we verified that:

1. the algorithm actually finds a global minimum: this is not trivial, since there may be the possibility that the actions defined in [2.2.3] make the algorithm fall in a

local minimum, missing the global one. Computing many times the $morph$ set for the same value of α, and doing the same for different values of α, we obtain an overlap of at least 90% of the chunks;

2. the chunks that emerge for the first values of α tend to be part of the $morph$ set for higher values too: this trend is consistent with the psychological interpretation of the α parameter, that can be put in relation to the proficiency of the readers [2.1.5]. In the figure below [fig.2.5], we added a small vertical line for each chunk (x-axis) if it is present in the $morph$ set for that given value of α (y-axis), for values increasing from 10 to 40000. We can clearly see that chunks belonging to $morph$ for the smallest values of α - in the left part - remain part of $morph$ while α increases; in general, almost every new chunk, once added, becomes permanently part of the best set with higher values of α.

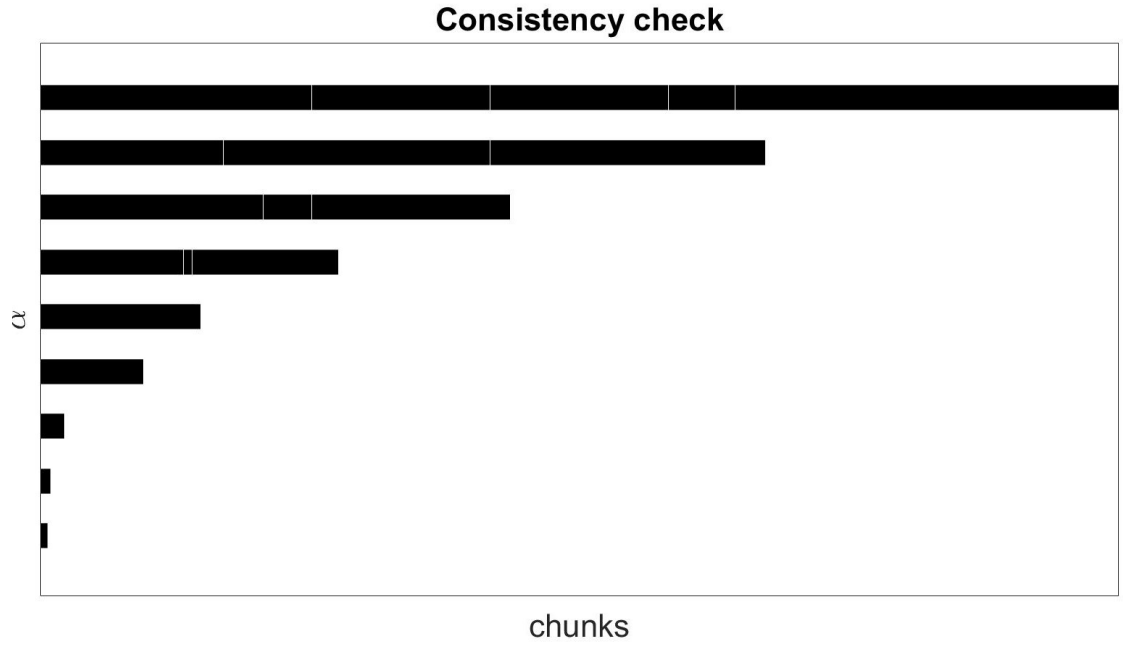

Figure 2.5: Consistency check: evolution of the chunks as α increases

2.3.2 The nature of the chunks in the best set

In order to analyze the chunks selected in the best set $morph$ for different values of $alpha$, it can be useful to divide them into three main categories:

1. **morphemes**: the meaning-bearing units considered in the classical morphological decomposition; they can still be subdivided into:
   - **roots**: morphemes that carry the main meaning of the words (e.g., *dark-* in *darkness*)
   - **affixes**: bound morphemes that can precede or follow a root, and are respectively defined
     - **prefixes**: e.g., *re-*, *un-*, *pre-* ...
     - **suffixes**: e.g., *-ly*, *-ness*, *-ful*, or the so called **inflectional** affixes, which have a grammatical function, such as *-s*, *-er*, *-ed*, *-ing*
2. **non-morphemes**: chunks of letters that are not classified in morphology, but that form a recurrent pattern easy to extract from the algorithm point of view.
3. **words**: when particularly frequent, the algorithm prefers to store entire words (including the initial and final spaces); we suppose that the storage cost is independent of the number of letters of the chunks added in $morph$, since we imagine that they become unitary representation, so it may be more convenient to have entire words as chunks in the best set.

In order to identify the trend in the chunk emergence, we should focus on smaller values of α than those we will use to compare the performance of the algorithm with the human one [see 3.10.3]; in this way, it is easier to understand the criteria and the order which the algorithm follows while integrating the chunks as α increases. In the following table 2.1, we report the most significant chunks that are **added** to $morph$ for increasing values of α, divided according to the categories explained above. We can notice that:

- those that the algorithm immediately identifies as chunks are the **inflectional suffixes**, which become part of $morph$ for very low α values: first, *s_*, used both

in the formation of noun plurals and of the verb present tense, 3^{rd} person singular (e.g., *_cat-s_*, *_read-s_*); then, *ing_* (e.g., *_read-ing_*), *er_* (e.g., *_read-er_*, but also *_great-er_*), *ed_* (e.g., *_work-ed_*), *en_* (e.g., *_soft-en_*);

- **prefixes** appear later than inflectional suffixes, at the same time as the **derivational suffixes**: *_in* (e.g., *_in-come_*) and *ly_* (e.g., *_slow-ly_*), *_pre* (e.g., *_pre-school_*) and *ful_* (e.g., *_care-ful_*), *_dis* (e.g., *_dis-like_*) and *ment_* (e.g., *_govern-ment_*) emerge simultaneously;

- a range of **non-morphological** chunks, which do not perform a precise grammatical function and carry no meaning, appear very soon in $morph$; among the first: *_th*, *ou*, *ow_*, *_wh*, *ight_*, *ould_*;

- the algorithm finds it hard to detect the **roots**, which only appear from relatively high values of α: for $\alpha = 1000$, *_govern*, *_happen*, *_look*, and so on begin to be part of $morph$;

- together with the integration of the roots, some **non-morphemes** that do not have a clear place in the psychological landscape are also added: again for values of α greater than 1000, we find chunks such as *king_*, *ding_*, *ted_*, *der_*, which join the already integrated morphemes *ing_*,*er_*, *ed_*;

- from very small values of α, the algorithm tends to store as units whole **words** and in particular conjunctions (e.g., *_and_*, *_but_*), prepositions (e.g., *_for_*, *_of_*, *_in_*), pronouns (e.g., *_you_*, *_I_*); as α increases, also some particularly frequent nouns and adjectives become part of $morph$; it is interesting to see that words like *_government_* and *_public_* are among the first to appear, and this is a clear reflection of the kind of sources that form the corpus.

2.3.3 First take home messages

From the analysis of the chunks that become part of the best set $morph$, we obtain interesting results on four main lines:

1. Even without any semantic knowledge or grammatical hint, the algorithm proves to be able to identify the main English affixes, with a propensity to identify the

inflectional suffixes. It is therefore reasonable to think that the perception of morphemes as units in the decomposition may have a different origin from that which stands out in classical linguistics, according to which them arise as a product of the connection with semantics, or, even before, in a mapping with phonology. Or rather, precisely because morphemes carry a specific meaning or perform a certain grammatical function, they are used in language with such regularity that the algorithm tends to integrate them almost immediately into the minimal maximally efficient set of chunks.

2. At the same time, semantics could actually play a fundamental role in the **crystallisation** of morphemes: from the table 2.1, it is clear that for example the morpheme -*ing* is among the first to be included in $morph$, but as α increases, thus increasing the storage availability of the 'reader', other chunks like -*king*, -ting, -ding are added to it, which are useful in the decomposition of many terms of the corpus, on the same level as the prefix *dis-* or the suffix -*en*. If from the point of view of the **statistical chunking** these units can be useful or significant, the **semantic level**, which is involved in reading at later stages, finds it more convenient to isolate roots and affixes (e.g. *reading=read+ing* instead of *rea+ding*). A decomposition aimed at capturing the meaning could influence also early visual processing for skilled readers, since a chunking of the *rea+ding* type would lead to a slowdown due to the need for a second morphology-based decomposition. That is, it may be that at a certain stage in learning, corresponding to a certain value of our α parameter, it becomes necessary to take into account the linguistic nature of the symbols of the code we are studying, that will subsequently be elaborated by other levels of processing: for a reader who reads language, the economy in the choice of the chunks to store will also be aimed at speeding up the grasping of meaning, a principle that is not included in an algorithm which just 'sees' symbols. In this respect, we notice that the algorithm struggles to identify **roots**, because they are many and much less productive than affixes. In the experiment by Hasenäcker et al. [41] mentioned in 1.6, it seems instead that the morphological decomposition in children starts first of all by the recognition of compounds, so the opposite asymmetry is observed. Indeed, if it is true that in the

affixes we find a hint about the function of each word (*-ing* expresses the action of the verb or its result, *-er* indicates a person or thing that performs an action), the roots are the most informative parts of words, carrying the main lexical content.

3. Another question that emerges from the analysis of the stored chunks, and that would be interesting to investigate, is the role of the **non-morphemes** in word processing: starting from very small values of α, some cohesive and recurrent chunks emerge (e.g. *wh-*, *-ould*, *-ake*, *-ough*) that do not have any grammatical function or role in the form-to-meaning mapping. Yet, as elements of regularity, they could actually be perceived as units by readers and be significant in the orthographic decomposition. Let us take, for example, the experiment cited in [9], in which six baboons were able to distinguish four-letter real English words from artificially generated non-words characterised by low bigram frequency. We might wonder whether non-words formed by these non-morphological units (e.g., *whot*) are equally difficult to classify as not belonging to English language as other non-words formed by real morphemes (e.g., *teing*), and thus whether this algorithm trying to identify the minimal set of chunks somehow manages to reproduce a mechanism followed in the processing of orthographic information in the absence of pre-existing linguistic representations, in a better way than considering only the bigram frequency as the experiment does. The presence of this kind of chunks is in line with the idea of a morpho-orthographic segmentation independent from the meaning: it is true that they do not carry any semantic information, but they can help to identify certain recurrent words (e.g., *what*, *who*, *would*, *could*) that are fundamental in the construction of sentences.

4. Finally, it may be interesting to see how realistic it is to suppose that the most frequent words are not decomposed into chunks: is it possible that the human brain finds it more convenient to create independent representations for them in order to save on processing? This possibility, which naturally emerges from our algorithm, is not contemplated by other models explicitly designed to model learning of morphological segmentation [e.g., 48], which always force a decomposition in stem+affix. Actually, if in the case where the prime is consciously visible to the participants it has been shown that the priming effect is greater for low-frequency

primes than for high-frequency ones, it is not clear if the prime frequency plays any role in the earlier stages of the word identification processing [see 49]. Further experimental evidence could help to understand whether morpho-orthographic decomposition remains a necessary step in word processing even for skilled readers, with a broad vocabulary knowledge. Moreover, in our algorithm we see that the number of words that become part of $morph$ increases with α. A very interesting experiment from this point of view is the one carried out by Häikiö et al. [50], who showed that Finnish children were in general slowed down when reading compound words in which the morphological components were explicitly separated by a hyphen, while for slow readers the fixation duration (indicative of the processing time) was shorter for these words: this suggests that slower readers process compounds through constituent morphemes, and so they are helped when the morphemes are well separated, but as proficiency increases skilled readers might acquire whole-word representation, as in the case of our algorithm.

α		10	20	30	50	100	200	1000	2000
morph.	prefixes						_dis	_pre	_dis
	suffixes	s_	ing_, y_		er_	al_, ed_, ion_, ly_	en_, et_	ful_, ent_, ment_	ence_
	roots					_go	day_	_every, _govern, _happen, _look, where_	_any, _enjoy, _friend, _some, _wonder, body_, side_, thing_
non-morphemes		_th	ou	_thi, ll_, ow_	_wh, ight_, ould_	_qu, _sh, _st, ake_	ble_, der_, king_, ding_, ted_	ally_, ble_, der_, ough_, own_	ded_, ings_, ortant_, _brea
words			_of_, _to_, _you_	_and_	_a_, _for_, _have_, _i_, _it_, _that_	_about_, _be-cause_, _but_, _from_, _if_, _is_, _like_	_can_, _do_, _know_, _think_, _very_, _well_	_against_, _always_, _nice_, _public_, _would_	_best_, _football_, _govern-ment_, _tomor-row_
number of chunks in *morph*		31	44	52	71	114	169	505	768

Table 2.1: Chunk emergence as α increases

Chapter 3

Not all chunks are equal: an implemented version of the algorithm

3.1 Algorithm limitations

From the analysis of the chunks selected in the best set $morph$ by the algorithm [see 2.3.2], we notice that the main discrepancy with respect to what has been experimentally observed consists not so much in the formation of 'harmless' non-morphemes such as *_th, _wh, ight_*, which appear for very small values of α and which might actually emerge as cohesive units for the reader, but in the integration of non-morphological chunks that contain other morphemes, in particular inflectional suffixes, and that substitute the morphemes in the shortest path decomposition of words when they occur: *king_*, *ding_* join *ing_*, *ted_* and *ded_* are added to *ed_*, and *ter_* to *er_*. The emergence of this kind of chunks is not matched by experimental data, where the opposite trend is observed, that is the tendency to recognise these morphemes even when they do not play a real morphological role (e.g., corner-CORN) [see 1.4]. What the algorithm tends to do instead is a chunking of the kind *_departed_* $\rightarrow$ *_de-p-ar-ted_*, *_booking_* $\rightarrow$ *_boo-king_* ($\alpha = 5000$), which prevents it from correctly identifying roots. As the algorithm includes longer chunks in $morph$, and begins to identify the first meaningful roots, these non-morphological chunks also appear symmetrically, as if they were 'roots' at the end of words (*ortant_* appears with *_enjoy*, *rrow_* with *_ignore*).

In essence, this naïve reader, whose only purpose is to find the best compromise

between storage and computation without any semantic or morphological information, proves to be able to select some interesting chunks, but some choices are not the expected ones compared to human performance. We wonder whether it is possible to obtain something more consistent with what has been experimentally observed without adding assumptions that go beyond the data available from the corpus, i.e. the list of words with the associated frequency.

3.2 One possible improvement: redefining our aim

A natural evolution of the algorithm can be obtained with the introduction of some shades when establishing the weights of the chunks, that indeed are not all equivalent: ideally, we would like to assign a weight that is smaller for the chunks that are used more frequently in the shortest path decomposition of the different words in the corpus. Referring to the previous example, the chunk *ing_*, used to create the present participle and the gerund in English, is so frequent that should be in any case preferred in the decomposition to the less 'productive' chunk *king_*, that is much more uncommon since it is not a morpheme. Our purpose is now redefined as establishing different weights according to the chunks **productivity**, where the term (actually already present in Linguistics with a wide range of nuances in its use) is to be understood as indicative of the number of times a chunk is used in word processing. We now want to see if the introduction of these weights can lead to solving the problem of non-morphemes competing with morphemes.

3.3 How to choose the weights: a vicious circle

The problem in establishing the weights raises in the definition of productivity itself: if we want to quantify the weight of a chunk according to its use in the shortest path decomposition of the words, we end up in a loop: this is of course something we can't say a priori, before having run the algorithm, and, at the same time, if we run the algorithm with all the weights equal to 1, it would use the chunks as if they were all equivalent, so the results for the number of times a chunk is chosen in the process would not be the desired one: in a sense, in order to run the algorithm, at least the

first time, we are forced to make an assumption about their weights that cannot be based on their productivity.

3.4 A practical choice: the chunk occurrence as a weight

One possible way to overcome the problem, at least for a first step towards the ideal solution, is to define a chunk's weight according to its overall (token) frequency in the corpus: the more frequently a chunk appears in the corpus, the more we want it to be favourite in the word decomposition, that is, the less it should be weighted. The occurrence represents a sort of **upper bound** for the productivity of a chunk, since it will not of course be used in the shortest path for all the words in which it is present. We are doing an optimistic approximation, which we have to take into account in the analysis of the results; however, the strong point of this practical definition lies in the fact that it enhances the **cohesion** of the chunks, that is a fundamental property in our perspective: the probability for a bunch of letters to become part of our best set $morph$ is higher when they happen to be found together many times. Moreover, this allows creating a sort of connection between the words in the corpus, that are no longer analysed independently, but seen as a whole: the way in which a word is decomposed depends indeed on the statistical properties of all the corpus, and if a chunk appears in many different words, or in a few ones that are particularly recurring, it will be more easily stored and identified as a basic unit for the decomposition. This idea is in line with the principles used in the elaboration of computational models of speech segmentation and the results observed using artificial languages built with some frequent chunks in a fixed position [see 1.5]. However, the experiment recently led by De Rosa et al. [51] seems to exclude letter chunk frequency as a main player in visual word identification. It may be interesting for our purpose to understand it in more detail. In that case, indeed, no differences are observed in the priming effect induced by non-word primes formed by an existing stem (e.g., *bulb*) with

- genuine suffixes (e.g.,-*ment*)
- non-morphological but frequent word endings (e.g.,-*idge*)
- non morphological and non frequent word endings (e.g.,-*kle*).

A strong priming effect is observed in all conditions, independently of the morphological nature of the word ends (and this is in agreement with our model, which constructs non-morphological chunks) nor on the their frequency. But in contrast with the study of artificial languages, in this experiment they investigate material that is already known to the readers: the non-words are still created by the union of existing stems and word-endings, so it could be that the frequency of chunks stops being relevant when knowledge is already well-established, and it could instead play a fundamental role in the formation of chunks during reading acquisition, which is the moment we try to investigate with our algorithm.

Finally, since the experimental data do not give any indication in this regard, we decide to consider the **token frequency** of the chunks, that take into account the frequency of words containing a chunk, and not only the number of different words where we find it (type frequency). We imagine in fact that the corpus is a kind of single continuous text to which the algorithm is exposed: words will appear in this text a number of times equal to their frequency, and will not be grouped if they are the same as presented in the dataset. We will see later that by switching to surprisal, i.e. considering the logarithm of the frequency, the effects of considering the token frequency instead of the type frequency are considerably reduced.

3.5 An operative definition for the occurrence

If we define the occurrence o_i of the i^{th} chunk simply as the number of times it appears in the corpus, that is, the sum of the non-normalized frequencies of the words that contains it:

$$o_i = \sum_{j \in C} freq_j \cdot n_{ij}$$

where C is the set of words which contains the chunk i, n_{ij} is the number of times a chunk i appears in the j^{th} word, then the normalization results to be non-trivial: this is due to the fact that the chunks have a variable length – from 1, considering also the letters, to M –, so in principle each set of chunks with the same number of letters would have its own sample space, given by all the possible combinations of letters of that length. In this way, it would not be possible to compare the occurrence of chunks

of different length, that is actually what we need to do. In order to overcome the issue and have a uniform and common sample space, we normalize the occurrence over the total $O = \sum_i o_i$, where the summation goes over the occurrences of all possible chunks. In this way the resulting normalized occurrences

$$o'_i = \frac{\sum_{j \in C} freq_j \cdot n_{ij}}{O}$$

are quantities between 0 and 1 that can be interpreted as probabilities.

3.6 From the occurrence to the weight: the surprisal

Having defined the occurrence, we now have to pass to the weights, that intuitively should be smaller the higher is the chunk occurrence. But we can make some more precise requirements the weights have to satisfy:

- The less probable a chunk is, the higher its weight is: we want it to be disadvantaged in the shortest path decomposition, in favour of the more frequent ones.
- If a chunk has probability 1, its weight should be null; if a chunk has a probability 0, that is it is never used in the corpus, its weight should be infinite. More generally, the weights should be a non-negative, continuous and decreasing function of the chunk probability.
- The total weight of two independent chunks is the sum of the weights of the two chunks; if the chunks are not independent, then their total weight should be minor than the sum of their weights.

These requests led us to take as a weight Shannon *self-information* or *surprisal* [52]. In order to make them satisfied, this is indeed the unique function we can use, up to a multiplicative scaling factor:

$$w(p) = -log_{10} p$$

where we could have chosen whatever base for the logarithm, that corresponds to the arbitrary scaling factor.

There is indeed an idea of surprise in our definition of the weight: when decomposing a word, the more common the sequence of letters is, the lower their weight, while an

unusual, more "surprising" combination of letters costs more computational effort, and it is associated with a higher weight.

Let's analyse in particular the third requirement, in order to check that the logarithmic function satisfies it; as an example, we consider the chunk *ed_*, the English suffix used to form the past participle of regular verbs and of participial adjectives which express a condition or quality resulting from the action of the verb. Since the chunk is very common, the events 'use of the letter *e*', 'use of letter *d*', 'use of the space _' are non-independent, and so:

$$P(e \cap d \cap \_) = P(e)P(d|e)P(\_|e \cap d)$$

and in particular, we can suppose that they are positively correlated, that is:

$$P(d|e) > P(e)P(d)$$

and

$$P(\_|e \cap d) > P(\_)P(e \cap d) > P(\_)P(e)P(d)$$

and finally:

$$P(e \cap d \cap \_) > P(e)P(d)P(\_).$$

Let's see how this property is translated in terms of the weights: since the logarithm is a monotonic function:

$$logP(e \cap d \cap \_) > log[P(e)P(d)P(\_)]$$

and then

$$-logP(e \cap d \cap \_) < -log[P(e)P(d)P(\_)]$$

that becomes, using the property of logarithm:

$$-logP(e \cap d \cap \_) < -logP(e) - logP(d) - logP(\_)$$

and in terms of weights:

$$w(ed\_) < w(e) + w(d) + w(\_)$$

that is exactly the result that we were hoping to obtain: when a chunk is frequent, it is convenient to see it as a unit instead of process the sequence letter by letter.

3.7 Finding the shortest weighted path

With the introduction of the weights, it is clear that finding the shortest path takes on a different meaning: the path the algorithm has to choose in the decomposition is not anymore the one which involves the smallest number of steps – corresponding to the decomposition of the word that requires the lower number of chunks - but the one which ensures that the sum of the selected chunks' weights is the lowest possible one. This is done automatically by Dijkstra algorithm – that can compute the minimum path weight between two connected nodes in a weighted graph, for weights that are real numbers – once having updated the adjacency matrix, whose elements are not anymore just 0 or 1, but real values corresponding to the weights of the chunks. The introduction of the weights results in this sense a natural extension of the original algorithm: to restore it, it is sufficient to set all the weights equal to 1, and the minimum weighted path will correspond again to the number of hops.

3.8 Modifying the objective function

Equally naturally, it is possible to give an extended definition of the objective function, simply replacing in the second term of L the average number of chunks per word $\bar{n}$ with the average weight of the words in the corpus $\bar{w}$: indeed in this case what we want to minimize is not the number of chunks used in the decomposition of the words, but the total weight of the chosen chunks. Therefore, since $\bar{n} = \sum_{i=0}^{N_{tot}} freq_i \cdot nchunk_i$, it is sufficient to replace $nchunk_i$, the minimum number of chunks needed to process the ith word, with $wchunks_i$, the sum of the weights of the lightest chunks used to decompose the i$^t h$ word, while N_{tot}, the total number of words in the corpus, and $freq_i$, the normalized frequency of the i$^t h$ word in the corpus, remain the same, that is:

$$L = N + \alpha \cdot \bar{w}$$

with

$$\bar{w} = \sum_{i=0}^{N_{tot}} freq_i \cdot wchunks_i.$$

Once again, if all the weights are equal to 1, then $wchunks_i = nchunk_i$, and the original algorithm is recovered.

3.9 A new cutting for the reservoir

The last necessary adjustment that has to be made to the implemented code concerns the sufficient condition that a chunk has to satisfy to become a possible candidate for morph. From the previous version [see 2.2.2], the necessary condition to be fulfilled by a chunk is immediately translated into

$$\alpha(\bar{w} - \bar{w}') > 1$$

that is

$$\alpha \cdot \sum_{i=0}^{N_{tot}} freq_i \cdot (wchunks_i - wchunks_i') > 1$$

and in order to find which is the maximum value that the difference in the average word weight can take on, before and after the introduction of a chunk, we have to consider again the extreme condition where $morph$ is made up of only the letters and the space: in this case, the weight of each considered word will be simply the sum of the weights of the letters; then, supposing to add just one chunk to $morph$ and to use the new chunk in all the words in which it appears, the change in weight for that chunk will be given by the difference between the sum of the weights of the single letters that compose the chunk, and the weight of the chunk:

$$(wchunks_i - wchunks_i')_{max} = \sum_{l=0}^{k} weight(letter_l) - weight(chunk)$$

with k is the length of the chunk, and the result is independent of the considered word. The condition a chunk has to satisfy to become part of the new *trimmed* reservoir finally becomes:

$$\alpha \cdot freq_{chunk} \cdot \left[\sum_{l=0}^{k} weight(letter_l) - weight(chunk) \right] > 1.$$

3.10 Analysis of the results

3.10.1 Comparing the two versions of the algorithm

The introduction of the weights in the algorithm shows 3 main features:

1. the logic behind the algorithm remains unchanged: we have not introduced any kind of assumption beyond the knowledge of the corpus and the word frequency. The token frequency of the chunk was in fact already a discriminating criterion in the admission of a chunk into the best set, because it appears in L also in the original algorithm. It is therefore natural to expect that the analysis of the chunks belonging to morphs as α increases does not present substantial changes with respect to the previous case, in which the weights were all equal to 1 [see 3.1]. The values of α at which chunks of different nature appear are different, because in this case the trim to the reservoir, based on the frequency of the chunk, is sharper, but the trend according to which the chunks are integrated in the best set remains the same.

2. in the original algorithm, once the chunks became part of the best set $morph$ and stored, they are all considered equivalent, so the frequency comes into play **only** in the selection of the chunks, but not in the shortest path decomposition: in this second version instead, the algorithm is able to store the weights associated with the chunks, and will take them into account in the decomposition of the words. This greatly reduces the otherwise common possibility of having equivalent decomposition paths, from which the algorithm would have to choose arbitrarily, and would always choose the same one, possibly introducing an error in assessing the productivity of the chunks.

3. the introduction of weights makes it possible to insert **extra-orthographic** information for further tests, for example to associate a lower weight to chunks bearing a meaning, or to those associated with phonemes, which, appearing in more than one level of processing, could be more easily memorised and crystallised. A modification of this kind is therefore necessary in order to integrate new principles into the algorithm to assess their relevance in human performance.

α		100	200	500	700	1000	1500	2000	3000
morph.	prefixes				_in		_re	_pro	_ab
	suffixes		ing_, s_		er_, ed_	en_, ion_, ly_		ent_	al_
	roots					thing_, _go	_know, _look, _some	_every, _want, where_	_make, _see
non-morphemes		_th		at_, here_, ou	_wh, ve_, ould_, ll_	ally_, king_, ight_	ow_, ting_, tion_, ther_	ound_, ink_, ell_	ook_, ted_, ything_
words		_the_	_you_, _and_	_for_, _have_, _i_, _it_, _of_, _that_, _to_	_there_, _what_, _with_	_about_, _like_, _peo-ple_, _think_	_could_, _right_, _very_	_time_, _them_, _some_	_today_, _want_, _govern-ment_
number of chunks in *morph*		30	40	78	107	163	250	332	486

Table 3.1: Chunk emergence with weights as α increases

3.10.2 The non-morphemes including morphemes are still there

Although the introduction of weights proportional to the frequency of the chunks in the corpus should discourage the introduction in $morph$ of chunks of the type *king_*, *ter_*, *ter_*, because the morphemes *ing_*, *er_*, *ed_* they contain are obviously much more frequent (*ing_* is present every time we see *king_*), we see that these chunks appear anyway as α increases, and they are in any case preferred in the shortest path decomposition of words. Similarly, the algorithm chooses to consider *_years_* as a single chunk, instead of decomposing it into *_year+s_*.

However, we can explain the reason behind this phenomenon; consider as an example the empirical probability (i.e., the normalised frequency) of a given word:

$$P(\_works\_) = P(\_work)P(s\_|\_work)$$

but:

$$P(s\_|\_work) \gg P(s\_)$$

since, even if $-s\_$ is an inflectional suffix, therefore frequent, after the root $\_work-$ there can only be a few suffixes ($ing\_, er\_, ...$). Taking the logarithm and changing sign, we move on to the weights:

$$-\log_{10} P(\_works\_) \ll -\log_{10} P(\_work) - \log_{10} P(s\_)$$

and finally we get:

$$w(\_works\_) \ll w(\_work) + w(s\_)$$

that explains why the algorithm prefers to keep the word as a whole instead of decomposing it. We deduce that this is what happens with the chunk *king_*, where *k* is probably among the letters that most frequently precede *ing_*, or maybe there are few words ending in *king_* but very frequent in the corpus, since we are considering the token frequency. The same principle makes the algorithm happy to accept *ing_* as a chunk instead of the single letters, but here we see the downside of this way of defining the weights, that still does not solve the problem.

3.10.3 Predicting human performance

Masked priming dataset

To better understand which are the main discrepancies between the algorithm and the human performance, we used data from four morphological masked priming studies ([22], [24], [53], [54]), collected by Amenta et al. [55] (we decided to exclude the values from [56], used in [55], since in that case the participants are non-native English speakers).

The morphological masked priming experiment [see 1.4] is a written word priming experiment that tries to investigate the role of morphology in written word recognition processing. The participants are asked to determine as quickly and accurately as possible if a letter string, the target, is an existing word or a random string (lexical decision task): if they recognize an English word, they have to press a YES button, controlled by the dominant hand; if instead they are presented with a nonword, in general chosen to be pronounceable (e.g., *slint* or *bant*), they have to press another button. The presentation of the target is preceded by a very brief presentation of a related prime (*dealer-DEAL*) or a control prime, which is a word orthographically, morphologically, and semantically unrelated to the target (*cutter-DEAL*). The same is true for the nonword targets, that can be preceded by words containing the target (e.g. *banter-bant*) or by an orthographically unrelated word (e.g. *slinter-bant*). Being in a masked priming condition, the stimulus-onset asynchrony (SOA), that is, the time interval between the prime and the target presentation, is short (indicatively, less than 50 ms), and the participants are not told of the existence of the primes. Each prime is also preceded by a 500-ms forward mask (#####). The participants generally do not report having seen the prime. For each prime-target pair, the reaction time (RT, in milliseconds), that is, the time interval for the response of the participants to the task, is collected. The priming effect magnitude (PEM), computed as the difference between the RT in the related condition and the RT in the unrelated condition, is then the variable we are interested in. The PEM represents indeed an estimate of the relationship between the prime and the target: the more the presentation of a related prime facilitates the recognition of the target, compared to the presentation of an unrelated prime, the larger the priming effect magnitude will be.

Each related prime-target pair belongs to one of the following conditions, assigned in the original studies from which the data were collected:

- in the **transparent** condition, prime and target are morphologically and semantically related: in this case there is a true morphological relation, which follows the classical definition in linguistics (e.g., *dealer-DEAL*);
- in the **semi-transparent** and **opaque** condition, which we will group into a single category, the morphological relationship is only apparent, since there is no semantic relationship between primes and targets, that is, the meaning of the complex words cannot be derived from the meaning of their constituents; the distinction between semi-transparent and opaque conditions is not always clear: in the first case, there is still some sort of semantic relation, even if not direct (e.g., *archer-ARCH*: an archer is someone who does something that is related to the meaning of *arch*, and there is an etymological relation: Latin *arcus*), while in the second case this relation is completely absent (e.g., *corner-CORN*, where *er_* is clearly not a suffix and *corn* is not a root);
- finally, in the **form** condition, prime and target only shares an orthographic overlap: the prime is morphologically simple, and it happens to contain the target as an orthographic substring (e.g., *scandal-SCAN*).

Related primes and targets are typically matched across conditions as closely as possible on target frequency, prime frequency, target neighborhood size, target length, target family size (the number of its derivations), and form overlap (number of prime letters shared with the target divided by number of target letters); unrelated primes are matched as closely as possible on frequency and length to each corresponding related prime. In this way, an attempt is made to isolate the morphological relation as possibly being responsible for facilitating lexical decision.

Following the classification into these three conditions may be useful from our point of view because it has proved to be effective in predicting the priming effects: results from many different studies showed that priming effects are significantly greater for pairs belonging to the transparent and opaque/semi-transparent conditions than for pairs in the form condition. It is not clear instead whether there is a difference in

priming for transparent and opaque words: in the experiment by Rastle et al. [24], the two conditions seemed to produce equivalent priming effects, while Andrews et al. [54] found out that the phenomenon actually could depend on individual differences in spelling and vocabulary.

The overall dataset includes 296 different targets, which appear in one or more of the mentioned experiments. For each related prime-target pair, the PEM is computed and expressed in ms. The error on the PEM for each related prime-target pair was not provided in the original studies, so the analysis must be understood as qualitative, and as a possible forerunner for structuring priming experiments specifically designed to investigate the storage-computation trade-off in processing.

The algorithm performance on the dataset

We used the updated version of our algorithm to analyze the dataset: for different values of α, we computed the best set $morph$, which contains the optimal chunks and their weights. Having $morph$, we were able to obtain the weighted shortest path decomposition for each prime of the dataset (adding a space before and after it), using the graph representation of each word and Dijkstra's algorithm (as we did to obtain $morph$ 3.7): this allowed us to know which chunks belonging to $morph$ guarantee the most efficient processing of the word from the point of view of our algorithm. We chose not to exclude from the dataset the words of length greater than M=12, because it was interesting to understand how the algorithm processes words that do not belong to the corpus on which the optimisation was performed.

In the transparent and opaque/semi-transparent condition, the prime is a morphologically complex word (in the dataset, only the root+suffix combination is considered), the target coincides with the root (e.g., for the transparent condition, prime: *_cloudless_*, target: *_cloud_*; for the opaque/semi-transparent condition, prime: *_department_*, target: *_depart_*), and the difference between prime and target is a suffix (*less_*, *ment_*); in the form condition, the prime is morphologically simple, but it contains the target (e.g. prime: *_basilica_*, target: *_basil_*), and the remaining letters do not have a precise role or meaning (*ica_*).

In analysing the results, we separate 3 possible situations that can occur:

1. the algorithm can identify the **suffix** embedded in the prime, or the ending group of letters in the case of the form condition, but not the root/target (e.g. prime: *_drainage_*, target: *_drain_*, prime decomposition: *_dra-in-age_*; prime: *_dialect_*, target: *_dial_*, prime decomposition: *_di-al-ect_*)

2. it can correctly chunk the **root/target**, but not the suffix/ending letters (e.g. prime: *_parenthesis_*, target: *_parent_*, prime decomposition: *_parent-he-s-is_*)

3. it can identify **both** the root/target and the suffix (e.g. prime: *_soften_*, target: *_soft_*, prime decomposition: *_soft-en_*)

Our decision to consider separately the evolution of these three situations is in line with the literature on morpho-orthographic decomposition, where two main mechanisms have been proposed as responsible for the phenomenon: according to the **affix-stripping** approach, if a string contains an affix, this is automatically chunked and removed, so as to facilitate the processing of the remaining part of the word, that is usually a stem; according to an alternative proposal, instead, morpho-orthographic decomposition could start with the extraction of the embedded root (**embedded stem activation**), because this would contain more information than the affixes [57]. The first approach, proposed among the first by Davis [58], is supported for example by a letter search experiment where participants had to establish as quickly as possible whether a target letter was present or not in a pseudoword: the results showed that the target letter was more difficult to identify if it was contained in a prefix or suffix embedded in the pseudoword (e.g., *R* in *propoint* or in *filmure*, where *pro-* is a prefix and *-ure* is a suffix) than if it was included in a non-prefix beginning or non-suffix ending of the pseudoword (e.g., *R* in *cropoint* or in *filmire*). This effect would be explained by the automatic chunking of prefixes and suffixes, which would be perceived as units and therefore not analysed letter-by-letter. The idea of an automatic embedded stem activation, instead, arose in an attempt to explain another experimental evidence: Morris et al. [59] showed that in priming experiments where the primes are pseudo-words made of an existing root compounded with a suffix (e.g. *farmity-farm*) or with a non-suffix (e.g. *farmekt-farm*), the priming effects are equivalent in the two cases and do not depend on the morphological nature of the ending bunch of letters: this would suggest that

it is the presence of the root that determines the facilitation in the processing of the target when the prime is presented, regardless of the ending chunk. The absence of a priming effect in the form condition (no facilitation for pairs like *scandal-scan* despite the fact that the target is actually contained in the prime) would be explained by a lateral inhibition effect between co-activated word representations, that arises in the case where the embedded stem is itself a word (*scandal* inhibits *scan*).

Analyzing the behavior of the algorithm 3.1, we easily see that the suffix identification and the root identification cases follow a different development as α increases: as we have already noticed from the analysis of the chunks, suffixes begin to be detected starting from very low α values, to the point that the number of suffixes/ending chunks correctly processed by the algorithm is maximum (144, out of 296 pairs) for $\alpha = 10000$, when the correctly detected roots/targets are only 7; in fact, well chunked roots are null up to an α value of about 5000, and then, although they grow with α, they are always less numerous than the detected suffixes.

Between the two approaches aimed at explaining the mechanisms underlying morpho-orthographic segmentation, our algorithm, which is based on a principle of storage-computation trade-off optimisation, seems to proceed consistently with the affix-stripping idea; we can therefore hypothesize that if this second explanation turns out to be the correct one, then the origin of morpho-orthographic segmentation characterising the early stages of word processing could be attributed to a language-independent statistical learning mechanism, rather than to a form-to-meaning mapping that cannot disregard semantics. However, we will see below that semantics cannot be completely excluded from this first level of processing [3].

In the evolution of the decomposition of primes in the **transparent** or **opaque/semi-transparent** condition as α increases, we can identify some paradigmatic cases [see 3.2]:

- the case of *_farmer_*, transparent, is linear: for $\alpha = 700$, the decomposition correctly isolates the suffix *er_*; for $\alpha = 30000$ the root *_farm* is also identified as a single chunk; for $\alpha = 150000$ the word enters as an entire chunk in $morph$. Equally linear is the evolution of *_fruitless_*, opaque, with the difference that it is

H

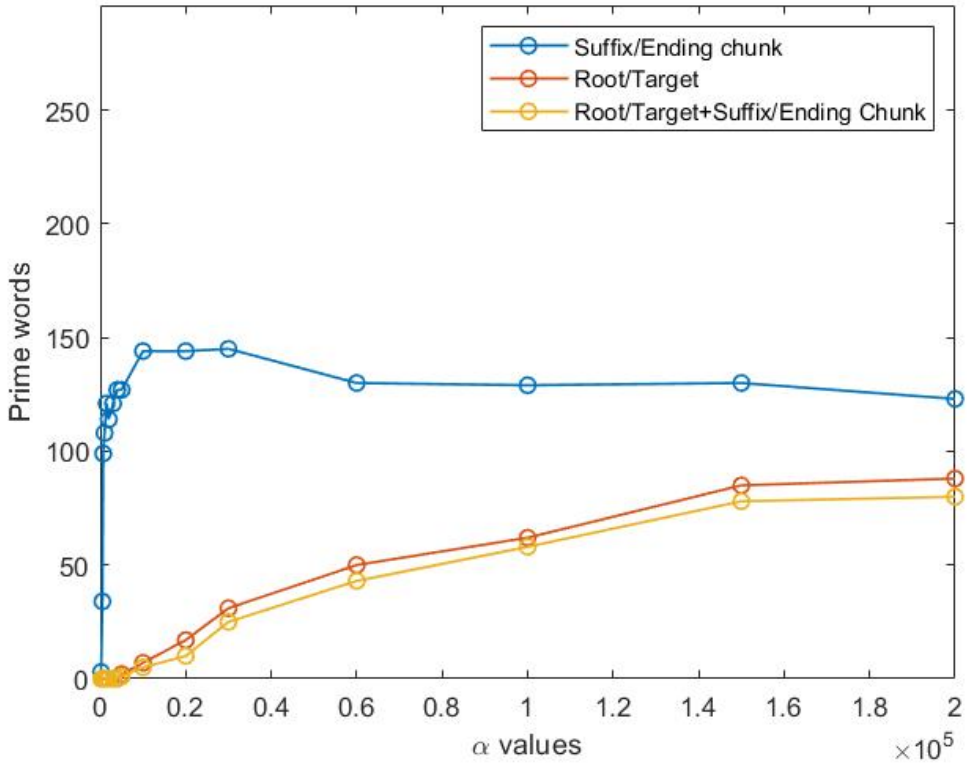

Figure 3.1: Number of primes for which the algorithm is able to chunk the suffix/ending bunch of letters or the root/target or both

not integrated as a whole word, but remains decomposed into *_fruit-less_* even for very high α values.

- in the case of *_alarming_*, transparent, the *ing_* suffix is correctly identified for small α values, since *ing_* is among the first chunks to be added to $morph$; from a certain value of α, around 5000, the *ming_* chunk is preferred in the decomposition; for $\alpha = 100000$, *_alarming_* is decomposed into two units *_al-arming_*, but these do not correspond to root+suffix. However, when the root *_alarm* becomes part of $morph$, then the decomposition is corrected in *_alarm-ing_*, and the chunk *ing_* is again correctly found. A similar evolution occurs for *_crafty_*, where the suffix *y_* is chunked in one unit for very low α, then replaced by *ty_*, and recovered at high α with the integration in $morph$ of the root *_craft*.

- in the case of *_united_*, opaque, the algorithm directly switches from a morphologically incorrect decomposition *_uni-ted_* (up to $\alpha = 20000$) to the inclusion of the whole word in $morph$ (for $\alpha = 30000$); the same happens for *_northern_*, transparent, for which the *_north-ern_* decomposition does not occur for any α value.

- in the case of *_bearded_*, transparent, as well as of *_fleeting_*, opaque, the decomposition is not correct even for $\alpha = 200000$, but the chunks *ting_*, *_be* and *arded_* are preferred, as they are useful in the decomposition of many other words in the dictionary.

What happens with the primes in the **form** condition must be evaluated differently: in this case, prime and target are related only because of an orthographic overlap, without any morphological relation, not even apparent. The priming effects observed in the form condition are not significant, therefore the presentation of a prime which is only orthographically related to the target seems not to facilitate the processing of the target more than any control word. We are therefore interested in checking whether the algorithm is able to capture the difference between form and transparent/opaque/semi-transparent conditions: if so, it would be reasonable to assume that morphemes can be identified by their statistical distribution in language, and not necessarily because they are associated with a particular meaning, and that they therefore differ from all other possible combinations of letters for purely orthographic reasons, which are independent of any linguistic structure. In the form condition, there are two trends of evolution of the decomposition that we can observe [see 3.2]:

- there are cases in which the algorithm actually decomposes the prime into the target + ending letters (e.g. *_fluid_*, the prime, decomposed into *_flu+id_*, and *flu* was the target); we note however that this often happens when the pairs classified as belonging to the form condition should be more properly considered as opaque/semi-transparent: this is the case for example of *colony-colon*, where *y_* is a suffix, or *accordion-accord*. Nevertheless, we have chosen to keep the labels of the original experiments, in order not to alter their conclusions.

- As in the case of *_cardiac_-car* pair, the target *_car* is identified for $\alpha = 5000$,

but since *diac_* is never recognised as a chunk, for very high chunk values the decomposition changes into *_card-iac_*.

prime	**target**	**condition**	α=500	α=700	α=5000	α=20000	α=30000	α=100000	α=150000	α=200000
alarming	alarm	transparent	_a-l-a-r-m-ing_	_a-l-a-r-m-ing_	_al-ar-ming_	_al-ar-ming_	_al-ar-ming_	_al-arming_	_alarm-ing_	_alarm-ing_
bearded	beard	transparent	_-b-e-a-r-d-e-d_	_be-a-r-d-ed_	_be-ard-ed_	_be-ar-ded_	_be-ar-ded_	_be-ard-ed_	_be-arded_	_be-arded_
farmer	farm	transparent	_-f-a-r-m-er-_	_-f-a-r-m-er_	_f-ar-m-er_	_far-m-er_	_farm-er_	_farmer-_	_farmer_	_farmer_
northern	north	transparent	_-n-o-r-th-er-n-_	_-n-o-r-ther-n-_	_no-r-ther-n_	_northern_	_northern_	_northern_	_northern_	_northern_
crafty	craft	opaque	_-c-r-a-f-t-y_	_-c-r-a-f-t-y_	_c-ra-f-ty_	_cr-aft-y_	_cra-f-ty_	_-craft-y_	_craft-y_	_craft-y_
fleeting	fleet	opaque	_-f-l-e-e-t-ing_	_-f-l-e-e-t-ing_	_f-le-e-ting_	_f-le-e-ting_	_f-le-e-ting_	_f-le-e-ting_	_fle-e-ting_	_fle-e-ting_
fruitless	fruit	opaque	_-f-r-u-i-t-l-e-s-s_	_-f-r-u-it-l-e-s-s_	_fr-u-it-l-ess_	_fr-u-it-less_	_fr-u-it-less_	_fruit-less_	_fruit-less_	_fruit-less_
united	unit	opaque	_-u-n-i-t-e-d_	_-u-n-it-ed_	_-un-i-ted_	_uni-ted_	_united_	_united_	_united_	_united_
cardiac	car	form	_-c-a-r-d-i-a-c-_	_-c-a-r-d-i-a-c-_	_car-di-ac-_	_car-di-ac-_	_car-di-ac-_	_card-i-ac-_	_card-i-ac-_	_card-i-ac-_
colony	colon	form	_-c-o-l-o-n-y_	_-c-o-l-on-y_	_co-l-on-y_	_colo-ny_	_col-on-y_	_colon-y_	_colon-y_	_colon-y_
fluid	flu	form	_-f-l-u-i-d_	_-f-l-u-i-d_	_f-l-u-i-d_	_fl-u-id_	_fl-u-id_	_fl-u-id_	_flu-id_	_flu-id_

Table 3.2: Examples of decomposition for different α values

In order to evaluate the performance of the algorithm, it can be even more indicative from our point of view, to subdivide the pairs into the three conditions, and compute the percentage of primes in each condition for which the algorithm is able to recognise the suffix, the root, or both: to have something compatible with human performance, we should have a higher decomposition percentage for transparent and opaque/semi-transparent words and much lower for pairs in the form condition .

- In the case of the **suffixes** 3.2, the **transparent** and **opaque/semi-transparent** conditions are not distinguished by the algorithm at the beginning, and the percentage of decomposed suffixes is even higher for pairs in opaque condition: after all, the primes that fall into the opaque category are precisely those that present a bunch of letters that is orthographically a suffix, but that does not play its function from a grammatical or semantic point of view. It is interesting to note that as α increases and the roots begin to be integrated, the tendency is reversed and the differences between the two conditions becomes relevant, with an higher percentage of correctly decomposed transparent primes: for $\alpha = 150000$, the difference in the number of correctly decomposed suffixes in the two conditions is statistically significant at the 5% probability level ($\chi^2 = 4.39$, p-value$= 0.036 < 0.05$).

  The number of pairs in the **form** condition for which the algorithm is able to isolate the ending bunch of letters is instead considerably lower than the transparent case for every α value (e.g., for $\alpha = 30000, \chi^2 = 15.33$, p-value$= 0.0001 < 0.05$; for $\alpha = 150000, \chi^2 = 14.57$, p-value$= 0.0001 < 0.05$): even without having an explicit notion of morphology, the algorithm actually shows a sensitivity to the morphological components of complex words with respect to the other groups of random letters that characterise the prime-target relation in the form condition. As far as suffixes are concerned, **opaque** and **form** conditions are well separated for small α values, while the difference becomes less clear with higher α.

- Looking instead at the number of primes in which the algorithm identifies **roots** in a single chunk, the differences between the conditions are never statistically significant 3.3: in fact, even in the form condition the target is perfectly embedded in the prime, and it can be recognised as a single chunk if it is particularly frequent. The approach that explains morpho-orthographic chunking as due to embedded

stem activation distinguishes the form condition from the others on the basis of a process of lateral inhibition between word representations that our algorithm could in no way take into account.

- Finally, the results that are more in line with the experimental observations are obtained if we consider the words for which the algorithm succeeds in identifying **both** the root and the suffix 3.4. The best performance is obtained by setting $\alpha = 150000$: in this case, the difference between **transparent** and **form** condition is again statistically significant at the 5% probability level ($\chi^2 = 13.48$, p-value$= 0.0002 < 0.05$), and so is the difference between **opaque** and **form** ($\chi^2 = 6.30$, p-value$= 0.02 < 0.05$), while the **transparent** and **opaque** conditions are indistinguishable ($\chi^2 = 1.87$, p-value$= 0.17 > 0.05$).

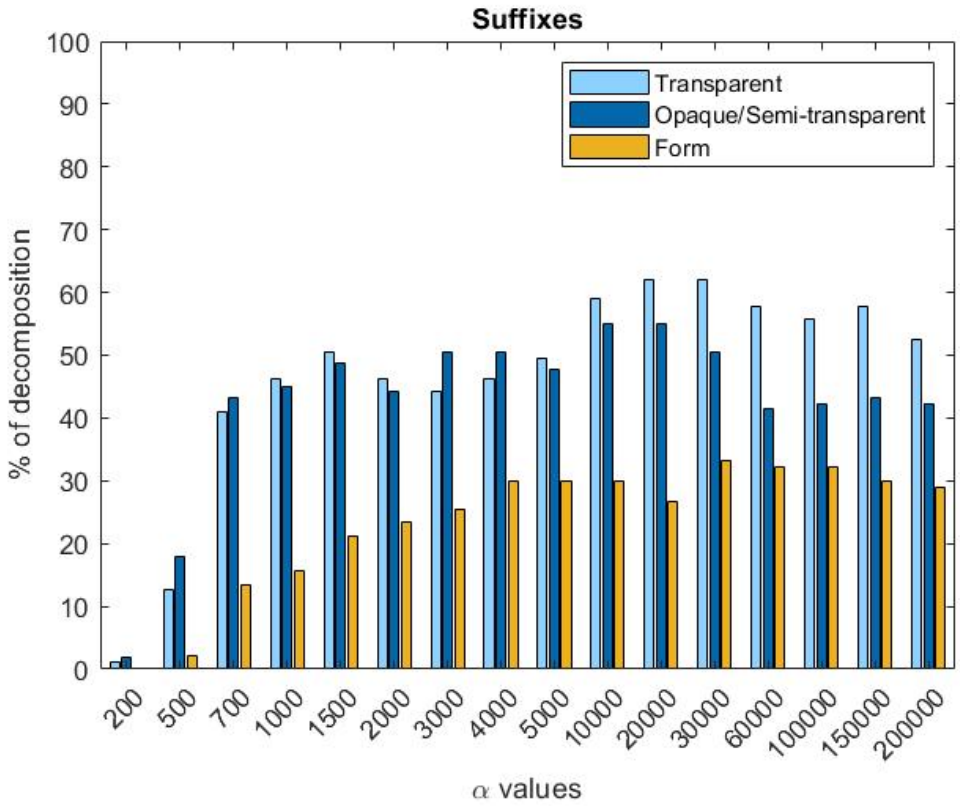

Figure 3.2: Percentage of primes, separated by condition, for which the algorithm is able to correctly chunk the suffix/ending bunch of letters

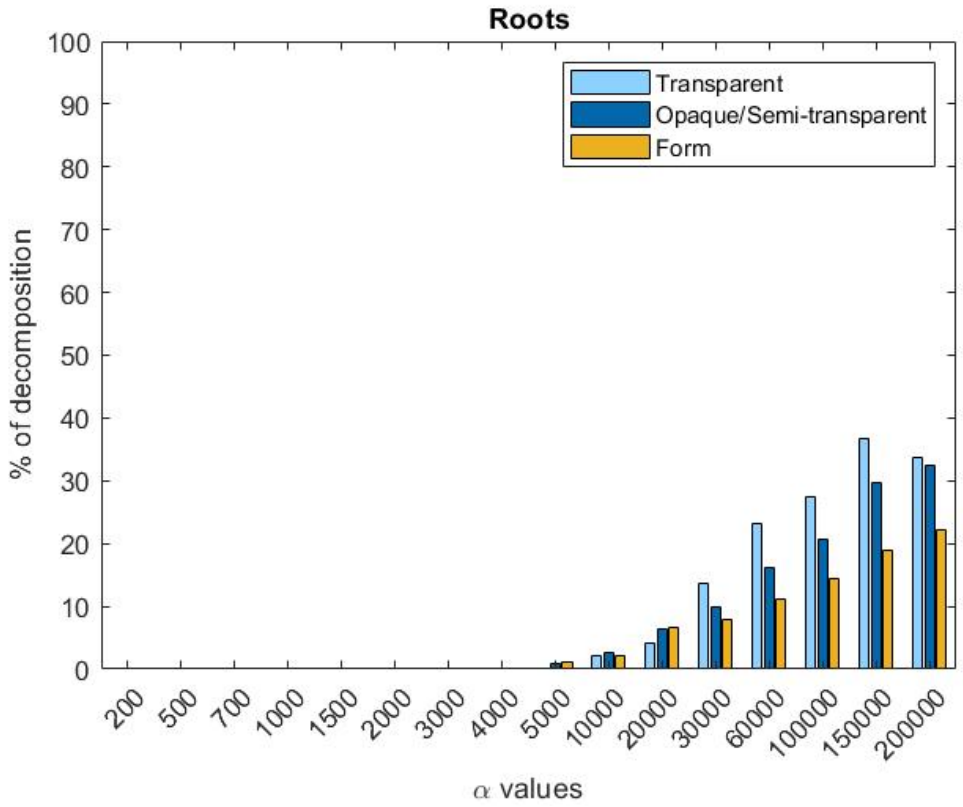

Figure 3.3: Percentage of primes, separated by condition, for which the algorithm is able to correctly chunk the root/target

Another type of analysis on the dataset at our disposal that could be interesting takes into account the **PEM**, priming effect magnitude, i.e. the difference between the reaction time measured for the control prime-target pair and the reaction time for the related prime-target pair: abandoning the classically assigned labels of the different conditions, the PEMs should reflect in a more direct and realistic way the effect of the processing of the prime on the processing of the target, without resorting to classifications introduced a posteriori. We therefore try to investigate whether the primes that the algorithm manages to decompose correctly actually show a higher PEM, and so an effective facilitation in processing, compared to the words for which the algorithm finds an alternative decomposition, or which it stores as whole units.

Choosing again $\alpha = 150000$, the value for which we have the most significant results in the decomposition of primes according to condition, we compare the average

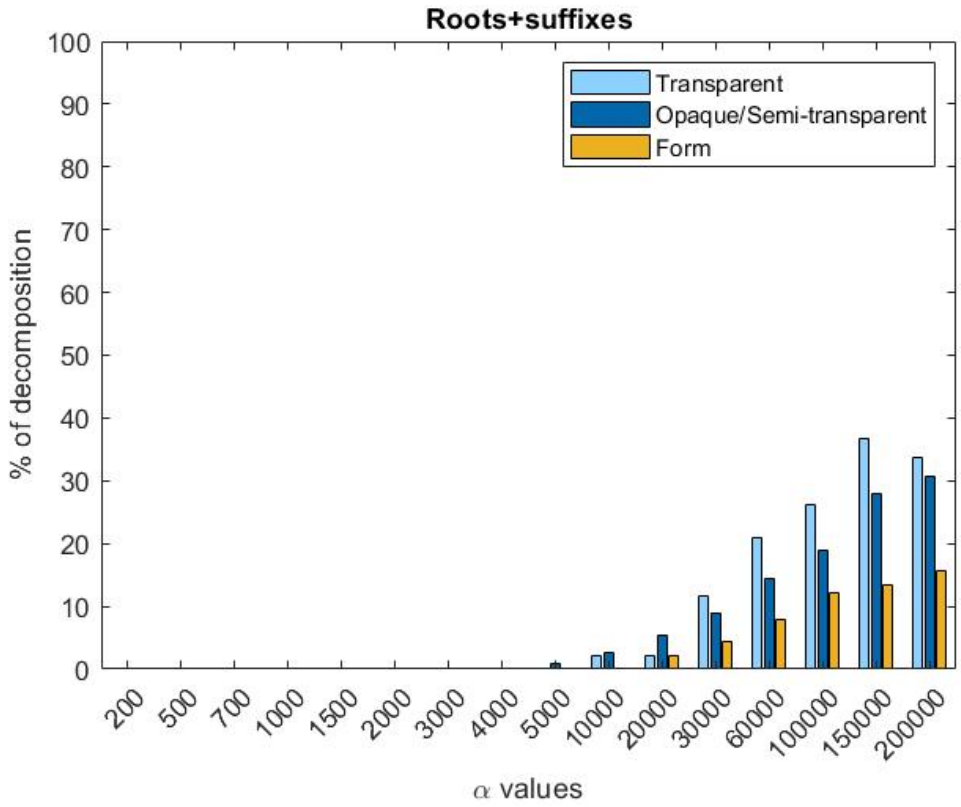

Figure 3.4: Percentage of primes, separated by condition, for which the algorithm is able to correctly chunk both the suffix/ending letters and the root/target

PEM of the words chunked in two units (target/root + suffix/ending letters) with the average PEM of all the others: the average PEMs are actually equivalent in their error interval ($\overline{PEM}_{both} = 25 \pm 4\,ms$, $\overline{PEM}_{all \setminus both} = 26 \pm 3\,ms$) [3.5]. Again, no statistically significant difference arises if we compare the average PEM for words for which the algorithm correctly chunks the suffix/last letters, with the average PEM for those for which it fails (whole words, suffix letters separated into different chunks, or suffixes integrated into longer chunks) [3.6]: $\overline{PEM}_{suff} = 28 \pm 4\,ms$, $\overline{PEM}_{all \setminus suff} = 24 \pm 4\,ms$. The same is true considering the primes for which the algorithm identifies the roots/targets compared to the others [3.7] ($\overline{PEM}_{root} = 23 \pm 4\,ms$, $\overline{PEM}_{all \setminus root} = 27 \pm 4\,ms$). The box plots of the distributions of the PEM values in each case help visualize that they do not show relevant differences. Note that we suppressed the display of the outliers in order to be able to observe more clearly the

possible differences between the distributions, since they are not particularly interesting to us.

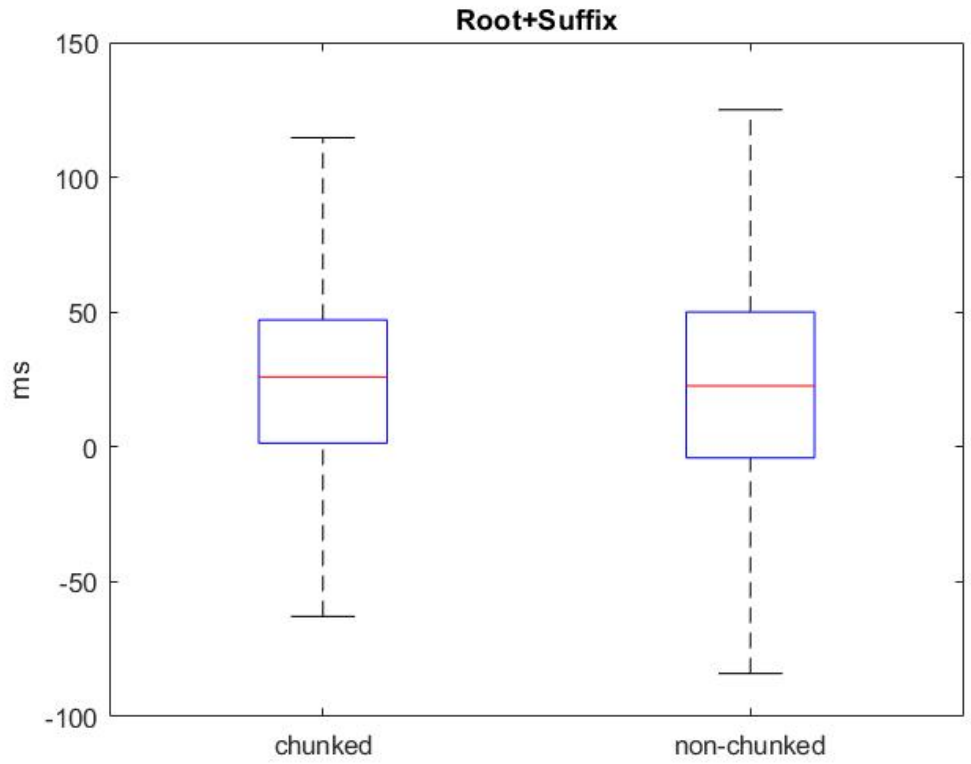

Figure 3.5: A box plot comparing the PEMs of the primes chunked as root+suffix and the PEMs of all the other primes.

A bit more noticeable is the difference between the distribution of the PEMs for prime-target pairs where the prime is integrated into $morph$ as a single chunk, compared to the PEMs of the correctly decomposed words (root+suffix) 3.8: since the words that become part of $morph$ earlier are the most frequent ones, considering the token frequency, this difference supports the hypothesis that they develop an independent representation that does not go through morpho-orthographic processing. But also in this case the difference between the average values is not statistically significant at a significance level of 0.05, since the standard errors of the means are high ($\overline{PEM}_{suff} = 28 \pm 4\,ms$, $\overline{PEM}_{words} = 15 \pm 7\,ms$, $Z = 1.24$, p=0.21).

In fact, working on the PEMs of individual pairs, and not by category (as for the

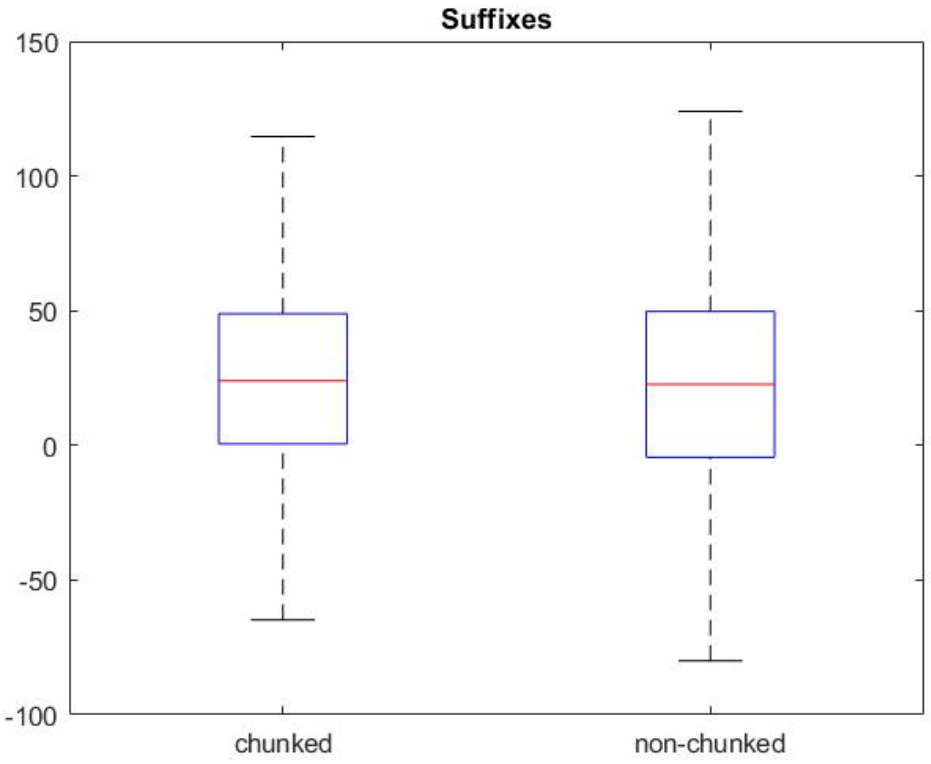

Figure 3.6: A box plot comparing the PEMs of the primes for which the algorithm correctly chunks the suffix/ending bunch of letters and the PEMs of all the other primes

conditions), can be risky for priming experiments: the variability is very high, and one cannot ignore the error on the individual data. It may therefore be appropriate to develop a more in-depth analysis on a new experiment, where conditions are homogeneous (in this case, data are collected from different datasets, so for example the SOA is not always the same) and the related prime-target and control prime-target pairs are constructed in such a way as to investigate the relevance of the storage trade-off principle in the chunking of prime stages.

Some notes on the composition of the dataset

1. The dataset is purposely constructed to cover a fairly wide range of suffixes, but some of them are repeated the same in different primes (*-er* in *archer*, *thriller*,

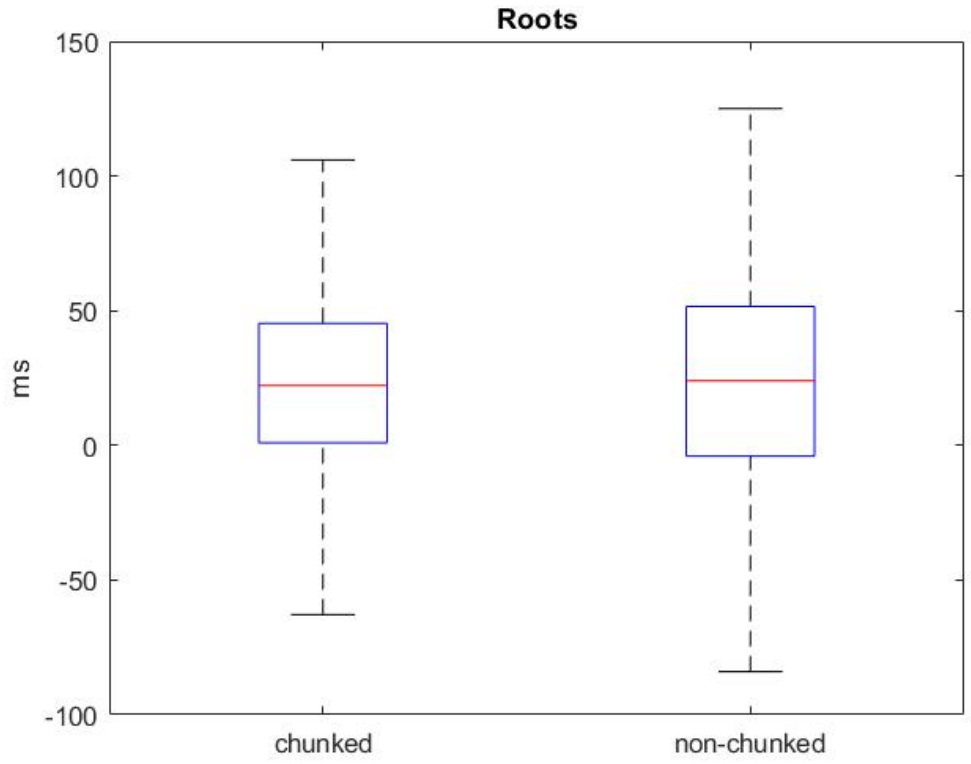

Figure 3.7: A box plot comparing the PEMs of the primes for which the algorithm correctly chunks the root/target and the PEMs of all the other primes

viewer, *whisker*...), and this explains why the number of correctly chunked suffixes immediately becomes very high (for $\alpha = 700$, the chunk *-er* is integrated in $morph$ and it is used as suffix/ending chunk in 11 primes); when α grows, however, the decomposition of these words is no longer necessarily done in the same way (for $\alpha = 1000$, we find *_a-r-c-her_*, but *_th-r-i-ll-er_*), so they may not have the same effect on the calculation of correctly decomposed words.

2. Our algorithm would work in the same way for prefixes, which are not present in the primes of the dataset, but which turned out to become part of $morph$ simultaneously with the derivational suffixes [see the analysis of the chunks in $morph$ in 3.10.1].

3. Among the related prime-target pairs in the transparent, opaque or semi-transparent

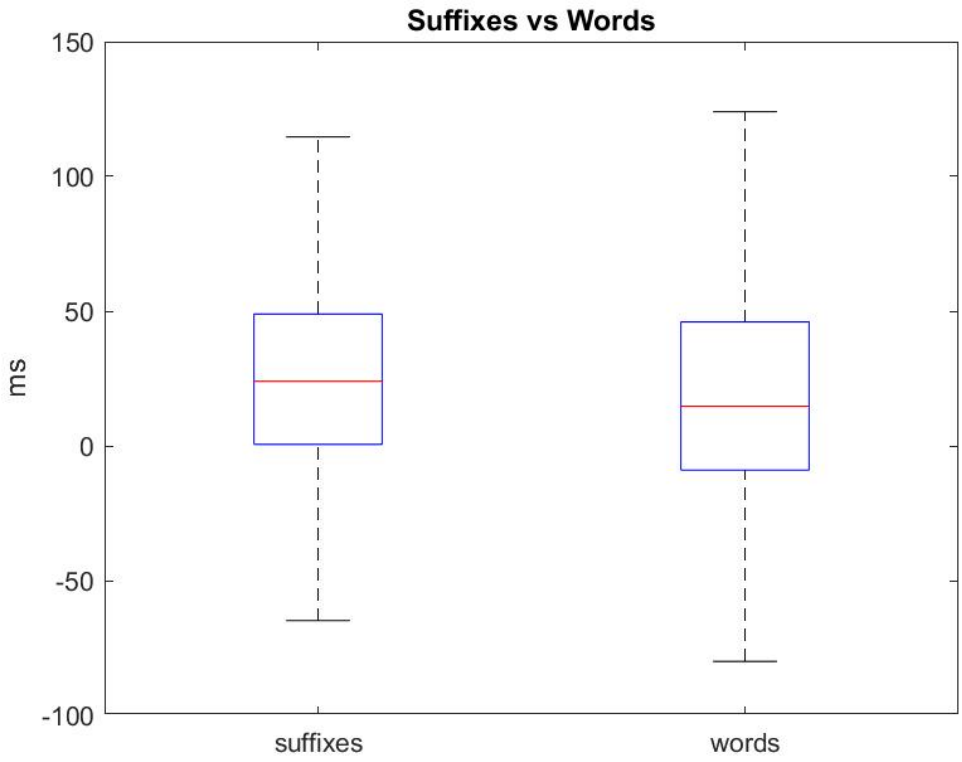

Figure 3.8: A box plot comparing the PEMs of the primes chunked as root+suffix and the PEMs of the primes integrated in $morph$ as single chunks

condition, 11 are nor fully parsable into stem+suffix: in pairs like *_baker_-_bake_*, the *e* is shared between the stem and the suffix; in *_awe_-_awful_*, the *e* of the stem is suppressed in the derived word; in *_swimmer_-_swim_*, the duplicated final consonant *_m_* of the stem emerges in the derivational process. McCormick et al. [60] reported standard morphological priming in all these cases, a result that suggests that readers are able to overcome the orthographic variations that arise in building complex words from morphemes. In doing so, meaning could play a fundamental role: indeed, Crepaldi et al. [61] showed that there is a significant priming effect even for irregular inflected words when preceded by the base form (*fell* does facilitate *fall* more than orthographically-matched control word *fill*), an evidence which excludes that morpho-orthographic decomposition is completely blind to semantics. Obviously, our algorithm is not able to treat these

pairs correctly: if it manages to detect the suffix, the root will not be correctly chunked (e.g., *swimmer*= *swimm-er*), and vice versa (*crabby*= *crab-by*).

Chapter 4

Conclusions

The chunking mechanism seems to play a fundamental role in many processes of perception and learning: although the fields where the concept of chunk can be found are very different, a chunk can be generally defined as "a collection of elements having strong associations with one another, but weak associations with elements within other chunks" [35]. The formation of independent chunks for particularly cohesive letter clusters in written language processing may be the way in which literate adults become able to read very quickly and almost without errors, despite the fact that reading, given its relatively recent origin, cannot be part of our biological endowment. The origin, nature and size of the chunks are, however, still far from clear.

Evidence in the morphological literature seems to confirm that the brain is indeed sensitive to sub-word letter chunks, the morphemes, the smallest meaning-bearing units in language [17]. However, the hypothesis that semantics could not be essential to characterise the chunks identified in processing is only quite recent: masked priming experiments showed that in the first stages of written word recognition readers are led to identify as morphemes also those groups of letters that have the same orthographic appearance but no morphological function (i.e., the chunk *-er* in *corner* induces a pseudo-morphological decomposition in *corn-er* that mimics the one of, e.g., *dealer*, where *deal* is a stem and *er* is a suffix). This result suggests that the role of morphemes in the chunking mechanism may be not necessarily related to their function in form-to-meaning mapping, but rather to their nature as high-frequency clusters of co-occurring letters: indeed, it has been shown in several cognitive domains that the brain is able

to detect the probabilistic pattern of the learning material to which it is exposed [29]. Consequently, morphemes, as intermediate units in the written word process, could be joined by other chunks that, even without any linguistic function, represent elements of regularity in the distributional pattern of letters in the language, and that could be identified with a language-agnostic cognitive mechanism, able to capture the statistical regularities of the written language.

The problem that naturally arises if we suppose that the brain can resort to intermediate units between letters and words is that of solving the trade-off between storing a small number of units (in the extreme case, only the letters), which would require a high computational effort, and minimizing the processing of the words, which implies an increase in storage, to the point of storing each word as an independent unit. In order to understand which chunks could resolve the tension between these two tendencies, in this thesis we developed a model in which the problem is formally translated into finding the set of chunks that allow us to parsimoniously decompose all the words of the language. This set corresponds to the one that minimises a one-parameter objective function featuring two terms, one pushing for the storage of the smallest possible number of chunks, the other tending to reduce the chunks needed to decompose each word of the corpus. The introduced parameter determines the relative weight of the two players in the competition.

This is the only parameter we choose to include, since our model is not designed with the ultimate goal of reproducing experimental data as accurately as possible, but rather to isolate this single principle, and study the nature and evolution of the set of chunks that arises in accordance with it. As every computational model of visual word recognition, with respect to purely theoretical linguistic models, it forces us to be explicit in every choice we make; this aspect, which has the advantage of pushing researchers to question many aspects that are not immediate from a theoretical point of view, is seen as a limitation by detractors of computational models, because it makes it necessary to make some arbitrary or cognitively irrelevant decisions. But are these decisions truly irrelevant from a cognitive point of view? In our case, the model is intended to be as simple as possible, precisely in order to minimise the possibility of introducing arbitrary elements that might steer the results in a desired direction. Whenever we were forced to make a choice (on the type of corpus, on the introduction

of parameters, on the need to provide a marker of the beginning and end of the words of the corpus, on the weight to be associated to the chunks), we tried to follow the hints from the experimental evidence available to date.

We have chosen not to introduce any linguistic clues that might arise from the mapping of the other levels in which words are interpreted and represented in the mental lexicon, the phonological and the semantic ones. Even in the updated version of the algorithm, the introduction of weights for the chunks does not change the rationale, because these are always related to the token frequency of the words in the corpus.

The other choices that we considered very important to include from the beginning, i.e., that the optimization should be performed on a realistic corpus and that chunks of different lengths should compete with each other, make the problem computationally difficult: the candidate sets of chunks are a super-exponential number (all possible combinations of chunks of length from 1 to M, where M is the maximum number of letters of the words in the corpus) and for each of them it would in principle be necessary to recompute the objective function, which involves the decomposition of every word in the corpus (upon cleaning, about 50000 words). So the algorithm proves to be interesting in the first place precisely because of the computational tricks adopted to make the problem tractable in a finite time: among the others, a clean cut of the set of chunks that could be potentially selected to become part of $morph$, in order to immediately exclude the ones that are so unusual that cannot contribute in any way to lowering the objective function; a great reduction of the computations required to obtain the new value of the objective function after every modification of the proposed optimal set; the translation of the problem of finding the parsimonious decomposition of each word in the corpus into a shortest path problem on a graph. Some consistency checks show that the results obtained in this way are robust and repeatable.

These choices have the advantage of making the results immediately interpretable: analysing the chunks we obtain in the optimal set for increasing values of α, we have seen that affixes soon become part of it, starting from inflectional suffixes (*-ing*, *-ed*, *-er*, *-s*), to continue with derivational suffixes (*-al*, *-ion*) and prefixes (*pro-*, *in-*, *re-*). This confirms the hypothesis that these morphological elements do indeed have a statistical distribution that makes them identifiable even in the absence of any semantic clues. At the same time, however, they are not the only units that meet this requirement:

together with some harmless units, characteristic of the English language, which could be effectively identified as chunks in order to speed up the processing (*-ould* of *could*, *would*, *wh-* of *where*, *which*, *-ought* of *thought*, *bought*, *sought*...), others would tend to replace morphemes in word decomposition, as α increases (*-ming* is used instead of *-ing* in *alarming*, *-ted* is used instead of *-ed* in *adopted*). This type of chunks emerges simultaneously with the roots, which the algorithm struggles to integrate as independent units: a meaningless chunk of the type *-arded* (e.g., in *regarded*, *discarded*) is as useful in the decomposition as the root *author-*.

From the point of view of the algorithm, therefore, there are many letter clusters which are suitable candidates for the role of chunks in processing: the fact that affixes are actually among these suggests that a parsimonious chunking of this type could actually be used during learning stages to select the units which can be considered convenient from the point of view of processing efficiency, but these could then be subjected to the examination of the other levels involved in reading. The association of a meaning, or a precise sound, to certain chunks might cause these to be preferred to others because of their validity at higher levels of processing.

However, we do not have enough experimental evidence to exclude that the units that are included in the best set $morph$ do not have any relevance in the processing of written language: the priming experiments conducted so far have always aimed at investigating the role of morphemes, and the presence of priming effects, i.e., a facilitation in processing, in the case in which a simple word, the target, is preceded by a morphologically complex prime containing the target, with respect to the case in which the prime is completely unrelated, or only orthographically related (e.g. *employer-employ* versus *addition-employ*). It might be interesting to construct a masked priming experiment aimed at investigating the possibility that chunks such as *-ming*, *-ting*, *-ter*, or *wh-*, *-ould*, *-ough* actually show priming effects, if not equal to those of real morphemes, at least greater than those due to other random clusters of letters. This could indirectly provide an answer on the effective role of meaning at least in the first stages of visual word recognition: if these units do not induce any priming effect, then semantic could be relevant already at this level, or at least it could contribute to crystallise those chunks that are actually convenient also for grasping the meaning of words.

It is instead more difficult to think that the roots could have an origin that is completely independent from the semantics: even if the algorithm proved to be able to select a great percentage of them for high α values, their probabilistic distribution seems not to make them immediately identifiable. Obviously, it cannot be excluded that other statistical principles than the one considered here come into play. There is always the possibility that the characterization of morpho-orthograpy given by our algorithm is not the right one: even if our approach is well-founded, because it is principled (i.e., based on the tension between storage and computation) and formal, it does not necessarily have to be the neurally correct one.

The asymmetry between affixes and stems also emerges clearly from the analysis of the masked priming experiment dataset on which we evaluated the performance of the algorithm: the first roots are correctly identified for a rather high α value, when many words start to be stored as integers in the best set. This can be an interesting cue to understand the kind of mechanism that underlies morpho-orthographic decomposition: between the two types of approaches proposed in the literature, the one that considers that priming effects are due to the automatic chunking of affixes (affix-stripping) and the one that considers that the decomposition is determined by the roots, since they are more informative (embedded stem activation), the results of the performance of this linguistic-agnostic algorithm lean towards the former. That is, our analysis might suggest that if the morpho-orthographic decomposition is driven by the chunking of clusters of frequently co-occurring letters, then the second approach would be more likely.

Another interesting result that we find is that, by increasing α up to a certain optimal value, the algorithm is able to distinguish the different priming conditions, since it is more likely to decompose in target+ending chunk those primes that are in transparent or opaque/semi-transparent condition, i.e. that have an apparent or real morphological relation with the target (e.g. *dealer-deal*, *archer-arch*), with respect to those that are in form condition, i.e. in an only orthographic relation with the prime (e.g. *scandal-scan*). As we said, masked priming experiments showed significantly greater priming effects in both the transparent or opaque/semi-transparent conditions than in the form condition, and our algorithm, that has no linguistic knowledge, seems to reproduce human brain sensitivity to the morphological appearance of words: this supports the hypothesis of

the probabilistic nature of morpho-orthographic decomposition, related to the role of morphemes as cohesive recurrent clusters.

Much less informative is instead the analysis of the relationship between the measure of the priming effects for single related prime-target pairs and the ability of the algorithm to decompose the prime: the variability that characterizes priming experiments might require a deeper analysis, in which all possible effects that accelerate or slow down the processing of a word are taken into account. Such an analysis would make sense on an experiment specifically designed to investigate the role of chunks in $morph$ in processing, whereas the information we have on the dataset is too approximate. Indeed, the initial choices adopted for designing the algorithm allow us to potentially use a technique of comparison with the experimental data that could be very accurate, and that does not require the introduction of arbitrary and artificial elements: if we had used an artificial language for the objective function optimization, we could have introduced arbitrary choices in constructing it, with the risk of not reproducing the probabilistic patterns typical of English, and we would have automatically eliminated the linguistic component in the comparison with human performance.

4.1 Further applications and possible improvements

If it is true that the α parameter could have a psychological interpretation related to the progressive mastering of literacy, an element which represents a clear limitation in the realism of the algorithm is the choice of optimising the objective function on the entire corpus, independently of the value that α assumes. A possible improvement could consist in increasing the corpus as α increases: in this way, we could try to reproduce the experience accumulated by the readers. However, it would not be obvious which criteria to choose in order to integrate new words into the corpus: we could for instance choose to provide the algorithm with words according to their frequency, from the most frequent ones, already available for low α values, to the rarest ones, as α increases, or we could imagine randomly sampling a fixed number of words for each α value, with a probability proportional to their frequency in the corpus.

Introducing this kind of dynamic corpus could also be useful to improve the definition of the weights of the chunks: starting from an equal weight for all the chunks, we

could calculate each time their productivity (the number of times a chunk is used in the shortest path decomposition), and use it to redefine the weights, so that the most productive chunks becomes the lightest. Such a choice, which would reflect the usefulness of the chunks more than the token frequency, is instead not feasible using a statistical corpus.

The introduction of weights could represent an easy way to include extra-orthographic clues: without changing the architecture of the algorithm, we could decrease the weights of chunks that prove to be relevant in other levels of language processing, to study whether this could bring the performance of the algorithm closer to human performance. In spite of the choice we made not to use any information other than that provided by the corpus, the algorithm proves to be suitable for integrating other linguistic elements in a controlled way, and the possibility of being easily generalized is fundamental for any computational model that tries to explain very complex mechanisms.

Considering the structure of the algorithm, it might also be interesting to reflect on the possibility of considering simultaneously different paths in the decomposition of a word. Again, the introduction of the weights is in this sense fundamental: on the one hand, because it reduces the number of equivalent decomposition paths, among which the algorithm would arbitrarily choose; on the other hand because it allows to assign to each decomposition path a certain weight. In our project, we have considered in word processing only the path with the lowest weight, but it could be that also other decomposition paths with a similar weight play a certain role, and they might even be preferred when the target is preceded by a certain prime. It might therefore be interesting to abandon a deterministic view in order to explore the possibility that different decompositions coexist and are chosen in proportion to their probability, the higher the lower their weight, according to a quantum perspective. Moreover, this idea could more likely represent what happens to human processing, where a deterministic processing is likely to be excluded.

Also from an experimental point of view, there are many possible applications that could be interesting from our point of view; besides the need to design a masked priming experiment in which prime and target are chosen in such a way as to have a more precise comparison with the performance of the algorithm, there are two other lines that could be investigated:

1. The algorithm works with any corpus in which the inputs are ordered sequences of symbols: one could then use pseudoletter strings of an artificial language, in a similar way to what was done by Lelonkiewicz et al. [32], using the same corpus for the familiarization for the participants and for the algorithm. One could then form new pseudowords containing the chunks selected in the best set by the algorithm, tuning the α parameter, and see if these tend to be effectively recognised as belonging to the artificial language (in the cited experiment, the role of some sort of "affixes" was investigated, that are sub-strings of 3 or 4 symbols that were repeated the same in different pseudowords). This would represent a backward step with respect to the use of the algorithm in our project, because the lexicon would certainly be smaller and easier to handle, and because we would exclude a priori any linguistic clue in human readers; however, it would allow us to establish with certainty whether the principle of minimising the storage-computation trade-off is actually used at least in the first phases of the learning of a new written language.

2. In order to investigate the evolution of chunks as proficiency increases, it might be interesting to carry out masked priming experiments on developing readers, perhaps for different ages or grades, similarly to what was done in the experiments conducted by Quémart et al. [62] and by Beyersmann et al. [63]: it may be possible, for example, that those non-morphological chunks that appear as α increases may show greater priming effects for non-proficient readers, because the crystallization of chunks based on their relationship with other linguistic levels could not have yet been stabilized for them, or because they may detect roots with greater difficulty (in the algorithm, we see that errors in the detection of chunks of the type *-ted* in *adop-ted* are corrected when roots begin to be integrated as well, *adopt-ed*). In the same way, we could understand if the priming effects for the most recurrent words actually reduce as the proficiency increases, showing a tendency to process them as whole units, or if the readers always go through a decomposition (in this case, the probabilistic interpretation of the different paths could be an explanation).

Glossary

$chunk$	basic unit stored by the algorithm and used to process new words
L	objective function to minimize
N	number of stored chunks
$\bar{n}$	average number of chunks per word
N_{tot}	total number of words in the corpus
$freq_i$	normalized frequency of the ith word in the corpus
$nchunk_i$	(minimum) number of chunks needed to process the ith word
$reservoir$	set of all proposed chunks
M	number of letters of the longest word in the corpus
$morph$	set of chunks that minimizes the objective function
$trimmed\ res.$	the reduced reservoir after eliminating the less frequent chunks
$\bar{w}$	average weight of the words in the corpus
$wchunks_i$	sum of the weights lightest chunks used to decompose the ith word

Bibliography

[1] C. Woods. *Visible Language*. The University of Chicago: editor, 2010.

[2] M. Brysbaert. «How many words do we read per minute? A review and meta-analysis of reading rate.» In: *Journal of Memory and Language* 109 (2019).

[3] S.R. Quartz and T.J. Sejnowski. «The neural basis of cognitive development: a constructivist manifesto.» In: *Behavioral and Brain Science* 20 (1997), pp. 537–556.

[4] S. Dehaene and L. Cohen. «Cultural recycling of cortical maps.» In: *Neuron* 56 (2007), pp. 384–398.

[5] S. Dehaene and L. Cohen. «The unique role of the visual word form area in reading». In: *Trends in Cognitive Sciences* 15 (2011), pp. 254–262.

[6] S. Dehaene, G. Le Clec'H, J.B. Poline, D. Le Bihan, and L. Cohen. «The visual word form area: a prelexical representation of visual words in the fusiform gyrus.» In: *Neuroreport* 13 (2002), pp. 321–325.

[7] C. Davis. «The Self-Organising Lexical Acquisition and Recognition (SOLAR) model of visual word recognition». PhD thesis. University of New South Wales, 1999. URL: `http://http://www.pc.rhul.ac.uk/staff/c.davis/Thesis`.

[8] J. C. Ziegler and U. Goswami. «Reading acquisition, developmental dyslexia, and skilled reading across languages: a psycholinguistic grain size theory.» In: *Psychological bulletin* 131(1) (2005), pp. 3–29.

[9] J. Grainger, S. Dufau, M. Montant, J. Ziegler, and J. Fagot. «Orthographic processing in baboons (papio papio).» In: *Science* 336(6078) (2012), pp. 245–248.

[10] D. Scarf, K. Boy, A. Uber Reinert, J. Devine, O. Güntürkün, and M. Colombo. «Orthographic processing in pigeons (Columba livia).» In: *Proceedings of the National Academy of Sciences* 113 (2016), p. 201607870.

[11] M. Riesenhuber and T. Poggio. «Hierarchical models of object recognition in cortex.» In: *Nature Neuroscience* 2 (1999), pp. 1019–1025.

[12] E. T. Rolls. «Functions of the primate temporal lobe cortical visual areas in invariant visual object and face recognition.» In: *Neuron* 27 (2000), pp. 205–218.

[13] J.J. DiCarlo, D. Zoccolan, and N.C. Rust. «How does the brain solve visual object recognition?» In: *Neuron* 73 (2012), pp. 415–434.

[14] F. Vinckier, S. Dehaene, A. Jobert, J.P. Dubus, M. Sigman, and L. Cohen. «Hierarchical Coding of Letter Strings in the Ventral Stream: Dissecting the Inner Organization of the Visual Word-Form System.» In: *Neuron* 55 (2007), pp. 143–156.

[15] J. L. McClelland and D. E. Rumelhart. «An interactive activation model of context effects in letter perception: I. An account of basic findings.» In: *Psychological Review* 88 (1981), pp. 375–407.

[16] H. Giraudo and M. Voga. «Measuring morphology: the tip of the iceberg? A retrospective on 10 years of morphological processing». In: *Carnets de Grammaire* 22 (2014).

[17] L. Bloomfield. *Language*. New York: Henry Holt, 1993.

[18] K. Rastle. «The Place of Morphology in Learning to Read in English». In: *Cortex* 116 (2018).

[19] Z. Shao and A. Meyer. «Chapter 6: Word Priming and Interference Paradigms». In: *Research methods in psycholinguistics and the neurobiology of language: a practical guide*. Ed. by A.M.B. de Groot and P. Hagoort. Wiley-Blackwell, 2017.

[20] K. Rastle. «Chapter 21 - Visual Word Recognition». In: *Neurobiology of Language*. Ed. by G. Hickok and S.L. Small. San Diego: Academic Press, 2016, pp. 255–264.

[21] R. F. Stanners, J. J. Neiser, W. P. Hernon, and R. Hall. «Memory representation for morphologically related words.» In: *Journal of Verbal Learning and Verbal Behavior* 18 (1979), pp. 399–412.

[22] K. Rastle, Davis M. H., Marslen-Wilson W.D., and L.K. Tyler. «Morphological and semantic effects in visual word recognition: A time course study». In: *Language and Cognitive Processes* 15 (2000), pp. 507–538.

[23] C. M. Longtin, J. Segui, and P. A. Halle´. «Morphological priming without morphological relationship.» In: *Language and Cognitive Processes* 18 (2003), pp. 313–334.

[24] K. Rastle, M. H. Davis, and B. New. «The broth in my brother's brothel: Morphoorthographic segmentation in visual word recognition». In: *Psychonomic Bulletin and Review* 11 (2004), pp. 1090–1098.

[25] S. Amenta, M. Marelli, and D. Crepaldi. «The fruitless effort of growing a fruitless tree: Early morpho-orthographic and morpho-semantic effects in sentence reading.» In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 41(5) (2015), pp. 1587–1596.

[26] K. Rastle and Davis M. H. «Morphological decomposition based on the analysis of orthography». In: *Language and Cognitive Processes* 23.7-8 (2008), pp. 942–971.

[27] C. M. Longtin and F. Meunier. «Morphological decomposition in early visual word processing.» In: *Journal of Memory and Language* 53 (2005), pp. 26–41.

[28] D. Crepaldi, K. Rastle, and C.J. Davis. «Morphemes in their place: Evidence for position-specific identification of suffixes». In: *Memory Cognition* 38 (2010), pp. 312–321.

[29] B. C. Armstrong, R. Frost, and M. H.. Christiansen. «The long road of statistical learning research: Past, present and future.» In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 372 (2017), p. 20160047.

[30] E. J. Gibson. «Perceptual learning and the theory of word perception.» In: *Cognitive Psychology,* 2(4) (1972), pp. 351–368.

[31] F. Chetail. «Reconsidering the role of orthographic redundancy in visual word recognition.» In: *Frontiers in Psychology* 6 (2005), p. 645.

[32] J. R. Lelonkiewicz, M. Ktori, and D. Crepaldi. «Morphemes as letter chunks: Discovering affixes through visual regularities». In: *Journal of Memory and Language* 115 (2020), p. 104152.

[33] R. H. Baayen, P. Milin, D. F. Đurđević, P. Hendrix, and M Marelli. «An amorphous model for morphological processing in visual comprehension based on naive discriminative learning.» In: *Psychological review* 118(3) (2011), pp. 438–481.

[34] D. C. Plaut and L. M. Gonnerman. «Are non-semantic morphological effects incompatible with a distributed connectionist approach to lexical processing?» In: *Language and Cognitive Processes* 15 (2000), pp. 445–485.

[35] F. Gobet, P. C. R. Lane, S. Croker, P. C-H. Cheng, G. Jones, I. Oliver, and J. M. Pine. «Chunking mechanisms in human learning». In: *Trends in Cognitive Sciences* 5 (2001), pp. 1364–6613.

[36] J. R. Saffran, E. L. Newport, and R. N. Aslin. «Word segmentation: The role of distributional cues.» In: *Journal of Memory and Language* 35 (1996), pp. 606–621.

[37] P. Perruchet and A. Vinter. «PARSER: A model for word segmentation.» In: *Journal of Memory and Language* 39(2) (1998), pp. 246–263.

[38] M. Adams. «What good is orthographic redundancy?» In: *Perception of Print*. Ed. by H. Singer and Tzeng O. J. L. Hillsdale, NJ: Erlbaum, 1981.

[39] C. Burani, S. Marcolini, M. De Luca, and P. Zoccolotti. «Morpheme-based reading aloud: Evidence from dyslexic and skilled Italian readers.» In: *Cognition* 108 (2008), pp. 243–262.

[40] K. Patterson and J. Kay. «Letter-by-letter reading: Psychological descriptions of a neurological syndrome.» In: *Quarterly Journal of Experimental Psychology* 34A (2005), pp. 411–441.

[41] J. Hasenäcker, P. Schröter, and S. Schroeder. «Investigating developmental trajectories of morphemes as reading units in German.» In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 43(7) (2017), pp. 1093–1108.

[42] V. Robinet, B. Lemaire, and M.B. Gordon. «MDLChuker: A MDL-Based Cognitive Model of Inductive Learning». In: *Cognitive Science* 35 (2011), pp. 1352–1389.

[43] *MATLAB version 9.9.0.1467703 (R2020b)*. The Mathworks, Inc. Natick, Massachusetts, 2020.

[44] W.J.B. Van Heuven, P. Mandera, E. Keuleers, and M. Brysbaert. «Subtlex-UK: A new and improved word frequency database for British English.» In: *Quarterly Journal of Experimental Psychology* 67 (2014).

[45] M. Brysbaert, M. Buchmeier, M. Conrad, A.M. Jacobs, J. Bölte, and A. Böhl. «The word frequency effect: A review of recent developments and implications for the choice of frequency estimates in German». In: *Experimental Psychology* 58 (2011), pp. 412–424.

[46] A. Kilgarriff. *BNC database and word frequency lists.* 2006. URL: `http://www.kilgarriff.co.uk/bnc-readme.html`.

[47] T. H. Cormen, C. E. Leiserson, Rivest R. L., and C. Stein. *Introduction to Algorithms*. MIT Press, 2009.

[48] J. Goldsmith. «Unsupervised learning of the morphology of a natural language.» In: *Computational Linguistics* 27(2) (2001), pp. 153–198.

[49] D. Crepaldi and S. Amenta. «Morphological Processing as We Know It: An Analytical Review of Morphological Effects in Visual Word Identification.» In: *Frontiers in Psychology* 3 (2012), p. 232.

[50] T. Häikiö, R. Bertram, and J. Hyönä. «The development of whole-word representations in compound word processing: Evidence from eye fixation patterns of elementary school children». In: *Applied Psycholinguistics* 32(3) (2011), pp. 533–551.

[51] M. De Rosa and D. Crepaldi. «Affix Frequency in Morphological Masked Priming.» Retrieved from osf.io/zeu2n. (2021).

[52] C.E. Shannon. «A Mathematical Theory of Communication». In: *Bell System Technical Journal* 27 (1948).

[53] W. D. Marslen-Wilson, M. Bozic, and B. Randall. «Early decomposition in visual word recognition: Dissociating morphology, form, and meaning.» In: *Language and Cognitive Processes* 23(3) (2008), pp. 394–421.

[54] S. Andrews and S. Lo. «Is morphological priming stronger for transparent than opaque words? It depends on individual differences in spelling and vocabulary.» In: *Journal of Memory and Language* 68(3) (2013), pp. 279–296.

[55] Amenta S., Crepaldi D., and Marelli M. «Consistency measures individuate dissociating semantic modulations in priming paradigms: A new look on semantics in the processing of (complex) words.» In: *Quarterly Journal of Experimental Psychology.* 73(10) (2020), pp. 1546–1563.

[56] K. Diependaele, D. Sandra, and J. Grainger. «Masked cross-modal morphological priming: Unravelling morpho-orthographic and morpho-semantic influences in early word recognition.» In: *Language and Cognitive Processes* 20(1-2) (2005), pp. 75–114.

[57] J. Grainger and E. Beyersmann. «Edge-aligned embedded word activation initiates morpho-orthographic segmentation.» In: *Psychology of Learning and Motivation* 67 (2017), pp. 285–317.

[58] S. Andrews and C. J. Davis. «Interactive activation accounts of morphological decomposition: Finding the trap in mousetrap?» In: *Brain and Language* 68 (1999), pp. 355–361.

[59] J. Morris, J. H. Porter, J. Grainger, and P. J. Holcomb. «Effects of lexical status and morphological complexity in masked priming: An ERP study.» In: *Language and Cognitive Processes* 26(4-6) (2011), pp. 558–599.

[60] S. F. McCormick, K. Rastle, and M. H. Davis. «Is there a 'fete' in 'fetish'? Effects of orthographic opacity on morpho-orthographic segmentation in visual word recognition.» In: *Journal of Memory and Language* 58 (2008), pp. 307–326.

[61] D. Crepaldi, K. Rastle, M. Coltheart, and L. Nickels. «'Fell' primes 'fall', but does 'bell' prime 'ball'? Masked priming with irregularly-inflected primes.» In: *Journal of Memory and Language* 63 (2010), pp. 83–99.

[62] P. Quémart, S. Casalis, and P. Colé. «'The role of form and meaning in the processing of written morphology: A priming study in French developing readers.» In: *Journal of experimental child psychology* 109(4) (2011), pp. 478–496.

[63] E. Beyersmann, A. Castles, and M. Coltheart. «Morphological processing during visual word recognition in developing readers: Evidence from masked priming.» In: *Quarterly Journal of Experimental Psychology* 65(7) (2012), pp. 1306–1326.

Acknowledgements

I would like to thank my supervisor, Davide Crepaldi, for his constant help, his enthusiasm and his ability to keep the whole lab alive and together despite the critical moment, and for always showing a great humanity. He has been very patient in introducing me to the world of Neuroscience and he has always helped me to think fearlessly about my ideas, and to search for simplicity. I want to thank Romain Brasselet, from whom the project was born, for his support, especially in the first months of the pandemic, where our weekly calls were a great stimulus for me, and for all the occasions in which I had the opportunity to appreciate his sincere curiosity and genuine way of living science. I would like to thank Alessandro Pelizzola and Alfredo Braunstein, for their patience and willingness to answer my every question, and for the passion they put into teaching, also this year.

Grazie a chi mi è stato vicino in questi anni, e a chi mi ha fatto ridere, che non è poco.

E soprattutto grazie ad Ale, a mamma e papà, a Brici, a nonna e a P., per tutto. O per quello che magari poi vi dico dal vivo.