

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.**

**Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.**

**Bell & Howell Information and Learning  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
800-521-0600**

**UMI<sup>®</sup>**



**UNIVERSITY OF CALIFORNIA**

**Los Angeles**

**Distributional Cues in Morpheme Discovery:  
A Computational Model and Empirical Evidence**

**A dissertation submitted in partial satisfaction of the  
requirements for the degree Doctor of Philosophy  
in Linguistics**

**by**

**Marco Baroni**

**2000**

**UMI Number: 9973186**

**UMI<sup>®</sup>**

---

**UMI Microform 9973186**

**Copyright 2000 by Bell & Howell Information and Learning Company.**

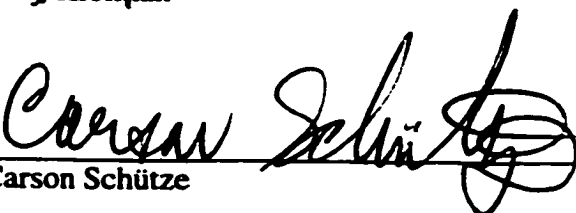
**All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.**

---

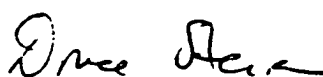
**Bell & Howell Information and Learning Company  
300 North Zeeb Road  
P.O. Box 1346  
Ann Arbor, MI 48106-1346**

The dissertation of Marco Baroni is approved.

  
Jody Kreiman

  
Carson Schütze

  
Ed Stabler

  
Donca Steriade

  
Bruce Hayes, Committee Chair

University of California, Los Angeles

2000

*To my parents, and Nicolina, Fina and Pasquina*

## TABLE OF CONTENTS

Introduction	1
Chapter 1: The Problem of Morpheme Discovery	9
1.1 The task of morpheme discovery	9
1.2 Morpheme discovery strategies based on linguistic knowledge	16
1.2.1 Phonological cues	16
1.2.2 Syntactic cues	17
1.2.3 Semantic cues	18
1.3 The role of innate knowledge in morpheme discovery	24
1.4 Distributional cues in morpheme discovery	25
1.4.1 Distribution-based morpheme discovery: general strategies	26
1.4.2 The role of distributional learning in morpheme discovery	37
1.4.3 Distribution-driven learning in morpheme discovery: the empirical evidence	39
1.4.3.1 Distributional learning and the status of morphologically complex, semantically opaque words	42
1.5 Summary	45
Chapter 2: Other models of morpheme discovery	47
2.1 Introduction	47
2.2 Harris 1955	48

2.3 Brent 1993	52
2.4 Mikheev 1997	59
2.5 Goldsmith <i>submitted</i>	63
2.6 The utterance segmentation model of Brent and Cartwright 1996	66
 Chapter 3: DDPL: An automated Distribution-Driven Prefix Learner	 75
3.1 Introduction	75
3.2 Modeling prefix discovery as an independent task	76
3.2.1 Is it legitimate to model prefix discovery as an independent task?	76
3.2.1.1 Utterance segmentation vs. morphological segmentation	77
3.2.1.2 Looking for different morphemes in different steps	80
3.2.1.3 Maximally binary parses	82
3.2.2 Why prefix discovery?	83
3.3 The DDPL model	85
3.3.1 Data compression and morphological analysis: the shortest lexicon criterion	86
3.3.2 Data compression and morphological analysis: the shortest lexicon + encoding criterion	91
3.3.3 The shortest lexicon + encoding criterion as an interpretation of the MDL principle	103
3.3.4 Data compression and morphological analysis: illustrating the relationship between the lexicon + encoding approach to data compression and distribution-based morpheme discovery heuristics	105
3.3.4.1 The high frequency heuristic	105



3.3.4.2 Co-occurrence with other potential morphemes	114
3.3.4.3 Word frequency and morphological complexity	120
3.3.4.4 Type vs. token frequency	126
3.3.4.5 An example of constraint interaction in the lexicon + encoding model	132
3.3.5 Prefixes, stems and prefix-stem asymmetries in the lexicon + encoding model	139
3.3.6 A note on boundaries	143
3.3.7 Why DDPL is not a realistic data compression scheme	146
3.3.8 Computing the length of lexicon + encoding pairs: the DDPL lexicon selection formula	148
3.3.9 Searching for the best lexicon: the lexicon generation algorithm	156
3.3.10 The DDPL model: summary	165
3.4 DDPL as a model of human morpheme discovery	166
3.5 Summary	169
 Chapter 4: The DDPL model and the discovery of English prefixes	 170
4.1 Introduction	170
4.2 The input corpus	170
4.3 Assessing the performance of the distribution-driven model, part 1: prefixes postulated by DDPL	176
4.3.1 False positives	177
4.3.2 Misses	178
4.3.3 Prefixes found and missed by DDPL: concluding remarks	180

4.4 Assessing the performance of the distribution-driven model, part 2: morphological parses assigned by DDPL compared to complexity ratings assigned by English speakers	180
4.4.1 Collecting morphological complexity ratings	182
4.4.1.1 Methodology and data collection	182
4.4.1.2 Are morphological complexity ratings valid data?	183
4.4.2 DDPL parses and speakers' complexity ratings: results and discussion	185
4.4.3 Assessing the performance of the distribution-driven model: concluding remarks	188
4.5 Evidence from the morphological treatment of semantically opaque words: a second survey	189
4.5.1 Constructing the semantically opaque word list	190
4.5.2 Methodology and data collection/analysis	193
4.5.3 Morphological complexity of semantically opaque words: discussion	194
4.6 Testing DDPL with a phonetically transcribed input	198
4.7 Summary	199
Chapter 5: Conclusion	201
Appendix 1	204
Appendix 2	206
Appendix 3	208

<b>Appendix 4</b>	<b>221</b>
<b>References</b>	<b>230</b>

## ACKNOWLEDGMENTS

First and foremost, I would like to thank Bruce Hayes, who, besides being a great teacher, was a wonderful advisor throughout my career as a UCLA graduate student, especially during the years in which I worked at this dissertation. Bruce was always very generous with his time, and he always gave me very insightful advice, about tiny technical details as well as about big philosophical issues. He was constantly able to see each problem from his point of view, my point of view (which he often saw more clearly than I did), and the point of view of my potential friends and foes. And each meeting was also very entertaining thanks to his great sense of humor.

I also wish to thank Carson Schütze, who was equally generous with his time, and gave me very detailed advice that was always useful and often of fundamental importance. Without Carson's advice, this thesis would contain many unjustified claims that I would have eventually been ashamed about, and many crucial issues would not have even been mentioned.

Thanks to Ed Stabler, who introduced me to the Minimum Description Length principle and many other important philosophical, computational and linguistic ideas, methods and theories, and was always very encouraging. Even in periods of discomfort, a meeting with Ed always renewed my enthusiasm about this project.

Thanks to Donca Steriade, who gave me many good ideas and noticed many important aspects of this work I would have missed completely.

Thanks to Jody Kreiman for accepting to be in my committee and for advice.

I would like to thank my linguistics teachers and advisors at the University of Padua, in particular Laura Vanelli, Alberto Mioni and Michele Loporcaro. Without their classes, I would have never thought that linguistics could be so much fun. I am very

grateful for how they were always very generous with their time, willing to discuss linguistic problems and theories with me, and very encouraging.

Thanks to Alberto Mioni also for helping me coming to UCLA as an undergraduate exchange student.

Special thanks to Laura Vanelli, who kept being generous with her time, giving me insightful advice, and being willing to engage in stimulating discussions even after I moved to a different continent.

Thanks to Pat Keating and Lynne Bernstein for advice and for all the occasions I had to learn new things working as their research assistant.

Thanks to Adam Albright for important advice, generous help and useful information about a wide spectrum of topics, ranging from computational morphology and psycholinguistics to entomology and Indonesian music.

Thanks for advice and help to my colleagues at UCLA, the members of the UCLA phonetics laboratory and the participants in the 9th International Morphology Meeting in Vienna, and in particular Harald Baayen.

Thanks to the Department of Linguistics staff, in particular Anna Meyer and John Bulger, for helping me with the bureaucratic side of things.

Thanks to UCLA, the UCLA Graduate Division and the UCLA Department of Linguistics for financial support.

I wish to thank my parents for their financial and moral support, for raising me in a house full of books, for being so open-minded and liberal but also so warm and affectionate, and for all the interesting conversations we had together.

Thanks to Zia Fina, Nonna and Pasquina for their affection and support.

Thanks to Fede, Paolo, Roberto, Stefano and all my friends from Bolzano, who kept in touch and make me still feel at home whenever I go back.

Thanks to all my LA friends, who made me love this weird but amazing city: Stefano, Adam, Paul, Sahyang, Taehong and too many others to mention...

Last but most definitely not least, thanks to Moto for love, stimulating discussions, great food, her willingness to explore the postmodern urban landscape with me and for putting up with my contamination obsessions and all the hours spent in record stores.

## **VITA**

<b>November 1, 1970</b>	<b>Born, Bolzano, Italy</b>
<b>1993</b>	<b>Summer School Fellowship University of Bucharest Bucharest, Rumania</b>
<b>1993-1994</b>	<b>Education Abroad Program Fellowship University of California, Los Angeles</b>
<b>1995</b>	<b>Laurea in Linguistica, summa cum laude University of Padua Padua, Italy</b>
<b>1995</b>	<b>Summer School Fellowship Center for Semiotic and Cognitive Studies San Marino</b>
<b>1997</b>	<b>M.A., Linguistics University of California, Los Angeles</b>

1998	<b>Research Assistant</b> <b>Spoken Language Processes Laboratory</b> <b>House Ear Institute</b> <b>Los Angeles</b>
1996-1998	<b>Teaching Assistant</b> <b>Department of Linguistics</b> <b>University of California, Los Angeles</b>
1999	<b>Research Assistant</b> <b>Department of Linguistics</b> <b>University of California, Los Angeles</b>
1995-2000	<b>Chancellor Fellowship</b> <b>University of California, Los Angeles</b>

## **PUBLICATIONS AND PRESENTATIONS**

**Baroni, M. (1993, March). Teorie della sottospecificazione e restrizioni sulle code consonantiche in italiano. Paper presented at the Italian Generative Grammar Meeting, Trento, Italy.**

**Baroni, M. (1993). Teorie della sottospecificazione e restrizioni sulle code consonantiche in italiano. Rivista di Grammatica Generativa, 18: 3-59.**



- Baroni, M. (1994). Moraic structure and vowel length in Galeatese. *Romance Linguistics and Literature Review*, 7: 24-52.
- Baroni, M. (1996). The natural classes of Lughese vowels and why they are natural. *UCLA Working Papers in Phonology*, 1: 1-17.
- Baroni, M. (1996, December). An acoustic study of Italian unstressed mid-vowels, poster presented at the Meeting of the Acoustical Society of America, Honolulu, Hawaii.
- Baroni, M. and Vanelli, L. (1997, September). Il contrasto di lunghezza vocalica in friulano. Paper presented at the 31st Italian Linguistic Society Conference, Padua, Italy.
- Baroni, M. (1998). The phonetic nature of the Northern Italian allophones [s] and [z] in words with variable realization: electroglottographic and acoustic evidence. *UCLA Working Papers in Phonetics*, 96: 166-174.
- Baroni, M. and Vanelli, L. (1999). Il contrasto di lunghezza vocalica in friulano. In P. Beninca', A. Mioni and L. Vanelli (eds.), *Fonologia e morfologia dell'Italiano e dei dialetti d'Italia*. Roma: Bulzoni: 291-317.
- Baroni, M. (2000, January). Using distributional information to discover morphemes: A distribution-driven prefix learner. Paper presented at the 74th Meeting of the Linguistic Society of America, Chicago, Illinois.

Baroni, M. (2000, February). A distribution-driven prefix learner, oral presentation at the 9th International Morphology Meeting, Vienna, Austria.

Baroni, M. (in press). Iambic senarii. *Quaderni Patavini di Linguistica*.

Baroni, M. (in press). The representation of prefixed forms in the Italian lexicon: Evidence from the distribution of intervocalic [s] and [z]. *Yearbook of Morphology 2000*.

Baroni, M. and Vanelli, L. (in press). The relationship between vowel length and consonantal voicing in Friulian. In L. Repetti (ed.), *Italian dialects and phonological theory*. Amsterdam: John Benjamins.

## **ABSTRACT OF THE DISSERTATION**

**Distributional Cues in Morpheme Discovery:  
A Computational Model and Empirical Evidence**

**by**

**Marco Baroni**

**Doctor of Philosophy in Linguistics**

**University of California, Los Angeles, 2000**

**Professor Bruce Hayes**

In an early stage of morphological acquisition, children must discover which strings correspond to affixes of their language, and which of the words containing those strings are actually affixed. For example, a child acquiring English must be able to discover that the word-initial string *re-* is a prefix, but also that the word *remake* is prefixed, whereas the word *retail*, probably, is not, even though it begins with *re-*.

In this study, I present a computational model of how the task of morpheme (in particular, prefix) discovery could be performed on the basis of distributional cues (cues based on the distribution, frequency and length of words and their substrings in the input).

The results of a simulation with a corpus of English words show that distributional evidence could in principle be very helpful to learners having to perform the task of morpheme discovery.

Moreover, I show that the morphological parses assigned by the distribution-driven model to a set of potentially prefixed but semantically opaque words are correlated with morphological complexity ratings assigned to the same words by native English speakers. I argue that this convergence between the model and the speakers, in a domain in which speakers cannot rely on semantic cues, constitutes evidence that humans do rely on distributional cues similar to the ones exploited by my model, when assigning morphological structure to words.

## Introduction

A child acquiring the morphology of her language must first of all discover which strings constitute the morphemes (in particular, affixes) of the language. For example, learners of English must realize that, say, the strings *re-* and *-s* are (or rather, can be) morphemes before they can learn the grammatical (morphological, semantic and syntactic) properties associated with these strings.

Moreover, the learners must discover which words containing strings identical to morphemes are actually morphologically complex. For example, in order to formulate the correct generalizations about the grammatical properties of the affixes *re-* and *-s*, an English learner must realize that, while words like *redo* and *cats* contain *re-* and *-s*, respectively, the word *retail* should not be decomposed into *re-* and *tail*, and the word *lens* should not be decomposed into *len* and *-s*.<sup>1</sup>

Even after the speaker figures out the morphology of her language and discovers the grammatical properties (morphological and syntactic function, meaning...) associated with each morpheme, she will have to occasionally decide whether a new word containing a string identical to a morpheme she is familiar with does indeed contain the morpheme. In this case, the learner can simply try to match the grammatical properties of the morpheme with the morphological, syntactic and semantic characteristics of the new word. However, learners must have at least a preliminary idea about which of the words containing a string

---

<sup>1</sup>In this study, most points are illustrated through examples presented in orthographic transcription. Unless explicitly mentioned, the same points could have been also illustrated by the same examples (or similar ones) presented in phonetic transcription.

identical to a certain morpheme are actually morphologically complex *before* they discover the grammatical properties of the morpheme, or else they would not be able to extract correct generalizations about such properties. In this study, I will be referring regularly to this preliminary, probably tentative kind of morphological parsing, which the learner must perform in order to discover which words are going to teach her something useful about a morpheme.

The task of finding the morphemes of a language and the task of discovering which words of the language are morphologically complex are related. Indeed, in the learning model I present here the task of discovering morphemes is almost entirely reduced to the task of deciding which words are complex (a string is a morpheme of the language if and only if it is parsed as a morpheme in at least one word containing it). I will refer to the sum of both these preliminary morphological learning tasks (finding the morphemes and discovering which words are complex) with the cover term *morpheme discovery*.

The primary goal of this study is to contribute to a better understanding of how language learners perform morpheme discovery. In particular, the study provides evidence in favor of the hypothesis that learners rely on distributional cues in order to succeed in this task. While the idea that distributional information plays an important role in language learning has not been popular in the generative literature on acquisition, many recent studies have provided new support for it (see Redington and Chater 1998 for a review of both the classic generative objections to distributional approaches and recent distribution-driven learning models). Thus, another goal of the present study is to provide further support for the general claim that language learners make crucial use of distributional cues.

Finally, the current study proposes a partial explanation for an interesting datum emerging from experimental studies of morphological processing and representation (discussed in 1.4.3.1 below), i.e., that speakers can represent words as morphologically

complex even if they lack semantic compositionality (i.e., the meaning of the whole word is not the product of the meanings of the component morphemes). I argue that this phenomenon (complex representation/treatment of semantically opaque words) is at least in part a by-product of distribution-driven morpheme discovery, and I present some empirical support for this hypothesis.

In order to assess the potential role of distributional cues in morpheme discovery, I designed an automated learner which performs a simplified version of this task on the sole basis of the distributional evidence it can extract from a corpus of untagged words. The most obvious simplification in the task performed by this computational model, as opposed to actual morpheme discovery as performed by children, is that the automated learner only looks for prefixes and stems. The automated learner does not consider the hypothesis that the language of the input corpus contains other kinds of morphemes, such as suffixes, infixes, prosodic templates, autosegmental morphemes etc. (I discuss in 3.2 below the reasons why I chose to concentrate on prefixation, and why I believe that it is legitimate to model prefix discovery as a separate subtask within morpheme discovery).

Most of the heuristics followed by the automated learner in its search for prefixes and stems are based on a simple observation about the distributional nature of morphemes, i.e. that morphemes are independent linguistic units and, as such, they occur in a number of different words where they combine with other morphemes. Heuristics following from this observation are implemented using a version of the Minimum Description Length (MDL) criterion (Rissanen 1978, Brent and Cartwright 1996 and references quoted there). Thus, while this is not one of the central points of this study, the research presented here also constitutes another example of the usefulness of the MDL criterion in modeling distribution-driven learning.

Given an input corpus of English words, the automated learner, equipped with a small number of simple distributional heuristics, is able to discover a large set of actual English prefixes, finding very few “false positives” (strings which are not English prefixes but are treated by the learner as such). Moreover, the morphological parses (prefix + stem vs. monomorphemic parses) assigned by the learner to the words in the input corpus are correlated with intuitions of native English speakers about the morphological structure of the same words.

Thus, the computational simulation presented here demonstrates that a limited number of simple distributional heuristics can help a morpheme discoverer a great deal. In particular, the success of the simulation constitutes evidence against the claim that children cannot *in principle* learn something about morphology from distributional evidence, since distributional evidence does not provide enough useful cues.

While I am not aware of similar claims in relation to morpheme discovery, in the past distributional approaches have been criticized on the basis of *a priori* arguments of this nature: Given that distributional cues in domain  $x$  can in principle be shown to be insufficient and/or misleading, there is no need to collect empirical evidence pertaining to the relevance of distributional learning in domain  $x$  (see Redington and Chater 1998 for the discussion of similar arguments, and in particular of the influential arguments presented by Pinker 1984 against distribution-driven learning in syntax<sup>2</sup>). Given the successful results

---

<sup>2</sup>In short, Pinker’s *a priori* arguments against distributional learning in syntax reviewed by Redington and Chater are: 1) some properties of syntax cannot be induced from positive evidence; 2) If a learner had to consider all the possible relationships between elements in the surface structure of sentences, there would be a combinatorial explosion of analyses to be evaluated; 3) the type of information that can easily be extracted from the input (serial positions, adjacency, etc.) is not very informative; 4) as linguistic properties are not



of the simulation presented here, I believe that future discussion of the relevance of distributional cues in morpheme discovery should be conducted on empirical grounds: Since children could, in principle, make successful use of distributional evidence, the question is whether they actually resort to distributional cues during morphological acquisition -- a question to be answered by developmental psycholinguistic studies, rather than on the basis of general *a priori* arguments.

I would argue, indeed, that the most plausible *a priori* hypothesis to be submitted to empirical testing is that children *do* resort to distributional cues in morpheme discovery. This claim is based on the following considerations: First, distributional information can be straightforwardly extracted from the input data before the child performs any kind of linguistic analysis. Thus, it is reasonable to hypothesize that the child formulates preliminary hypotheses about the morphological units of the language on the basis of distributional evidence. These preliminary guesses will make the goal of gathering relevant linguistic generalizations easier, and later the more sophisticated linguistic information acquired in this way can be used to refine those preliminary guesses. Second, it is likely that distributional information must be collected by learners anyway, in order to find reliable linguistic generalizations. Third, psycholinguistic research suggests that adult speakers are sensitive to the distributional properties of morphemes. Thus, we have direct evidence that humans do, at some stage in morphological development, are sensitive to distributional information about morphemes.

---

independent from each other, it would be anti-economical to try to extract all properties from the data, rather than deriving some properties from other properties; 5) a distribution-driven learner could come to wrong conclusions due to spurious correlations in small samples of data.

Going beyond these general considerations, the comparison of the output parses assigned by the automated prefix learner to English words with morphological intuitions of native speakers also provides a form of more direct empirical evidence in favor of the hypothesis that learners resort to distributional cues in morpheme discovery. Above, I observed that the fact that the automated learner assigns plausible parses (parses matching adult speakers' intuitions) shows that human learners could in principle successfully rely on distributional cues. However, this does not *per se* constitute evidence that humans *do* rely on such cues, since humans, unlike the automated learner, could have used different types of evidence in order to discover the same structures found by the automated learner. As I will argue in 3.2.2 below, it is unlikely that English learners can extract much useful information about prefixation from phonological and syntactic cues. However, the learners could still have relied on semantic cues.

For example, the learner, exploiting distributional cues only, came to the conclusion that *renamed* is a prefixed word, composed of the prefix *re-* and the stem *named*. Although all the native speakers surveyed shared the intuition that *renamed* is indeed a prefixed form composed of *re-* and *named*, this convergence between the automated learner and humans does not prove that humans exploited distributional cues in morpheme discovery, since humans could have decided that *renamed* is prefixed simply on the basis of its meaning.

Consider, however, the case of semantically opaque but potentially morphologically complex words such as *recitation* or *remain*. Words of this kind are potentially prefixed, at least in the sense that they begin with a string identical to a prefix (*re-*, in this case). However, the meaning of *recitation* is not synchronically related to the meaning of the prefix *re-* nor to the meaning of the stem *citation* (or *cite*). In the case of *remain*, not only is the meaning of the word not related to the meaning of the components, but it is not even clear that the potential stem, a bound verbal form *-main*, could be associated with any

semantic content. Several experimental studies (see 1.4.3.1 below) have shown that speakers treat some semantically opaque forms as morphologically complex.

Now: if it turned out that there is a convergence between the parses assigned by the distribution-driven learner and the speakers' intuitions about semantically opaque forms, then this would constitute a stronger form of evidence in favor of the hypothesis that speakers used distributional cues to assign morphological structure to words. For example, if it turned out that both the automated learner and the speakers treated *recitation* as morphologically complex (*re+citation*), but *remain* as monomorphemic, then it would be reasonable to conclude that speakers are sensitive to distributional cues similar to the ones implemented in the automated learner, since they could not have assigned a structure to *recitation* on the basis of its meaning (and also, it is unlikely that they could have used syntactic or phonological cues to distinguish *recitation* from *remain*).

Indeed, I will show that, even when only semantically opaque words are considered, there is a significant correlation between the parses assigned by the learner and speakers' intuitions. Thus, this study provides strong support for the claim that humans use distributional cues in morpheme discovery.

Notice that this type of evidence in favor of distribution-driven learning is not available in other domains. For example, even if it has been shown that distributional cues can be very effective for segmenting utterances into words (Brent and Cartwright 1996), there is no equivalent to semantically opaque morphemes in this domain.

At the same time, the data on semantically opaque words presented here are also of interest to the theory of morphological representations and processing, in that they provide a partial explanation for the phenomenon of complex representation of semantically opaque forms: Adult speakers treat some semantically opaque forms as morphologically complex

because they used distributional cues in morpheme discovery which lead them to treat such forms as complex.

The remainder of this study is organized as follows: In chapter 1, I discuss the problem of morpheme discovery in general terms, focusing on the nature and possible role of distributional cues in this domain. In chapter 2, I review models of morpheme discovery proposed by other authors. In chapter 3, I present and discuss the computational model I am proposing. In chapter 4, I present the results of a simulation in which this model was tested, and I compare the morphological parses assigned by the model to morphological complexity ratings assigned by humans. Finally, in the conclusion I briefly discuss outstanding problems and future directions that this project could take.

# **Chapter 1**

## **The Problem of Morpheme Discovery**

### **1.1 The task of morpheme discovery**

During the process of language acquisition, learners must discover which strings constitute the morphemes of their language<sup>3</sup> and which words of the language can be decomposed into morphemes. These are prerequisites to morphological acquisition.

Ultimately, a learner acquiring a language must discover the syntactic and semantic properties associated with each morpheme (in particular, affix) of the language, in order to be able to produce and understand new words, and possibly for other reasons (such as reducing the amount of information stored in the lexicon by avoiding storing certain morphologically complex words). For example, a learner acquiring English must discover that, say, *re-* is a prefix which attaches to verbs to create other verbs with an iterative meaning.

However, in order to learn the morphological properties of an affix (or class of affixes), learners must first of all notice the existence of that affix. Moreover, in order to discover the linguistic properties associated with the affix the learner must consider the

---

<sup>3</sup>Even theories in which affixes do not constitute lexical entries (such as the theory of “a-morphous morphology” of Anderson 1992) must assume a stage in which learners discover phonological strings corresponding to affixes. Even if in such theories these strings are not going to form lexical entries, speakers must become aware of their existence in order to encode them in the relevant morphological rules or constraints.

semantic, syntactic and morphological characteristics of a set of words containing that affix. For example, in order to discover the properties of the prefix *re-*, English learners must first of all, of course, notice that the string *re-* is a prefix. Moreover, the learners must collect and analyze a number of words containing the prefix *re-* (*redo*, *rename*, *remake*...) in order to extract the correct generalizations about this prefix.<sup>4</sup>

However, not all the words containing a string identical to an affix actually contain that affix. In order to discover the correct generalizations about the properties of the affix, the learners must have a preliminary idea of which of the words containing a string identical to the affix are actually affixed. If an English learner tried to decide what is the meaning and function of *re-* on the basis of, say, *redo*, *retail* and *really*, the learner would probably come up with the wrong generalizations about the prefix or, more likely, she would not notice any generalization at all and she would conclude that *re-* is not a prefix.

---

<sup>4</sup>It seems plausible that learners will need to consider a certain number of forms containing a certain affix before they draw conclusions about the properties of that affix. A learner should not rely on a single form for at least two reasons: First, even if the form seems to be related to another word that the learner knows, the relation could be a matter of chance (or etymology) -- it would be risky for the learner to conclude, on the sole basis of the words *redo* and *do*, that *re-* is a prefix attaching to verbs and carrying an iterative meaning. Notice that it is not sufficient for the learner to make sure that the single, potentially affixed form she chose to consider is semantically related to its potential base. For example, if a learner were going to try to extract the (synchronic) semantic properties of the prefix *re-* from the semantically related pair *represent/present*, she would probably fail. Moreover, a learner should analyze more than one form per affix to determine whether the affix can be attached to words belonging to different syntactic categories, whether the meaning of the affix changes depending on certain properties of the stem, etc.

Of course, if the string corresponding to an affix mostly occurs in words which do indeed contain the affix, the learner is probably going to extract the correct generalizations even if there are a few pseudo-affixed words (i.e., words containing the string corresponding to the affix without actually being morphologically affixed). However, this is not always the case. For example, Schreuder and Baayen 1994 have shown that, for several common English and Dutch prefixes, the number of pseudo-prefixed words is higher than the number of truly prefixed words.<sup>5</sup>

Thus, it would not be safe, for a learner, to assume *a priori* that any word containing a string identical to an affix does indeed contain the affix from a morphological point of view. Consequently, the learner, besides hypothesizing that a string corresponds to an affix of her language, must also decide which of the words containing the affix are truly morphologically complex, and which are pseudo-affixed, i.e. the learner must assign morphological parses to the words she hears.

Notice that we are referring to the task of discovering that a certain string is an affix (or more generally a morpheme) and the task of assigning parses to words as separate aspects of morpheme discovery, however the two tasks are closely related. In particular, since a learner probably does not hear affixes in isolation, the task of discovering the affixes will typically involve assigning morphological parses to words. A string is an affix of the language if at least one of the words containing the string in the language is parsed as

---

<sup>5</sup>Different factors can affect the ratio of truly affixed to pseudo-affixed forms per affix, ranging from the phonological length of the affix (a string identical to a longer affix is less likely to occur by chance than a string identical to a short affix) to its productivity (inflectional and highly productive derivational affixes are likely to have fewer pseudo-affixed forms than less productive derivational affixes).

morphologically complex, and the string constitutes one of the morphological components in the parse.

This study explores the possible role of distributional cues in the morpheme discovery stage of language acquisition, i.e. in the stage in which a learner assigns tentative morphological parses to words, before she determines which strings constitute the set of affixes of her language and before she figures out the morphological properties of these affixes. Even in adult life, when speakers hear a new word containing a string identical to an affix, they must decide whether the word is morphologically complex or not. However, adult speakers can rely on their knowledge of the morphological, syntactic and semantic properties of morphemes, when assigning morphological structure to new words. This is an essentially different task from the one faced by learners during morpheme discovery, when they have to guess the morphological structure of words in order to discover the linguistic properties of the potential morphemes they contain. Of course, as morpheme discovery proceeds, learners will progressively acquire bits of morphological knowledge, and it is likely that they will immediately put this knowledge to use in order to parse new words. However, in the beginning stages of morpheme discovery, this knowledge will be rather fragmentary and not entirely reliable. When, below, I discuss the role of linguistic cues in morpheme discovery, I am referring to this kind of early, fragmentary knowledge.

Morpheme discovery is of course successful, in the sense that learners are eventually able to discover the set of affixes/morphemes of their language and to extract the correct generalizations about the meaning and function of morphemes. However, it is extremely unlikely that the tentative parses assigned by a learner to potentially complex words during morpheme discovery are entirely appropriate: Given that the learner does not *a priori* know the semantic, syntactic and morphological properties associated with an affix (she is assigning morphological parses to words *because* she needs to discover such



properties!), it would be a miracle if she managed to parse all and only perfectly transparent, semantically compositional forms as complex. It is more likely that the parses assigned by the learner are, in general, accurate enough to allow the learner to discover the relevant generalizations, but, still, some forms that should be treated as complex are not, and some opaque forms are treated as complex.

Thus, the following question arises. Once the learner successfully terminates the process of acquiring the morphology of her language, does she update the earlier parses on the basis of the newly acquired knowledge? For example, suppose that during the morpheme discovery stage a learner acquiring English decided that, say, *recite* must be represented as a prefixed word (*re+cite*). Once the learner discovers the morphological properties of the prefix *re-* (something like: ‘*re-* is a prefix which attaches to verbs to form verbs with an iterative meaning’), would she change her representation of the word *recite*, since *to recite* is not the same as *to cite again*?

The psycholinguistic studies I mentioned in the introduction, suggesting that adult speakers treat some semantically opaque words as morphologically complex, as well as the results that will be reported here, support the view that speakers *do not* update their parses of words that were “wrongly” represented as complex during morpheme discovery (under the assumption, of course, that learners started representing the relevant set of opaque words as complex during morpheme discovery). Still, I am not necessarily claiming that words of this sort have exactly the same status as completely transparent complex words. In particular, in theories in which the notion of morphological complexity is not categorical (simple vs. complex) but gradient, such as the one I sketched in Baroni *in press* or the one defended by Gonnerman and Andersen 2000, we would expect opaque but complex words to have an intermediate status: A word such as *recite* would have a higher complexity index than, say, *really*, but would still be less complex than a perfectly transparent form such as

*redo*. Under a gradient view of this sort, we predict that it is possible to distinguish forms of intermediate morphological complexity from both completely simple and perfectly complex forms.<sup>6</sup> This second type of comparison is not pursued in this study.

Before we turn to consider how different cues can help learners in morpheme discovery, let me remark that, even if it has not been frequently studied by morphologists (see the review of previous studies in chapter 2), morpheme discovery is by no means a

---

<sup>6</sup>Another theory that could account for a three-way distinction of this sort, without requiring the assumption that morphological complexity is a gradient property, was suggested to me by Carson Schütze: In this alternative model, morphologically complex but opaque words are represented as lexical units associated with the corresponding regular affix entry, and with a special stem entry specifying the idiosyncratic properties of the form (alternatively, the special stem entry contains only the phonological representation of the stem, and the idiosyncratic properties associated with the form are associated with the lexical unit connecting affix and stem). For example, the semantically opaque word *department* would be represented by a node associating the unit corresponding to the regular suffix *-ment* with a special entry corresponding to the stem *depart* as used in this word, which constitutes a different entry from that of the verb *depart* (there is no semantic relation between departments and departing). This theory predicts a three-way distinction between transparent complex words (represented by a link between the regular, independent entries for the corresponding stems and affixes), opaque complex words (represented as just discussed) and monomorphemic words (not associated with affixes). On the other hand, under the gradient approach, there is no reason to expect the distinction between words of different degrees of complexity to be limited to three levels. Thus, the two theories make different empirical predictions. However, for our current purposes it is not necessary to choose between them, as the whole point of my discussion in the text is that the claim that some semantically opaque forms are morphologically complex does not automatically commit us to a theory in which the representation of semantically opaque and transparent forms is indistinguishable.

trivial task. Not only does the learner have to consider many possible segmentations of each potentially complex word she hears, but she does not *a priori* know which meanings and/or syntactic functions are expressed by morphemes in her language, and consequently she cannot *a priori* know whether a word has to be decomposed into morphemes or not. Furthermore, the learner does not know which types of morphemes (prefixes, suffixes, circumfixes, infixes, autosegments, templates...) are present in the language. Thus, even if the learner had a reason to expect a certain word to be morphologically complex (for example, thanks to her innate knowledge), she still would not *a priori* know whether the word should be divided into a prefix and a stem, or into a stem and a suffix, or into consonantal and vocalic templates, or into other morpheme combinations.

It is probable that learners follow a number of different morpheme discovery strategies, looking for phonological, syntactic and semantic cues and relying on innate knowledge. Moreover, the frequency and distribution of words and their substrings constitute potentially useful sources of evidence that learners can exploit. While each of these approaches can help the learner in the morpheme discovery task, none of them is likely to be sufficient by itself. In the current project I am modeling morpheme discovery as a purely distribution-driven task because I am interested in trying to determine how much and what kind of information a learner could in principle extract from distributional evidence alone. I am *not* trying to argue that this is the only kind of evidence used by human learners.

In the remainder of this chapter, I will consider the potential role of different forms of evidence in morpheme discovery. Of course, the focus will be on the role of distributional learning.

## 1.2 Morpheme discovery strategies based on linguistic knowledge

Children looking for morphemes and morphologically complex forms can rely on their knowledge of other linguistic domains, such as phonology, syntax and, especially, semantics. Of course, these strategies assume that children already acquired the relevant knowledge in the relevant domains.

### 1.2.1 Phonological cues

Sometimes, special phonotactic or prosodic patterns mark morphemes and morpheme edges. For example, in Northern Italian only the voiced alveolar fricative allophone [z] can occur intervocalically, except in stem-initial position, where its voiceless counterpart [s] occurs (Nespor and Vogel 1986, Baroni *in press*): Cf. *ri[z]altare* ‘to stand out’ vs. *ri[s]altare* ‘to jump again’ (from *ri-* ‘re-’ plus *saltare* ‘to jump’). Italian learners could notice the (relatively) unusual occurrences of intervocalic [s], and hypothesize that the forms with intervocalic [s] have a special morphological status.

While phonological cues of this sort are potentially very useful, they are also obviously limited to the special cases in which morphemes or morpheme boundaries are signaled by special phonotactic or phonological configurations. For example, as far as I can tell, Italian suffixes and stem-suffix boundaries display no single phonotactic or prosodic characteristic distinguishing them from monomorphemic strings.

Thus, while phonological cues can constitute a precious “bonus” in morpheme discovery, a sensible learner cannot *a priori* expect that all the affixes/morphemes of her

language will display some special phonological mark. Phonological cues are likely to play only an auxiliary role.

Moreover, it is not clear that, in cases in which morphemes serve as the domain of particular phonological patterns, it is phonology that is providing cues to morphological structures and not *vice versa*: language learners could be using their knowledge of morphology to discover the relevant (morpho-)phonological generalizations. For example, Italian learners who figured out which words are morphologically complex could notice that, in all the forms displaying intervocalic [s], the latter occurs in stem-initial position, and thus conclude that the phonological generalization that [z] is the intervocalic alveolar fricative allophone is exceptionless, but morphology-sensitive.

### 1.2.2 Syntactic cues

Learners can look for systematic relationships between phonological substrings and the distribution of words in sentences or the syntactic category of the words containing the substrings. For example, children acquiring English could notice the distribution of -s in pairs such as *the cat sleeps* vs. *the cats sleep* and infer from it that the final -s's of nouns and verbs are morphemes. In cases like this, syntactic patterns can be very helpful to a morpheme learner.<sup>7</sup>

---

<sup>7</sup>Aronoff 1994 (section 2.5) discusses the case of semantically empty morphemes with a purely morphosyntactic function, such as Latin theme vowels, whose only function is to mark the membership of a verb in a certain conjugation class. It is clear that syntactic/morphosyntactic cues must play an especially important role in the identification of this type of morpheme.

However, syntactic cues are mostly relevant to the discovery of inflectional morphemes and, possibly, of specific classes of derivational morphemes (for example, category-changing or category-marking affixes). A large portion of derivational morphology is independent from syntax, and consequently a syntax-driven morpheme discovery strategy is not likely to be very useful in this domain.

### **1.2.3 Semantic cues**

Probably, semantic cues are the first kind of evidence that comes to mind when one ponders the issue of how children can discover morphemes. Thus, I will dedicate some space to arguments suggesting that, while it is very likely that semantics plays a major role in morpheme discovery, morpheme discovery could not be and is not performed on the sole basis of semantic cues.

In order to find morphemes, learners can look for systematic relationships between phonological strings and meaning components. For example, an English learner could notice that the word-initial substring *re-* tends to occur in words with an iterative meaning (*redo*, *recharge* etc.) and conclude that *re-* is a prefix meaning something like ‘again’ (of course, knowing the meaning of *do* and *charge* will also be helpful). Moreover, before the learner figures out the exact semantic properties of *re-*, she could decide that, say, *retail* is not a prefixed form on the basis of the fact that it is not even remotely semantically related to *tail*.

This is a very plausible and effective strategy, and, obviously, learners must eventually become aware of the semantic (and/or syntactic) features associated with morphemes, or else they would not be able to create and understand newly formed

morphologically complex words. However, there are arguments suggesting that learners could not succeed in the morpheme discovery task on the sole basis of semantic information and some evidence that, indeed, they also use other kinds of cues.

First of all, a purely semantically based strategy appears to be rather inefficient and anti-economical. I am not able to provide formal support for this claim, given that I am not familiar with any theory of the acquisition of lexical semantics explicit and detailed enough to allow a full formalization of how the semantic approach to morpheme discovery would work. However, let us consider a very schematic model which should illustrate the point.

Let us assume that a learner has to acquire the morphology of a language in which all words have four segments and four semantic features. In this stage, the learner is looking for prefixes, and thus she considers only binary (prefix + stem) parses of words. Each four segment string has three possible segmentations ( $a+bcd$ ,  $ab+cd$ ,  $abc+d$ ). Assuming that each potential stem must be associated with at least one semantic feature, each segmentation can have fourteen ( $2^4 - 2$ ) semantic analyses. For example, consider the segmentation  $a+bcd$ , and the four semantic features 1, 2, 3, 4. The following analyses are possible:  $a = 1, bcd = 234$ ;  $a = 2, bcd = 134$ ;  $a = 3, bcd = 124$ ;  $a = 4, bcd = 123$ ;  $a = 12, bcd = 34$ ;  $a = 13, bcd = 24$ ;  $a = 14, bcd = 23$ ;  $a = 23, bcd = 14$ ;  $a = 24, bcd = 13$ ;  $a = 34, bcd = 12$ ;  $a = 123, bcd = 4$ ;  $a = 124, bcd = 3$ ;  $a = 134, bcd = 2$ ;  $a = 234, bcd = 1$ . The same fourteen analyses are possible for each of the other two segmentations. Thus, for each word, a learner has to store a total of forty-two (fourteen times three) possible morphological/semantic parses. While it is likely that the distribution of semantic features is more constrained than this example would suggest (at least, it is likely that some features must/cannot be associated with the same morpheme), it is also the case that in a real

language words are likely to be on average longer than four segments, and the semantic features associated with words are likely to be more than four.<sup>8</sup>

Continuing with our schematic example, each time the child learns a new word and generates the forty-two potential parses that can be associated with the word, she then must go through the set of previous words, checking if one of the two parts of any of the forty-two parses associated with each word matches one of the parts of any of the forty-two parses of the new word. Clearly, this is not a very efficient and economical way to explore the relevant hypothesis space.

Compare this with a strategy based on distributional cues: Given that a distributional strategy does not require semantic analysis, a learner relying on distributional evidence has to consider only three possible morphological parses of each four segment word, independently of the number of semantic features associated with the word. Moreover, very simple distributional heuristics can trim down the hypothesis space radically. For example, a learner could consider as potential affixes only strings which

---

<sup>8</sup>To discover the structure of complex words with free stems, learners could first identify the meaning of shorter words, and then consider whether any longer word can be reduced to a combination of the phonological material and semantic features of a shorter word plus a remainder of phonological material and semantic features which belong only to the longer word, and which is likely to correspond to an affix of the language. This strategy is more efficient than the one discussed in the text, but learners cannot *a priori* assume that their language does not contain bound stems. Moreover, the strategy suggested here is implicitly based on a semantics-independent, purely distributional constraint (something like: “longer words are more likely to be morphologically complex”).



occur in at least a certain number  $n$  of words. No analogous heuristic is available to a purely semantics-driven learner.<sup>9</sup>

Arguments based on efficiency and economy are not conclusive, since we do not know much about the actual processing and storage limitations of the human linguistic faculty, but the previous considerations suggest that it would be sensible for learners to trim down the hypothesis space using at least some simple distributional strategy before they start evaluating possible morpheme-to-meaning-component mappings.

Possibly, the following is a stronger argument against the hypothesis that a learner could successfully complete the morpheme discovery task by relying on semantic evidence alone. As we remarked earlier, substrings corresponding to morphemes tend also to occur in words that are not morphologically complex, or in words that are etymologically complex but semantically opaque (such as *presume*). If the percentage of cases in which the string corresponding to a particular morpheme occurs in semantically transparent forms is small, it becomes very hard for learners to notice the relevant semantic generalizations.

Similar cases are more common than one may think, as shown by the lexico-statistic analysis of Schreuder and Baayen 1994. Schreuder and Baayen present statistics (based on the Cobuild corpus -- Baayen, Piepenbrock and van Rijn 1993) on the proportion of pseudo-prefixed forms for each of the seven most frequently occurring English prefixes (they also present similar data for Dutch). Pseudo-prefixed words, according to Schreuder and Baayen's definition, are words like *refer* or *remit*, which begin with a string identical to a prefix (*re-*, in this case), but whose potential stems (*-fer* and

---

<sup>9</sup>It is indeed not clear what would count as a "pure" semantics-driven learner. At the very least, it seems that a semantics-based learner should collect statistics on the frequency of co-occurrence of each potential morpheme with various semantic feature sets, in order to look for reliable semantic generalizations.

*-mit*) never occur in semantically fully transparent combinations, and are never used in productive word formation (forms in which the potential stem would be an orthographically/phonologically ill-formed word and forms in which the potential stem is orthographically very short are not counted as pseudo-prefixed).

The average proportion of pseudo-prefixed word tokens across the seven prefixes considered is 81% for orthographic forms and 83% for phonologically transcribed forms. The proportion of pseudo-prefixed word *tokens* ranges from 24% (orthography) / 44% (phonemic transcription) in the case of *un-* to 98% (both orthography and phonemic transcription) in the case of *de-*.

The proportion of pseudo-prefixed word *types* per prefix is lower but still considerable, ranging from 17% (orthography) / 19% (phonemic transcription) in the case of *mis-* to 72% (orthography) / 71% (phonemic transcription) in the case of *de-*. The (relative) mismatch between token- and type-based counts indicates that pseudo-prefixed words tend to have high token frequency and truly prefixed words tend to have low token frequency. Thus, if only high token frequency words had been considered when computing type frequency, the proportion of pseudo-prefixed words per prefix would have probably been considerably higher.

Thus, under the reasonable assumption that language learners are mostly exposed to high frequency words, it follows that children acquiring English must hear/read pseudo-prefixed words much more frequently than semantically transparent, truly prefixed words. Thus, they can very easily overlook the relevant semantic generalizations.<sup>10</sup>

---

<sup>10</sup>It could be that learners are able to ignore the “noise” (or evidence against postulating a morpheme) due to the large number of pseudo-prefixed words per prefix, and they are willing to postulate a prefix *x* as soon as they hear a few forms of shape *xy*, *xw*, *xz* that are clearly related, in meaning, to *y*, *w*, *z*, respectively.

On the other hand, notice that distributional generalizations are insensitive to the distinction between prefixed and pseudo-prefixed forms. For example, the string *de-* is a frequent word-initial string, even if many of the words in which it occurs do not contain the prefix *de-* from a semantic point of view. Once a learner started suspecting that *de-* is a prefix on the basis of distributional cues, she could pay special attention to the meaning of all the forms containing this string, and eventually discover the semantic properties of *de-*.

The previous discussion suggests that pure semantic learning is an inefficient, anti-economical strategy and that it is likely run into problems with certain types of affixes. Indeed, some of the experimental evidence presented in the literature on morphological processing indicates that speakers must also have been sensitive to other cues in morpheme discovery, or else they would not have assigned a morphological structure to semantically opaque words.

For example, Emmorey 1989 found a strong facilitatory priming effect between semantically opaque but morphologically related forms such as *submit* and *permit*, but not between matched phonologically related pairs such as *balloon* and *saloon* (see also the work discussed in 1.4.3.1 below). If learners were only relying on semantic cues, they would not treat forms such as *submit* and *permit* as morphologically complex, since these forms could not be analyzed as complex on the basis of their semantic content.

Similar cases provide strong evidence that learners are not entirely relying on semantic cues when assigning morphological parses to words. Notice, on the other hand,

---

However, Baayen and Schreuder's results indicate that, at least for prefixes such as *de-*, it is not unlikely that learners simply do not hear enough transparent forms to even be able to follow a strategy along these lines, independently of their ability to immediately notice the relevant forms in the middle of a majority of pseudo-prefixed words.

that a distributional learning strategy could easily lead the learners to conclude that a form such as *submit* is prefixed: for example, because of the occurrence of *-mit* in a number of words in which it is preceded by prefix-like strings (*submit, permit, admit, remit, commit...*)<sup>11</sup>

To conclude, the arguments and evidence presented here suggest that, while it is clear that semantic information plays an important role in morpheme discovery, it is not likely that learners use semantics as their only source of evidence.

### **1.3 The role of innate knowledge in morpheme discovery**

It is possible that certain aspects of morpheme discovery are driven by innate knowledge. At a very general level, speakers may be innately endowed with the knowledge that words can be composed of morphemes and with the set of possible morpheme types. Furthermore, the morpheme discovery strategies *per se* may be innate. For example, the learners could innately know that they have to pay attention to distributional cues such as the ones discussed in this study.

Moreover, learners may be endowed with more specific forms of knowledge relevant to the morpheme discovery task. For example, the set of all the possible meanings and syntactic functions that affixes can express could be innate.

---

<sup>11</sup>The existence of semantically empty morphemes with a purely morphosyntactic function (Aronoff 1994, mentioned in footnote 7 above) also constitutes evidence against the hypothesis that learners discover morphemes on purely semantic grounds.

However, since many properties of morphemes are language-specific, it is not possible for learners to succeed in morpheme discovery on the sole basis of innate knowledge. For example, learners cannot use (only) innate knowledge to discover which particular set of meanings are expressed by morphemes in their language, nor to discover which morpheme types are present in their language nor, of course, to discover which particular strings constitute the morphemes of their language.

Thus, while innate knowledge may provide the learner with general tools helping her in morpheme discovery, it is clear that the morpheme discovery task cannot be completed on the sole basis of this kind of knowledge.

#### **1.4 Distributional cues in morpheme discovery**

Having discussed how other types of evidence can help learners during morpheme discovery, and having argued that none of them is likely to be sufficient by itself, I now turn to the main focus of this study, i.e. the nature and role of distributional cues in morpheme discovery. I first present some basic ideas about how distribution-based morpheme discovery could work (these are the same ideas which are implemented in the computational model I will present in chapter 3). Then, I discuss my view about the possible role played by distributional cues in morpheme discovery. Finally, I present some empirical evidence showing that, at the very least, adult speakers are sensitive to the distributional properties of morphemes, and I discuss the significance of semantically opaque but morphologically complex forms as evidence for distributional learning, since evidence of this nature will be presented in chapter 4 below.

### 1.4.1 Distribution-based morpheme discovery: general strategies

While this is not necessarily the only kind of distributional evidence a learner could use, the heuristics explored here are based on the observation that morphemes are *syntagmatically independent* units. This is one of the most basic distributional facts about morphemes (and linguistic units in general): If a substring corresponds to a morpheme, then that substring can occur in different contexts (words) independently of the substrings surrounding it.

According to this definition, if a string is syntagmatically independent, the string can *in principle* occur in different, independent contexts. However, children looking for morphemes have no way of telling whether a string they hear could in principle occur in different contexts; they can only observe that some strings *do* in actuality occur in a number of different words. Thus, actual frequency of occurrence of a string must be used by children as a heuristic approximation of the more abstract property of potential occurrence in different contexts.

The most obvious morpheme-searching heuristic based on the syntagmatic independence of morphemes is the following: A learner should look for substrings which occur in a high number of different words; these strings are likely to be morphemes. The rationale for this heuristic is the following: If a substring occurs in a high number of different words, then it is likely (but not necessary: see below) that the string is syntagmatically independent from the strings it occurs with.<sup>12</sup>

---

<sup>12</sup>Notice that the reverse of what is stated in this heuristic is not true: stems and monomorphemic words are morphemes even if they occur in only one or few words. Distributional independence implies that

Even a very primitive heuristic of this sort can provide learners with surprisingly good evidence. Consider for example the list of the ten most frequent word-initial two letter strings in the PHLEX database:<sup>13</sup>

(1)	<i>String</i>	<i>Type Frequency</i>
	co	1345
	in	992
	re	955
	pr	734
	de	645
	ma	611
	ca	598
	st	590
	di	551
	pa	523

The ten most frequent three letter word-initial strings in the PHLEX database are:

---

morphemes, unlike random, linguistically insignificant strings, can in principle occur in a high number of independent contexts, but it does not imply that they will.

<sup>13</sup>The PHLEX database (Seitz, Bernstein, Auer and MacEachern 1998) contains the 18460 most frequent word types in the Brown corpus (Kucera and Francis 1967) and the 12,118 word types listed in the Hoosier Mental Lexicon (Nusbaum, Pisoni and Davis 1984), in orthographic and phonemic transcription, with token frequency information.

(2)	<i>String</i>	<i>Type Frequency</i>
	con	534
	pro	358
	dis	295
	com	261
	pre	250
	int	233
	per	186
	tra	183
	sta	173
	par	165

Of the ten most frequent two letter word-initial strings in the PHLEX database, five are actual English prefixes (*co-*, *in-*, *re-*, *de-*, *di-*). Furthermore, the string *pr* constitutes the beginning of two three letter strings (*pro-* and *pre-*) which are among the ten most frequent three letter strings in the PHLEX database and are actual English prefixes.

Of the ten most frequent three letter word-initial strings in the PHLEX database, six are actual English prefixes (the allomorphs *con-* and *com-*, *pro-*, *dis-*, *pre-*, *per-*). The strings *int*, *tra* and *par* constitute the beginnings of longer prefixes (*inter-*, *intra-*, *trans-*, *para-*).

These data show that even a primitive distributional strategy (such as “look for substrings occurring in a high number of words”) can help a learner searching for the morphemes of her language. However it is also clear that, to be truly effective, this



distributional strategy should be greatly refined. Besides the fact that many of the strings in (1) and (2) are not actual English prefixes, notice that, by considering only two and three letter strings, we are artificially controlling for string length. When the type frequency of strings of all possible lengths is compared, the high type frequency criterion becomes very problematic. At one end, single characters / phonemes are, of course, extremely frequent (the ten most frequent word-initial strings in the PHLEX database, when strings of all lengths are compared, are: *s, c, p, a, d, m, b, r, t* and *i*). At the other end, longer strings, including long strings corresponding to morphemes, will tend to have a low frequency when compared to shorter strings.

The main problem with the high type frequency heuristic is that it is not sensitive to an important characteristic of morphologically complex words: It is not sufficient for a word to contain a potential morpheme to be classified as complex -- true morphologically complex words can be exhaustively parsed into morphemes. Thus, a learner should not simply consider the frequency of occurrence of a potential prefix, but how many times the string corresponding to the prefix occurs in words in which what follows is a potential stem.<sup>14</sup> Stems, like prefixes, are syntagmatically independent units. As such, they can in principle occur in different contexts independently of the strings surrounding them. Again, in order to guess whether a unit can in principle occur in different contexts, the learner must rely on the actual occurrence of the unit in different contexts in the input stream. It is unlikely that stems will actually occur in a high number of different contexts in the input, but it is reasonable to expect free stems of prefixed forms to also occur as independent words, and bound stems to occur in more than one affixed word.

---

<sup>14</sup>In a very broad sense, in which any constituent or sequence of constituents that can follow a prefix is labeled as a stem.

A prefix-searching strategy based not only on absolute frequency of word-initial strings, but on frequency of occurrence before potential stems, will take care, in part, of the problem arising from the comparison of strings of different length. A string such as word-initial *s* occurs in a high number of words, but in many of these words it is not followed by potential stems. Contrariwise, a string such as word initial *over* occurs in a small number of words, but in many of these words it is followed by potential stems.

I discussed the previous heuristics as prefix-searching heuristics, but of course they can also help the learners in deciding whether words are complex or not. If a word contains a frequent word-initial string, such as *re-*, it is more likely that the word is morphologically complex. More importantly, the possibility of exhaustively parsing a word into morphemic components constitutes the basic distributional criterion to decide whether a word is morphologically complex or not. A word like *reason* should not be treated as prefixed, even if it begins with the string *re-*, because what remains when we remove *re-* is not a potential stem.

Another heuristic based on syntagmatic independence provides further help in determining the morphological status of words. Words, like morphemes, are syntagmatically independent units. If a string corresponds to a word of the language, that string can occur in different sentences, surrounded by independent strings (other words). Of course, the most obvious application of this property is as a heuristic helping to segment sentences into words (indeed, syntagmatic independence is the basis of the sentence segmentation algorithm proposed by Brent and Cartwright 1996, described in 2.6 below).

Here, we assume that children undertake morpheme discovery after they have successfully completed the sentence segmentation task (see discussion in 3.2.11 below). The property of syntagmatic independence of words, however, can also help learners trying to decide whether a form is morphologically complex or not, in the following way: If

a word frequently occurs in the input stream, i.e., if a word has a high token frequency,<sup>15</sup> the learner has strong evidence that the word constitutes an independent linguistic (lexical) unit of the language and, thus, it should not be decomposed into morphemes, even if it could be in principle analyzed as complex. *Vice versa*, if a word occurs very rarely, the learner has less evidence that the word constitutes an independent lexical unit. Thus, if the word can be decomposed into morphemes, it is reasonable to hypothesize that the word is indeed morphologically complex.

If a morphologically complex word is very frequent, the word is likely to have its own lexical entry, distinct from the entries of its component parts (at the very least, for reasons of ease of lexical access). However, once a word has an independent lexical entry, the word can acquire its own semantic features and thus it is likely to lose, over the course of time, its connection with its component parts. In other words, high frequency words are less likely to be morphologically complex because, even if they *were* complex from an etymological point of view, they are likely to have acquired a lexicalized meaning due to heavy usage.

At the other extreme, productively formed complex words must be *hapax legomena* (words with a token frequency of 1), or in any event have a very low token frequency. Indeed, Baayen has shown in several studies (see for example Baayen 1994, Baayen and Lieber 1991) that the number of *hapax legomena* containing a certain morpheme is a good indicator of the productivity of the morpheme. If a morpheme is productive, then the

---

<sup>15</sup>I assume that in general the token frequency of a word (the number of times the word occurs in the input corpus) will be highly correlated with the number of times the word occurs in different sentences, surrounded by different words. Thus, I consider token frequency a reasonable estimate of the frequency of occurrence of a word in different contexts.

morpheme is often used to create nonce forms, and nonce forms by definition have a token frequency of 1.

Thus, all else being equal, a learner should be more willing to guess that a word is complex if the word has a low token frequency than if the word has a high token frequency. As a form of evidence supporting the hypothesis that token frequency and likelihood of lexicalization are correlated, consider the following data.

In (3), I list the twenty most frequent words beginning with orthographic *re-* from the PHLEX database (token frequency values in parenthesis):

- (3) really (275), real (258), result (244), reason (241), red (197), required (181), return (180), recent (179), report (174), read (174), research (174), reached (169), religious (165), received (163), rest (163), results (149), ready (143), reading (141), remember (138), record (137)

None of the words in (3) is non-controversially prefixed from a synchronic point of view. On the other hand, consider the list in (4), which is composed of twenty randomly selected words from the seventy-five PHLEX *hapax legomena* beginning with *re-*:

- (4) referendum, regalia, resplendent, repugnance, restive, reorganize, rendition, reopen, regain, renown, regimentation, revery, rend, rejoice, replica, revitalize, retch, reimburse, rejoinder, relict

Probably most morphologists would agree that the words *reorganize*, *reopen*, *regain*, and *revitalize* are synchronically prefixed. Similarly, (5) lists the twenty most frequent words beginning with *de-* in the PHLEX database:

- (5) development (333), death (276), department (230), dead (174), developed (170), defense (167), deal (143), decided (141), degree (125) described (120), decision (119), determined (119), design (114), deep (109), democratic (109), designed (108), determine (107), despite (104), demand (102), develop (89)

None of the words in (5) is non-controversially prefixed. On the other hand, consider the list in (6), which is composed of twenty randomly selected words from the fifty-two PHLEX *hapax legomena* beginning with *de-*:

- (6) detain, debonair, derogate, decompose, depress, deem, dehumanize, delectation, dexterity, deprivation, delimit, debatable, deceive, detach, decentralization, deadweight, desegregate, deity, derogatory, derivative

Probably most morphologists would agree that at least the words *decompose*, *dehumanize*, *decentralization* and *desegregate* are synchronically prefixed. Finally, in (7) I listed the twenty more frequent words beginning with *in-* in the PHLEX database (words beginning with *inter-* not counted):

- (7) into (1789), information (269), individual (239), increase (195), inside (174), instead (173), including (171), industry (171), indeed (162), involved (147), increased (146), industrial (143), influence (132), include (113), income (109), indicated (108), institutions (98), included (97), inch (89), inches (86)

Again, none of these words is non-controversially prefixed. Consider instead the following list of twenty randomly selected words from the one hundred and fifteen PHLEX *hapax legomena* beginning with *in-* (words beginning with *inter-* were excluded from the random sampling):

- (8) informality, inane, indigent, insurgent, inkling, inexpressible, infirmity, instigation, incongruous, inconspicuous, inmate, inoperable, insatiable, infertile, indonesian, indelicate, incalculable, invulnerable, incorrigible, indivisible

Probably, most morphologists would agree that at least the words *informality*, *inexpressible*, *incongruous*, *inconspicuous*, *inoperable*, *insatiable*, *infertile*, *indelicate*, *incalculable*, *invulnerable* and *indivisible* are synchronically prefixed.

The small survey I just presented supports the intuition that the likelihood of a form being prefixed is (loosely) correlated to the inverse of the token frequency of the form. However, it is not clear that learners would *a priori* (innately?) know that frequent words are likely to develop idiosyncratic meanings, and thus become morphological opaque.<sup>16</sup> However, a heuristic disfavoring morphological analysis of frequent words could naturally emerge from some basic assumptions about the structure of language that a distributionally aware learner could make.

If the learner is looking for prefixes, she could assume, for heuristic purposes, a simple model like the following: Prefixed words are formed by independently selecting a prefix from the list of prefixes of the language and a stem from the list of stems of the

---

<sup>16</sup>Learners probably store frequent complex words in the lexicon as units, but this does not imply, *per se*, that they should chose to treat these words as morphologically simple.

language.<sup>17</sup> Under this assumption, the learner should expect that the probability of occurrence -- and hence the relative frequency -- of a prefixed form will be approximately related to the product of the probabilities of occurrence -- relative frequencies -- of its component parts (this follows from the fact that the selection of the prefix and the selection of the stems are treated as independent events).<sup>18</sup> Thus, if a learner encounters a word that could be split into morphemes, but whose relative frequency is significantly higher than the product of the relative frequencies of the potential constituents, the learner will be inclined to conclude that the word is not complex, since it occurs more frequently than what would be predicted by the morphological model assumed.

What emerges from this discussion is a more sophisticated strategy than the one proposed above: The frequency threshold above which learners will decide that a word is not complex will vary from word to word, depending on the frequency of the potential constituents. However, since in general the product of two relative frequencies is going to be rather low, the more general heuristic I proposed (high frequency words are less likely to be complex) is probably a good approximation to the more sophisticated strategy deriving from the assumption that morphologically complex words are formed by independently selecting a prefix and a stem.

The most obvious application of this heuristic concerns the decision to treat words as complex or simple. However, the heuristic also serves as a prefix-searching strategy, in the sense that substrings occurring in a high number of low frequency words will be more

---

<sup>17</sup>Clearly, this model is too simple, but it could serve as a first, heuristic approximation of how morphology works.

<sup>18</sup>For the notions of probability theory employed here, see, for example, the first chapter of Roman 1996.

likely to be treated as morphemes than substrings occurring in a high number of high frequency words.

Summarizing, in this section I have proposed distributional heuristics deriving from the observation that linguistic units are syntagmatically independent. Putting the various pieces together, I suggested that a learner could hypothesize that frequent word-initial strings are likely to be prefixes. They are more likely to be prefixes if they frequently occur before strings which are potential stems, i.e. strings which occur as independent words and/or in other potentially complex words. Words which can be exhaustively parsed into component morphemes should be treated as morphologically complex by the learner. Moreover, all else being equal, the learner should be more willing to treat low rather than high frequency words as morphologically complex. As I will show in chapter 3, the MDL criterion provides a natural way of implementing a distribution-learning model based on the heuristics discussed here.

While I illustrated basic ideas about distributional morpheme discovery with examples from prefixation, the same strategies could be applied to the search for suffixes, or infixes, templates and other forms of discontinuous morphological constituents (see Spencer 1991: chapter 5 for a review), given that the property of syntagmatic independence is not necessarily restricted to adjacent sequences of segments or letters. However, the number of possible morphological parses of words that a learner has to evaluate when looking for discontinuous morphological constituents is much higher than in the search for prefixes and suffixes. Possibly, a learner could trim down the number of discontinuous parses to be explored by using phonological and prosodic constraints on the distribution of infixes, templates, etc. (see the work on prosodic morphology by McCarthy and Prince 1986, 1993 and others).



Thus, at the very least, it is unlikely that distributional cues would be sufficient by themselves to successfully identify discontinuous morphemes. How much the learners can discover about the suffixes and prefixes of their language on the basis of distributional cues is an open empirical question. The results of the simulation I will present in chapter 4 provide some preliminary insights into the limits inherent to distributional evidence.

#### **1.4.2 The role of distributional learning in morpheme discovery**

I have suggested that it is plausible that learners use a number of strategies to discover the morphemes of their language, relying on innate knowledge, exploiting cues from various linguistic domains and keeping track of distributional patterns. I argued that none of these strategies is likely to stand by itself as the only one employed by learners. In this section, I will defend a specific hypothesis about the role played by distributional cues in morpheme discovery, i.e., that distributional cues play a primary role in the earliest stages of morpheme discovery.

Distributional information can be straightforwardly extracted from the data without requiring any prior linguistic knowledge. Moreover, distributional cues such as the ones discussed here reflect such basic properties of morphological usage that they are likely to be universal, independent of language-specific features. For example, in any language affixes will correspond to strings occurring in a number of different words and, thus, will tend to have a high type frequency.

Thus, it seems reasonable that, if learners rely on distributional cues at all, they would use them earlier on to assign preliminary parses to the words they hear, using these preliminary parses to extract linguistic generalizations about the morphology of their

language, and later using the linguistic knowledge gathered in this way to refine the coarse guesses on morphological structure made on the basis of distributional strategies.

For example, we discussed above the case of the English prefix *de-*. According to the data presented by Schreuder and Baayen 1994, the large majority of English words beginning with a phonological or orthographic string identical to this prefix is not semantically transparent. Thus, English learners could easily overlook the relevant semantic generalization. More generally, even if certain morphemes probably *do* mostly occur in semantically transparent forms, we observed that it is not plausible that learners try each possible combination of each possible phonological and semantic decomposition of each word they hear, to check whether the word is semantically transparent.

It is plausible that distributional properties alert the learners to the possibility that certain strings are morphemes, making it easier for them to notice the systematic semantic patterns that occur in a certain number of words containing those strings. For example, the fact that word-initial *de* occurs in a high number of words could lead the learners to hypothesize that *de-* is a prefix. Subsequently, the learners can look for semantic similarities among the potentially prefixed words beginning with *de-*. Ultimately, *de-* is a prefix because it has certain semantic and syntactic properties, which allow the speakers to understand and form new words using it, and not because the string *de-* has a peculiar distribution. However, easy-to-extract distributional evidence provides the learner with a simple heuristic strategy to search for potential morphemes.

Moreover, learners must probably go through the step of extracting some form of distributional information anyway, for the purpose of other forms of morpheme discovery. For example, if a learner is evaluating the likelihood of possible form-to-meaning mappings, it is very likely that she has to keep track of how many times a certain string is associated with various semantic features. But if learners have to collect distributional

evidence anyway, then it would be strange if they did not try to extract some useful information from distributional data by themselves before they combine them with other types of evidence.

If the hypothesis presented in this section is correct, then, by modeling morpheme discovery as a purely distribution-driven task, we are not simply making a useful abstraction in order to assess to what extent distributional cues could be useful by themselves: we are actually modeling the first stage of morpheme discovery, before bootstrapping based on linguistic knowledge kicks in.

### **1.4.3 Distribution-driven learning in morpheme discovery: the empirical evidence**

The (admittedly inconclusive) arguments presented in the previous section support the view that, if learners *do* resort to distributional strategies in morpheme discovery, then it is likely that they use such strategies in the earliest stages of this process. I am not aware of empirical evidence in favor of this claim, and this study will not be providing evidence in its support. However, I will present some indirect evidence in favor of the more general hypothesis that humans do rely on distributional strategies during morpheme discovery, and I will discuss the nature of the data supporting this hypothesis that will be presented in this study.

Unfortunately, as far as I know, there are no studies presenting data on how children discover the morphemes of their language and on whether they are sensitive to the

distributional properties of these morphemes.<sup>19</sup> However, some studies have shown that adult speakers are sensitive to distributional properties of morphemes. Stolz and Feldman 1995 report results from a segment shifting experiment (a production task meant to simulate spontaneous speech errors in a laboratory setting)<sup>20</sup> showing that the time it takes English subjects to shift a suffix from a word to another is significantly correlated with the type frequency of the suffix (and with the ratio of truly suffixed to pseudo-suffixed words containing a word-final string identical to the suffix).<sup>21</sup>

Laudanna and Burani 1995 report results from a series of experiments which show that two distributional properties of prefixes (their length and the ratio of truly prefixed forms to pseudo-prefixed forms) have a significant effect on Italian subjects' performance in a visual lexical decision task. The longer a prefix and the higher the corresponding truly

---

<sup>19</sup>Saffran, Aslin and Newport 1996 have shown that 8 month old infants are sensitive to distributional information relevant to the task of continuous speech segmentation.

<sup>20</sup>In this task, speakers have to separate a designated segment from a word and attach it to a new word as quickly as possible. For example, if the designated segment is *en*, the source word is *harden* and the target word is *bright*, the speaker has to say the word *brighten* as soon as possible. It has been shown that speakers are faster at this task if the designated segment functions as a morpheme in the target word than if it does not (e.g., it is easier to detach *en* from *harden* than from *garden*).

<sup>21</sup>The variable used by Stolz and Feldman in this analysis is the mean difference in shifting time from complex words and control words (simple words ending in a string identical to the relevant suffix). Thus, a higher value of this variable indicates faster shifting times from actually suffixed words. Thus, the positive correlation between this measure and type frequency (and /or ratio of suffixed to pseudo-suffixed words) and this measure indicates that speakers are faster at shifting suffixes that have a higher type frequency (and/or ratio of suffixed to pseudo-suffix words).

prefixed/pseudo-prefixed ratio, the longer it takes speakers to decide that nonsense forms beginning with a string identical to the prefix are not real words (prefix length also has a significant effect on the number of errors made by subjects).

In Baroni *in press*, I show that the likelihood that potentially prefixed words are treated as complex for the sake of a morphophonological process of Northern Italian is significantly correlated with prefix length.

While these studies do not provide evidence in favor of the claim that learners use distributional cues in morpheme discovery, they show that adult speakers are sensitive to some distributional properties of morphemes, which implies that the language processing component of humans is, at some level, keeping track of such properties.

The morpheme discovery simulation presented in this study shows that simple distributional cues can be very effective in morpheme discovery. Moreover, the comparison of the parses assigned by the automated learner to semantically opaque forms with the parses assigned by English speakers to the same forms provides evidence that humans use distributional cues similar to the ones used by the learner when attributing a morphological structure to words. Thus, I turn now to the discussion of previous studies showing that speakers are aware of the morphological structure of some semantically opaque words, and I explain why this type of word can play an important role in providing evidence for the reality of distributional learning.

### **1.4.3.1 Distributional learning and the status of morphologically complex, semantically opaque words**

An interesting and challenging aspect of modeling how children learn to segment words into morphemes is that we are actually not sure of what the “correct” output of this process should be. Of course, if our algorithm fails to parse a word like *reconstruct* as morphologically complex, or if the algorithm treats *giraffe* as a prefixed word, we have reasons to worry. However, there are many intermediate cases of words (such as *resist*, *resume* etc.) whose morphological status is not clear.

Several studies have presented evidence from different experimental tasks that speakers are aware of the morphological structure of words that are (partially or completely) semantically opaque. I mentioned in section 1.2.3 the findings of Emmorey 1989. Other studies providing empirical evidence for the claim that some semantically opaque words are treated as complex by adult speakers include Bentin and Feldman 1990, Feldman and Stotko *unpublished* (quoted in Stolz and Feldman 1995), Baroni *in press*, Baayen, Schreuder and Burani *submitted*, and Roelofs and Baayen *submitted*.

Bentin and Feldman 1990 report the results of a series of repetition priming experiments in Hebrew showing, among other things, that there is a facilitatory effect among words whose morphological relation is semantically opaque. Feldman and Stotko also used a repetition priming paradigm to show that semantically opaque suffixed words prime their stem in English. In Baroni *in press*, I show that Northern Italian speakers treat some prefixed semantically opaque words as complex for the sake of a productive morphophonological rule. Baayen, Schreuder and Burani *submitted* present data from a series of lexical decision experiments in Dutch, showing, among other things, that the frequency of the constituents of semantically opaque words affects lexical decision speed.

Roelofs and Baayen *submitted* show that some semantically opaque compounds of Dutch behave like semantically transparent compounds for the purposes of syllabification and that some semantically opaque prefixed words of Dutch behave like complex for the sake of a morphophonemic process. Moreover, Roelofs and Baayen show that semantically opaque prefixed words pattern with transparent prefixed words in an implicit priming experiment.<sup>22</sup>

I will not discuss the results of these studies in more detail here. The important point for current purposes is that, when taken together, these results provide evidence for the claim that speakers treat some semantically opaque words as morphologically complex.

I observed above that these findings constitute a form of evidence against the hypothesis that morpheme discovery is entirely semantics-driven. Moreover, they suggest a way to test distribution-driven learning models: The morphological parses that an

---

<sup>22</sup>Marslen-Wilson, Tyler, Waksler and Older 1994 found that semantically opaque affixed words did not behave like transparent affixed words in a cross-modal priming experiment (in the sense that they did not significantly prime morphologically related forms). This result is in apparent contradiction with the ones discussed in the text. However, Marslen-Wilson and collaborators presented the target immediately after the prime, and it has been observed (see discussion in Stolz and Feldman 1995) that purely semantic effects are strongest at very short lags. Thus, the strength of semantic effects in the immediate priming paradigm could have obscured weaker, purely morphological effects shared by transparent and opaque forms (notice, however, that the paradigm used by Marslen-Wilson and collaborators is similar to the one used by Emmorey 1989, who did find purely morphological effects). Alternatively, the apparent contradiction could be due to the fact that morphological complexity is a gradient property, and semantically opaque forms have an intermediate degree of complexity, so that they pattern with complex forms in certain respects and with simple forms in other respects.

automated learner assigns to semantically opaque but potentially complex words (such as *recite*) can be compared to morphological parses assigned to the same words by human subjects (as recorded in some form of behavioral test).<sup>23</sup> If the distribution-driven algorithm produced parses similar to the ones assigned by humans, this would first of all constitute strong evidence that the automated learner is generating morphological analyses analogous to the ones learned by human beings.

Moreover, the convergence between the distribution-driven learner and human beings in parsing semantically opaque words would also provide evidence that humans adopt distributional strategies similar to the ones implemented by the automated learner. If both the automated learner and the human subjects treated, say, *resist* as a complex form but, say, *resume* as monomorphemic, the most plausible explanation of this fact would be that, when humans stored such forms in their lexicon, they followed distributional criteria similar to the ones used by the automated learner.

Thus, if a distribution-based model of morpheme discovery produced parses of opaque words matching those assigned by humans, this would provide a form of direct evidence for distribution-driven morpheme discovery (and hence distributional learning in general) which is not available in other domains. For example, consider the related issue of modeling how children learn to segment utterances into words (see Brent and Cartwright 1996 and the references quoted there). Even if a distribution-driven model of sentence segmentation were completely successful, this would still be a weaker form of evidence for the relevance of distributional cues in language acquisition, since all the words discovered

---

<sup>23</sup>Of course, this approach is more convincing if the semantically opaque words under analysis are not marked by obvious syntactic or phonological cues of morphological complexity.



by the model could also have been discovered by human learners on the basis of semantic (or syntactic) cues.<sup>24</sup>

From the point of view of morphological theory, if the distributional model correctly predicted which semantically opaque words are treated as complex by speakers, this would provide the basis for a (partial) explanation of the fact that speakers do treat some semantically opaque words as morphologically complex: They do so because they used distributional schemes to search for the morphemes of their language, and these schemes lead them to analyze some words as morphologically complex even in the lack of semantic cues supporting the complex analysis.

In chapter 4, I present the results of a survey in which adult English speakers rated a set of potentially complex but semantically opaque forms, and I compare the speakers' ratings to the morphological parses generated by the automated distribution-driven learner, showing that, indeed, speakers' ratings and learner's parses are correlated.

## 1.5 Summary

In this chapter, I discussed the problem of morpheme discovery and I examined different types of evidence which could help the learners in this task. I have presented arguments

---

<sup>24</sup>Idiomatic phrases such as *kick the bucket* are not the equivalent of semantically opaque morphologically complex forms such as *permit*. First, idiomatic phrases also have a literal, semantically transparent meaning, and it is unlikely that speakers are not aware of this meaning. Second, words occurring in idioms also occur in non-idiomatic sentences. This is not the case of a bound stem like *-mit*, which occurs only in opaque forms.

supporting the claim that, while several morpheme-searching strategies can be useful to learners, there is no single strategy which could fully account on its own for the process of morpheme discovery. I then proposed some general distributional heuristics that could be useful to learners, and I discussed what could be the role of distributional learning in morpheme discovery, and how the comparison of the morphological parses assigned by a distribution-based model and by human beings to a set of semantically opaque but potentially complex words could provide important evidence in favor of the relevance of distributional learning in language acquisition.

## **Chapter 2**

### **Other models of morpheme discovery**

#### **2.1 Introduction**

In this chapter, I review models of morpheme discovery proposed by other authors. While there has been considerable work on modeling the similar task of utterance segmentation, very few explicit models of the morpheme discovery process have been proposed.<sup>25</sup> Interestingly, all the models proposed (at least, the ones I am aware of) are either entirely distribution-driven, or characterized by a mixture of distributional and syntactic information. None of the models proposed makes use of semantic information.

Harris 1955 treats utterance segmentation into words and word segmentation into morphemes as the same problem, and presents an algorithm which takes unsegmented and unlabeled utterances as its input and segments them into words or morphemes. Harris' algorithm uses a boundary-based segmentation strategy, in which words/morphemes are identified on the basis of the patterns which follow or precede them.

The suffix discovering algorithm discussed in Brent 1993 (see also Brent, Murthy and Lundberg 1995) represents, as far as I know, the first attempt to apply the Minimum

---

<sup>25</sup>I do not discuss here models, such as the one proposed by Albro 1998, in which the input contains information on which forms are paradigmatically related. By providing this information, models of this kind are enormously simplifying the morpheme discovery task, typically because morpheme discovery is not the main focus of the study (the ultimate goal of Albro's algorithm, for example, is to find the positional classes of the inflectional morphemes of a language).

Description Length principle (Rissanen 1978) to the problem of morpheme discovery. The algorithm is presented in two versions: in one version, the algorithm is entirely distribution-driven; in the other version, the algorithm also exploits syntactic category information.

Mikheev 1997 presents a part-of-speech tagger which happens also to work as a good morpheme discovering algorithm. Mikheev's algorithm exploits both distributional and syntactic category information.

Goldsmith *submitted* presents an algorithm which discovers stems, suffixes and "signatures" (elementary forms of paradigms). Goldsmith's algorithm is entirely distribution-driven, and like the one by Brent and the one presented here, it is based on the Minimum Description Length principle.

Finally, I will discuss the model proposed by Brent and Cartwright 1996 for the utterance segmentation task, since their work provided the main source of inspiration for the morpheme discovery procedure I present in this study.

## **2.2 Harris 1955**

Harris 1955 proposes an algorithm to segment utterances into morphemes. The basic intuition behind Harris's approach is that it is easier to predict the following segment within the same morpheme (or word) than across morpheme (or word) boundaries. Thus, Harris proposes the following procedure:

- Count how many phonemes could follow the string formed by the first  $n$  segments of the utterance in some well-formed utterance, with  $n$  ranging from 1 to the number of

segments in the utterance. The number of phonemes that can follow a string in a well-formed utterance is the *number of successors* of the string.

- Insert a morpheme boundary after each string of  $n$  segments whose number of successors is higher than or equal to the number of successors of the  $(n-1)$  and  $(n+1)$  strings (i.e., insert a morpheme boundary after each *local maximum*).

Harris presents an analysis of the sentence *He's clever* /hiyzklevər/ as an example. Following his procedure, we first count how many phonemes can occur after /h/ in some well-formed utterance: the number of successors of the string /h/ turns out to be 9. Then, we count the number of successors of /hi/, which is 14, and so on until the end of the utterance. The number under each segment in (9) is the number of successors of the string ending with that segment (as computed by Harris):

(9)	h	i	y	z	k	l	e	v	ə	r
	9	14	29	29	11	7	8	1	1	28

The three-phoneme string /hiy/ has a higher number of successors than the two-phoneme string /hi/ and the same number of phonemes as the four-phoneme string /hiyz/. Thus, a morpheme boundary is inserted after /hiy/. Following Harris' procedure, morpheme boundaries are also inserted after /hiyz/, /hiyzkle/, /hiyzklevər/, and we end up with the segmentation: /hiy#z#kle#vər/.

The algorithm guessed all the right morpheme/word boundaries, but it also inserted an unlikely boundary between /kle/ and /vər/. This problem can probably be solved by requiring that boundaries be inserted only after strings whose number of successors is higher than a certain threshold.

Another example presented by Harris is the sentence *He's quicker* /hiyzkwikər/:

(10)	h	i	y	z	k	w	i	k	ə	r
	9	14	29	29	11	6	10	28	14	28

This time, the segmentation generated by Harris' algorithm is: /hi#z#kwik#ər/. Thus, Harris' algorithm captures the fact that the same string (/ər/) is a morpheme in *quicker* but not in *clever*.

The algorithm is able to distinguish between *quicker* and *clever* in virtue of the fact that *quick* is an independent word and, as such, it can occur in a sentence before most English sounds, whereas *clev* is not an independent word, and thus it can occur only before the restricted number of strings which make it a real word.

In order to apply Harris' algorithm to the prefix searching problem, we must adopt one of the modifications of the basic procedure proposed in the paper, i.e. we must apply the procedure backwards, counting the possible predecessors of a string. Consider the difference between (orthographic) *reprint* and *regatta*. If we applied Harris' basic procedure to these words (in isolation, or embedded in identical sentences), the procedure would fail to recognize that the *re* in *reprint*, but not the one in *regatta*, is a morpheme, since the two strings *re* are identical, and hence they have the same successor count.

If we apply the procedure backwards, we compare the strings *print* and *gatta*. Since the first string is an independent word of English, it can be preceded by more strings than the second word, which can occur only after the restricted number of strings (1?) which make it a real word.

An obvious problem with Harris' algorithm is that, in general, it is unlikely that it will be able to recognize bound stems. The reason why the algorithm can distinguish

*quicker* from *clever* and *reprint* from *regatta* is that in both pairs the morphologically complex member contains a stem which is an independent word. Typically, languages impose stricter phonotactic constraints within words than across word boundaries. For example, *quick* can occur before /f/ in the phrase *a quick phonetician*; but /kf/ could not be a (possible) word-internal cluster of English (except in compounds). Thus, stems that are also independent words can typically be preceded/followed by a very large number of phonemes, whereas this is not necessarily the case with bound stems, whose distribution will be typically subject to stricter word-internal phonotactics.

Furthermore, it is not the case that all the (possible) phonotactically well-formed combinations of English segments form existing English words. Even if a word in which *gatta* is preceded by *r* could be a well-formed English word (*ergatta*, for example), such a word does not exist. This fact about the English lexicon limits the successor and predecessor counts of strings which are not autonomous words. On the other hand, the combinatorial possibilities of syntax make it possible for words to be preceded by virtually any word-final segment of the language, and followed by virtually any word-initial segment.

Bound stems need an affix to form a word and hence their predecessor/successor counts are limited by word-internal phonotactic constraints and by the fact that they can be preceded and followed by only a limited number of affixes. Consider as an example the Italian suffixed form *ver-i* /veri/ (*ver-* is a bound stem meaning ‘true’, and *-i* is the masculine plural suffix). Since *ver-* is not an autonomous word, it can be followed only by the strings (most of them suffixes) which make it a word. These strings begin with a very limited number of segments. Consequently, *ver-* has a low successor count and it is not recognized as a morpheme by Harris’ procedure:

(11)	v	e	r	i
	7	17	11	27

Harris' procedure fails to recognize *ver-* as a morpheme because this string is not an independent word of Italian, and consequently it can occur only before a limited number of suffix-initial segments.<sup>26</sup>

In English and some other Indo-European languages (such as Italian), productive prefixation<sup>27</sup> always involves free stems. Thus, if we accepted the conservative assumption that a morpheme-discovering algorithm should look only for productively formed complex words, Harris' algorithm could still be used in a prefix-searching procedure. However, even within this conservative approach it is not reasonable to assume that learners *a priori* know that the productive prefixation rules of their language are restricted to free stems.

### 2.3 Brent 1993

Brent 1993 proposes an algorithm which discovers suffixes using the MDL criterion of Rissanen 1978 (see also Brent, Murthy and Lundberg 1995 for a more technical description

---

<sup>26</sup>The example in (11) shows another problem of the successor count procedure: in a language like Italian, in which vowels are typically followed by consonants, consonants are typically followed by vowels and there are considerably more consonants than vowels, the successor counts of vowels will tend in general to be higher than those of the surrounding consonants. In (11), the string *ve* would be wrongly identified as a morpheme for this reason. This problem was already noticed by Harris.

<sup>27</sup>*Productive* in the sense that it can be used to form new words.



of the same algorithm). As far as I know, Brent's work represents the first attempt to apply the MDL criterion to the problem of morpheme discovery. I will discuss the MDL criterion and its relation to plausible morpheme searching heuristics in 3.3, below.

Brent's basic approach can be summarized in the following way: Assume that a lexicon is a list of suffixes and stems, and that each word in the input was generated by combining a stem and a suffix (monomorphemic words have a zero suffix). Then, when comparing alternative lexica which could have generated the input stream, simply select the shortest lexicon, i.e. the lexicon represented by the smallest number of letters.

In order to understand why this is a sensible strategy, consider the following example (taken, with some simplifications, from Brent 1993). We want to select the best lexicon accounting for the following input:

(12)	walk	referral
	walks	refer
	walked	refers
	walking	dump
	referred	dumps
	referring	preferential

The input in (12) could have been generated by the following lexicon (I use the symbol  $\emptyset$  to represent the zero suffix attached to monomorphemic words):

(13) *stem list:*      *suffix list:*

walk	Ø
referr	s
refer	ed
dump	ing
preferenti	al

*total number of letters used to represent lexicon: 38*

An alternative lexicon which could have generated the input in (12) is the following:

(14) *stem list:*      *suffix list:*

wal	k
refer	ks
refe	ked
dum	king
preferent	red
	ring
	ral
	r
	rs
	p
	ps

ial

*total number of letters used to represent lexicon: 53*

Yet another lexicon which could have generated the input in (12):

(15) *stem list:*      *suffix list:*

walk      Ø

walks      d

walke      ng

walki      l

referre

referri

referra

refer

refers

dump

dumps

preferentia

*total number of letters used to represent lexicon: 75*

If we adopt the shortest lexicon criterion, we must select the lexicon in (13), which is represented by a total of 38 letters, vs. the 53 letters needed to represent the lexicon in (14)

and the 75 letters needed to represent the lexicon in (15). Clearly, (13) is also the lexicon corresponding to the most reasonable morphological analysis of the input (among these three).

Sensible morphological decompositions shorten the lexicon because true morphemes, unlike random substrings, occur in a number of different words. For example, if *walked* is decomposed into *walk* and *-ed*, and *referring* is decomposed into *referr-* and *-ing*, the same morphemes can be “recycled” to represent the input words *walking* (*walk* plus *-ing*) and *referred* (*referr-* and *-ed*), without need to add new items to the lexicon. On the other hand, if *walked* is decomposed into *wal-* and *-ked*, and *referring* is decomposed into *refer-* and *-ring*, we cannot recycle the same morphemes to derive other input words such as *walking* and *referred*, and thus we have to add more stems and suffixes to the lexicon, lengthening it.

The way in which representation length is actually computed in Brent’s model is more complex than simply counting the number of letters used to represent a lexicon, but this simple criterion illustrates the basic idea of how the model works. The most important difference between the simplified criterion used in the example above and the actual method to compute representation length adopted by Brent is that, using the actual criterion, lexica containing a small number of morphemes that occur in the input corpus very frequently are preferred to lexica in which corpus frequency is evenly distributed among morphemes.

While the criterion used to select the best lexicon constitutes the core of Brent’s algorithm, the algorithm must also generate a set of candidate lexica. Brent’s strategy is to allow the generation of a very large number of lexica, leaving the burden of finding the most sensible one entirely to the shortest lexicon criterion.

Still, to avoid an explosion of the search space, only lexical analyses of the input which meet the following constraints are evaluated as possible lexica (I will discuss below

some problematic aspects of these constraints): 1) only stem-suffix decompositions in which the stem is at least as long as the suffix are considered (this also has the function of minimizing the risk of treating prefix-stem combinations as stem-suffix combinations); 2) if a word-final substring is treated as a suffix, the substring is treated as a suffix in all the words ending with it. For example, if *-ing* is treated as a suffix, then not only *walking*, but also *string* must be represented as a suffixed form (*str+ing*).

It also follows from the latter condition that suffixes cannot be substrings of other suffixes. Consider a potential lexicon containing both the suffix *-ness* and the suffix *-s*. Given such a lexicon, any word ending in *ness* would violate the constraint requiring that a substring identical to a suffix is treated as a suffix in all the words ending with it. If a word like *kindness* was parsed as *kind+ness*, the constraint would be violated because this word ends in *s* but it is not treated as a word containing the suffix *-s*. If *kindness* was parsed as *kindnes+s*, the constraint would be violated because this would be a word ending in *ness* but not treated as containing the suffix *-ness*.

The algorithm was tested with corpora of different sizes (from 500 to 8000 words) sampled from the *Wall Street Journal*. The total number of real English suffixes (or other linguistically significant units, such as second members of compounds and suffix combinations) found by the model ranged from 6 (500 word corpus) to 55 (8000 word corpus). The number of errors (strings treated as suffixes that are not actual English suffixes) ranged from 1 (2000 word corpus) to 10 (500 word corpus).

Brent also presents a version of the algorithm which takes syntactic category information into account. Without going into details, this version of the model takes a corpus of tagged words as its input, and the entries in the suffix list are associated with one or more syntactic categories. All else being equal, this version of the model favors lexica with suffixes associated with a small number of categories, and suffix + category pairs

which occur in a high number of words in the corpus. Interestingly, the model incorporating syntactic information performs only marginally better than the model based on distributional information alone.

To conclude, Brent's MDL-based lexicon selection criterion constitutes an elegant and effective implementation of distributional morpheme-searching strategies of the kind I discussed in the previous chapter. Indeed, the lexicon selection criterion of the procedure I will present below is almost identical to the one adopted by Brent. However, the lexicon generation algorithm used by Brent has some obvious limitations, in that the constraints restricting the set of possible lexical analyses appear to be too strong.

First, it is obviously not always the case that stems are as long or longer than the morphemes they occur with (consider, for example, the word *ill+ness*, with a three letter stem and a four letter suffix). Second, it is also obviously not always true that all words ending with a substring identical to a suffix should be parsed as suffixed. For example, the word *string* ends in *-ing* but it clearly should not be parsed as *str+ing*. Moreover, learners should not exclude the hypothesis that some suffixes are substrings of other suffixes. For example, the English suffix *-s* is a substring of the English suffix *-ness*.

Notice that Brent explicitly claims that the aim of his procedure is to discover suffixes, and not to assign morphological parses to input words. In this perspective, only the last issue I mentioned (the ban against suffixes identical to endings of other suffixes) is problematic (as observed by Brent himself).

## 2.4 Mikheev 1997

Mikheev 1997 presents a part-of-speech tagging algorithm which takes a corpus of tagged words as its input and formulates part-of-speech guessing rules which can then be used to label an untagged corpus.

Some of the guessing rules predict the part of speech of a word by removing a word-initial string and checking whether the remainder is identical to a word which has a certain tag in the input corpus. For example, the rule in (16) says that if, by removing the word-initial string *un* from an unknown word we obtain a word that is tagged in the input corpus as a past participle, then the unknown word is an adjective:<sup>28</sup>

(16) <un> + *past-participle* -> *adjective*

If the past participle *known* was in the input corpus, the rule in (8) would correctly guess that *unknown* is an adjective.

Rules such as the one in (16) are essentially prefixation rules. Here, I will describe only how Mikheev's algorithm finds prefixation rules. Mikheev's suffixation rules are similar to the prefixation rules. I will not discuss how Mikheev's algorithm discovers suffixation and non-morphological part-of-speech tagging rules. I will also ignore the "rule merging phase" of Mikheev's algorithm and the way in which the performance of the algorithm as a part-of-speech tagger is evaluated.

Prefix guessing rules are extracted in the following way: for each pair of tagged words in the input corpus, the algorithm checks whether the longer of the two words is

---

<sup>28</sup>I am using a simplified version of Mikheev's notation.

composed of *prefix+short\_word*, where *prefix* is a non-empty word-initial string and *short\_word* is a string identical to the shorter of the two words. If the operation is successful, the system creates a rule according to the template in (17):

(17) *prefix + tag\_short -> tag\_long*

Where *tag\_short* is the part-of-speech tag of the shorter word and *tag\_long* is the part-of-speech tag of the longer word. For example, if *regain* and *gain* are tagged as (base forms of) verbs in the input corpus, the algorithm, when comparing these two words, will output the rule (18):

(18) *<re> + verb -> verb*

If the input corpus contains the noun *delivery* and the adverb *very*, the procedure will extract the rule (19):

(19) *<deli> + adverb -> noun*

If a rule extracted by this procedure has already been generated in the comparison of another pair, the frequency count of the rule is incremented.

After extracting all the possible prefixation rules in this way, the algorithm trims the rule set by eliminating all the rules with a very low frequency count (such as, presumably, the rule in (19)). Among the remaining rules, the algorithm tries to select the most effective ones: optimally, a rule should both apply to many forms and correctly predict the part-of-speech tag of the forms it applies to.



For each rule, the algorithm counts the number of forms in the input corpus meeting its structural description (i.e. to which the rule could in principle apply) and the number of forms in the input corpus for which the application of the rule would be successful. For example, if the verbs *make*, *remake*, *ally* and the adverb *really* are in the input corpus, the algorithm will count *remake* and *really* among the forms compatible with the rule in (18), since both forms are analyzable as *re-* plus verb; however, only *remake* will be counted among the forms for which the application of the rule is successful, since the rule, if applied to *really*, wrongly predicts that this word is a verb.

The algorithm calculates the estimated proportion of success ( $\hat{p}$ ) of each rule, which is the proportion of successful applications of the rule over the total number of words compatible with it:<sup>29</sup>

$$(20) \quad \hat{p} = \frac{\text{number of successful applications of the rule}}{\text{number of words compatible with the rule}}$$

The  $\hat{p}$  estimate is a measure of the accuracy of a rule. However, intuitively, we are more willing to trust a rule which successfully applies to 95 of the 100 words compatible with it than a rule which applies successfully to the only word compatible with it. The  $\hat{p}$  estimate does not take this intuition into account: a rule which successfully applied to the only compatible word would have the highest possible  $\hat{p}$  value (i.e. 1).

Thus, Mikheev's procedure, rather than considering  $\hat{p}$  *per se*, computes the lower confidence limit  $\pi_L$  of  $\hat{p}$ , which can be interpreted as the minimal expected value of the  $\hat{p}$  of a rule if the number of samples (= words compatible with the rule) were larger, assuming a certain confidence level (in (21),  $\alpha$  is set to .90):

---

<sup>29</sup>Mikheev computes these quantities over the word tokens in the input corpus.

$$(21) \quad \pi_L = \hat{p} - t_{.05}^{n-1} * \sqrt{\frac{\hat{p} (1 - \hat{p})}{n}}$$

In (21),  $t_{.05}^{n-1}$  is the coefficient of the  $t$ -distribution when  $\alpha = .90$  and there are  $(n - 1)$  degrees of freedom;  $n$  is the number of words compatible with the rule being scored. The value of  $t_{.05}^{n-1}$  decreases as the degrees of freedom increase.

The  $\pi_L$  of a rule depends on the  $\hat{p}$  estimate of the rule but also on the absolute number of forms compatible with the rule. The smaller this number, the larger the term subtracted from  $\hat{p}$  to compute  $\pi_L$  will be.

Mikheev observes that, if the affix associated with a rule is long, then we should be more confident that the rule is not a coincidental one, even if the number of words compatible with the rule is small. Thus, in order to compute the final score of each rule, Mikheev's procedure divides the error term of  $\pi_L$  by the logarithm of the rule's prefix length (represented by the symbol  $|S|$  in the equation in (22)):

$$(22) \quad \text{rule score} = \hat{p} - \frac{t_{.05}^{n-1} * \sqrt{\frac{\hat{p} (1 - \hat{p})}{n}}}{1 + \log(|S|)}$$

In this formula, the term subtracted from the  $\hat{p}$  decreases as string length increases. Thus, using the formula in (22) to compute the score of each prefixation guessing rule, Mikheev's procedure corrects  $\hat{p}$  by taking both the absolute number of words compatible with a rule and the length of the postulated prefix into account.

Given an input of about 18,000 words from the Cobuild corpus (Baayen, Piepenbrock and van Rijn 1993), the ten prefixation rules with the highest scores found by

Mikheev's algorithm involve the following word-initial strings: *re-*, *ex-*, *self-*, *inter-*, *non-*, *un-*, *dis-*, *anti-*, *de-*, *in-*. These are all real English prefixes. Mikheev also mentions that some of the highest scoring rules involve strings which are not standard English prefixes (such as *st-*). Unfortunately, Mikheev does not provide more detailed data on the performance of the algorithm as a morpheme discovering procedure, since the main goal of the rule scoring component is to select guessing rules for the purpose of unlabeled word tagging.

Mikheev's algorithm appears to follow an effective affix-searching strategy, given a tagged corpus as input. Since Mikheev is not trying to model morphological acquisition, he does not provide hints of how his procedure could be extended to assigning morphological parses to words, besides finding the list of affixes of a language.

## **2.5 Goldsmith *submitted***

Goldsmith proposes a morpheme (stem and suffix) discovery procedure or, rather, a set of related procedures that, like the model presented by Brent 1993 and the one presented here, are based on the MDL principle. Goldsmith's proposal is characterized by a series of MDL-related heuristics which are sometimes applied sequentially, sometimes presented as alternative procedures. Here, I present only a simplified sketch of the basic steps of Goldsmith's algorithm.

Given an untagged corpus of words, the first step of this model is to run a probabilistic algorithm which splits all the words in the corpus into a stem and a suffix. For each word, the algorithm selects the parse with the highest value of the following measure:

$$(23) \quad H(\text{stem/suffix}) = - (|\text{stem}| * \log \text{freq}(\text{stem}) + |\text{suffix}| * \log \text{freq}(\text{suffix}))$$

Where  $|\text{stem}|$  is the length, in letters, of the stem and  $|\text{suffix}|$  is the length, in letters, of the suffix. This measure favors parses in which both the stem and the suffix correspond to strings which are also treated as morphemes in a number of other words, and longer stems and suffixes over shorter ones.

After the first step of the procedure terminates, assigning a morphological parse to each word in the corpus, the stems are organized into mini-paradigms called *signatures*. A signature is a list of all the stems which occur exactly with the same set of suffixes. For example, in one run of the procedure with an English corpus, the stems *despair*, *pity*, *appeal*, *insult* all occurred with the suffixes *-ing* and *-ingly*, and as free-standing words (I use  $\emptyset$  to indicate a zero suffix). Thus, one of the signatures generated by the procedure was:

(24)	despair		$\emptyset$
	pity	+	ing
	appeal		ingly

At this point, all signatures with only one stem or only one suffix (the overwhelming majority) are discarded. In this way, a high number of linguistically insignificant analyses are eliminated -- for example, the one corresponding to the following signature:

(25)		ch
		e
		erial
mat	+	erials
		rimony
		rons
		uring

The suffixes which are found in the remaining signatures, labeled *regular signatures*, are not all linguistically significant, but they do represent a very good approximation to the list of suffixes of the languages analyzed. Moreover, the regular signatures often correspond to linguistically significant mini-paradigms which could be exploited, for example, for purposes of part-of-speech tagging, or as a first step in constructing larger paradigms.

Goldsmith reports convincing results which were obtained with corpora of English, French, Spanish, Latin and Italian words. Thus, Goldsmith's work constitutes strong evidence of the effectiveness of an entirely distribution-based morpheme discovery strategy.<sup>30</sup>

---

<sup>30</sup>Among the most important aspects of Goldsmith's model which I did not review here are his discussion of how the proposed heuristics are related to the MDL criterion, and later steps of the procedure, some implemented, some yet to be implemented, which further improve the performance of the model. In future research, it would be interesting to compare the performance of Goldsmith's model with the performance of the model presented here.

## **2.6 The utterance segmentation model of Brent and Cartwright 1996**

Brent and Cartwright 1996 present an utterance segmentation algorithm which is also based on the MDL principle (Brent and Cartwright refer to the same principle as the Minimum Representation Length principle). The algorithm developed by Brent and Cartwright selects the segmentation of input utterances which minimizes the sum of the length of the lexicon postulated by the segmentation and the length of the derivations of each input utterance encoded using the postulated lexicon.

Consider the following input utterances:

- (26)    **doyouseethekitty**  
          **seethekitty**  
          **doyoulikethekitty**

Several segmentations of these utterances are possible. For example:

- (27)    a.    **d o y o u s e e t h e k i t t y**  
              **s e e t h e k i t t y**  
              **d o y o u l i k e t h e k i t t y**
- b.    **do you see the kitty**  
              **see the kitty**  
              **do you like the kitty**

- c.     do you see thekitty  
        see thekitty  
        do you like thekitty
  
- d.     do yousee thekitty  
        see thekitty  
        do you like thekitty
  
- e.     doyouseethekitty  
        seethekitty  
        doyoulikethekitty

Each of these segmentations can be represented by:

- a *lexicon*, composed of the word types occurring in the segmentation, each paired with an arbitrary, unique index;
- a *derivation*, constructed by replacing each occurrence of a word type in the segmentation with its index.

Brent and Cartwright's algorithm selects the segmentation with the minimum representation length, i.e. the segmentation for which the sum of the length (number of characters) of the lexicon and the length of the derivation is minimal. The five segmentations in (27) have the following representations:

(28) *Segmentation (27.a):*

*Lexicon:*

01 d 02 o 03 y 04 u 05 s 06 e 07 t 08 h 09 k 10 i  
11 l

*Derivation:*

01 02 03 02 04 05 06 06 07 08 06 09 10 07 07 03  
05 06 06 07 08 06 09 10 07 07 03  
01 02 03 02 04 11 10 09 06 07 08 06 09 10 07 07 03

*Segmentation (27.b):*

*Lexicon:*

01 do 02 you 03 see 04 the 05 kitty 06 like

*Derivation:*

01 02 03 04 05  
03 04 05  
01 02 06 04 05

*Segmentation (27.c):*

*Lexicon:*

01 do 02 you 03 see 04 thekitty 05 like



*Derivation:*

01 02 03 04

03 04

01 02 05 04

*Segmentation (27.d):*

*Lexicon:*

01 do 02 yousee 03 thekitty 04 see 05 you 06 like

*Derivation:*

01 02 03

04 03

01 05 06 03

*Segmentation (27.e):*

*Lexicon:*

01 doyouseehekitty 02 seethekitty 03 doyoulikethekitty

*Derivation:*

01

02

03

For each segmentation, we sum the number of characters (letters and digits) in the lexicon and derivation, obtaining the following values:

(29) Segmentation (27.a): 33 (lexicon) + 88 (derivation) = 121

Segmentation (27.b): 32 (lexicon) + 26 (derivation) = 58

Segmentation (27.c): 30 (lexicon) + 20 (derivation) = 50

Segmentation (27.d): 38 (lexicon) + 18 (derivation) = 56

Segmentation (27.e): 50 (lexicon) + 6 (derivation) = 56

The segmentation of the input in (26) for which the sum of the lexicon and derivation lengths is minimal is the one in (27.c) (*do you see thekitty, see thekitty, do you like thekitty*). This is the segmentation selected by Brent and Cartwright's algorithm.

Notice that the minimum representation length criterion disfavors the "extreme" segmentation strategies in (27.e) and (27.a). In (27.e), no segmentation is attempted, and the three utterances in the input are stored as lexical units (words). This strategy minimizes the derivation length, but it is anti-economical in terms of the lexical representation length.<sup>31</sup> At the other extreme, in (27.a) each letter in the input is treated as an independent word and stored as such in the lexicon. This strategy minimizes the lexical representation length,<sup>32</sup> but it is anti-economical in terms of derivation length.

---

<sup>31</sup>This would be easier to see in the case of an input composed of a larger number of utterances.

<sup>32</sup>The advantage of this strategy in terms of lexical representation length becomes more evident as the number of utterances in the input increases.

Since in the input in (26) the strings *the* and *kitty* always occur together, the minimum representation criterion favors the representation in which they are treated as one word (27.c) over the representation in which they are treated as independent words (27.b), since the latter option increases both the lexicon length (requiring an extra index) and the derivation length.

On the other hand, the segmentation (27.d), in which *you* and *see* are stored as a single lexical unit, is not optimal, since (given the utterances *seethekitty* and *doyoulikethekitty*) these strings must also be stored as independent lexical units, making the lexical representation longer. However, the minimum representation length principle does not necessarily disfavor segmentations in which some lexical units are identical to the composition of other lexical units. In the case at hand, if the string *yousee* were very frequent in the input, then the gain in terms of derivational length of representing it as a unit would outweigh the cost of having both *yousee* and the substrings *you* and *see* represented in the lexicon.

If we assume an indexing system such as the one adopted in (28), in which all indices are two digits long, the representation length of a segmentation *S* is given by:

$$(30) \quad RL(S) = 2|TYPES(S)| + \sum_{w \in TYPES(S)} l(w) + 2|TOKENS(S)|$$

Where  $|TYPES(S)|$  is the number of items listed in the lexicon,  $l(w)$  is the length in phonemes (or letters) of lexical item *w* and  $|TOKENS(S)|$  is the total number of occurrences of all the indices in the derivation. The minimum representation length principle favors segmentations with fewer and shorter types and fewer tokens.

Brent and Cartwright use a representational system which is more economical and less biased than the one used in (28). Adopting Brent and Cartwright's representational system, the representation length of a segmentation is given by...

$$(31) \quad RL(S) = 3|TYPES(S)| + (\log_2 P) \left( \sum_{w \in TYPES(S)} l(w) \right) + |TOKENS(S)| * H(S)$$

... where  $P$  is the number of phonemes (or letters) in the input alphabet and  $H(S)$  is the entropy of the relative frequencies of the lexical items in the segmentation:

$$(32) \quad H(S) = - \sum_{w \in TYPES(S)} \frac{f(w)}{|TOKENS(S)|} \log_2 \frac{f(w)}{|TOKENS(S)|}$$

The value computed by the formula in (31), like the value computed by (30), increases with the number and length (in phonemes or letters) of the types in the lexicon, and with the number of tokens in the derivation. Furthermore, the entropy term increases when frequency is evenly distributed across types.

Thus, the minimum representation length criterion, when applied to segmentations whose representation lengths are computed using (31), will favor segmentations which achieve a compromise between the following trends:

- minimize the number of word types in the lexicon;
- minimize the length (in segments or phonemes) of the word types;
- minimize the number of word tokens in the derivation;
- have an uneven frequency distribution of types, in which a small number of words accounts for most of the frequency distribution.

The minimum representation length criterion provides a way to select among lexical analyses, but it does not indicate how these analyses should be generated. In Brent and Cartwright's model, candidate analyses are generated using a "greedy" strategy (for a general introduction to greedy algorithms, see Cormen, Leiserson and Rivest 1990: chapter 17). Their algorithm evaluates a large number of lexical analyses of different generality, ranging from the one in which no input utterance is analyzed to the one in which each letter is treated as a different word. However, of all possible analyses generated by inserting  $n + 1$  boundaries in the input utterances, only those analyses are evaluated in which  $n$  boundaries are inserted exactly where they are in the best (w.r.t. the minimum representation length criterion) analysis generated by inserting  $n$  boundaries. This means that if, say, in the best analysis which was generated by inserting 3 boundaries in the input corpus the utterance *doit* is segmented as *do it*, then, of all the analyses generated by inserting more than 3 boundaries in the input corpus, only those in which the utterance *doit* is segmented as *do it* are considered.

Brent and Cartwright compare the performance of an algorithm based on the minimum representation length criterion with the performance of a "baseline" procedure which inserts the correct number of word boundaries per utterance at random (e.g. the baseline algorithm inserts two boundaries in the utterance *seethekitty*, but the position of the two boundaries is random). The algorithm based on the minimum representation length criterion performs significantly better than the baseline on an input of phonemically transcribed child-directed English with word boundaries removed (the input utterances come from the CHILDES corpus). In the remainder of their paper, Brent and Cartwright show how the performance of their algorithm can be further improved when the minimum representation length strategy is supplemented with phonotactically-driven strategies.

The utterance segmentation heuristics emerging from Brent and Cartwright's application of the MDL / minimum representation length criterion can also be re-interpreted as reasonable morpheme searching heuristics. The morpheme discovery model that I present in the next chapter can be seen as an adaptation of Brent and Cartwright's lexicon selection and generation methods to the morpheme discovery problem.

## Chapter 3

### DDPL: An automated Distribution-Driven Prefix Learner

#### 3.1 Introduction

In order to assess the effectiveness of distributional heuristics in morpheme discovery, I designed and implemented a learning model which performs a particular aspect of this task -- prefix discovery -- on the sole basis of distributional evidence. The algorithm presented here takes a corpus of untagged orthographically or phonetically transcribed words as its input and it outputs a lexicon composed of a list of prefixes and stems. Moreover, the algorithm assigns morphological parses (prefix+stem or monomorphemic parses) to all the word types in the input corpus. The algorithm relies entirely on the distributional information that can be extracted from the input, and it uses a formula based on the MDL criterion (discussed in 3.3.3 below) to select the best lexicon compatible with the input data. From now on, I will refer to the model presented here with the acronym *DDPL*, which stands for *Distribution-Driven Prefix Learner*.

I will first motivate and defend the choice of modeling prefix discovery as an independent subtask within morpheme discovery, then describe the basic idea and the details of the model and, finally, I will briefly discuss to what extent DDPL is a plausible model of how human process distributional information for the sake of morpheme discovery.

### **3.2 Modeling prefix discovery as an independent task**

DDPL takes a list of words as its input, it generates a lexicon composed of prefixes and stems, and assigns maximally binary morphological parses to the word types in the input corpus. In this section, I (try to) answer the following important preliminary questions regarding this model: Is it legitimate to assume that the input to morpheme discovery is a list of words, and not a list of unsegmented utterances? Is it legitimate to assume that the search for prefixes takes place independently of the search for other types of morphemes? Since in real life words can contain more than one prefix, how dangerous is it to simplify the task by assuming that words maximally contain one single prefix? Are there particular reasons to choose to model prefix discovery rather than suffix discovery? The first three questions concern the more general issue of whether it is legitimate to model prefix discovery as an independent task, whereas the last question concerns the issue of whether there are good reasons to model prefix discovery as an independent task. I will dedicate different subsections to these two separate issues.

#### **3.2.1 Is it legitimate to model prefix discovery as an independent task?**

The general answer to the first three questions asked above is based on the following observation: It is much easier and more economical to perform a sequence of simpler



analyses, looking for one kind of constituent at a time, than to look for all possible kinds of morphological (and syntactic) units at the same time.<sup>33</sup>

The fact that the sequential approach makes the task of modeling utterance and morphological segmentation more tractable from a computational point of view does not imply that this approach is simply an expedient dictated by psycholinguistically unmotivated necessities. To the contrary, it is reasonable to assume that children, like computational linguists, would approach the complicated problem of discovering words and morphemes from an angle which makes the problem more tractable.

### **3.2.1.1 Utterance segmentation vs. morphological segmentation**

Let us start by considering the issue of utterance segmentation vs. morpheme discovery. By using a list of words as the input to DDPL, I am implicitly assuming that morpheme discovery should be modeled as a later task independent from utterance segmentation. I believe that this is a reasonable assumption for the following reasons.

First of all, notice that (most) words are easier units to identify than bound morphemes, on the basis of distributional, phonological and semantic cues. Since the

---

<sup>33</sup>The whole discussion in this section is based on the idea that there is a clear-cut distinction between words and bound morphemes/affixes, and between different types of morphemes. Of course, it is sometimes hard to tell whether a certain unit is a word or a bound morpheme, whether a unit is a prefix or the first member of a compound etc. However, I believe that it is reasonable to claim that most units children have to discover can be straightforwardly classified as independent words or bound morphemes, and, if they are morphemes, as morphemes of a specific type.

distribution of words is typically much less restricted than the distribution of bound morphemes, words are easier to identify on purely distributional grounds than bound morphemes. Moreover, words tend to display specific prosodic marks (typically, some form of word-level accent) and their edges are often easily identifiable on the basis of phonotactic evidence (since it is rare for phonotactic constraints to apply across word boundaries, word boundaries can be the locus of otherwise illegal segmental combinations). Furthermore, words in general are associated with more specific and complex semantic representations than bound morphemes.<sup>34</sup>

Thus, on the one hand it is plausible that learners will be able to perform (a significant amount of) the task of word segmentation much earlier than the harder task of morpheme segmentation.<sup>35</sup> Moreover, if the learners were trying to perform the two tasks at the same time, the stronger cues signaling words and word boundaries would be likely to obscure the weaker cues marking bound morphemes and morpheme boundaries.

Furthermore, by performing the task of morpheme discovery on a corpus of words, rather than unsegmented utterances, the learners can make their task a lot easier. In other words, children can greatly simplify the task of morpheme discovery if they first solve the problem of utterance segmentation. For example, if a learner were looking for prefixes in the input sentence *doitagain*, she should consider the hypothesis that each of the following

---

<sup>34</sup>Function words are probably closer to morphemes in terms of their phonological and semantic properties. However, function words tend to occur in a very large number of sentences, surrounded by a very large number of different strings. Thus, function words are easier to discover on the basis of distributional cues than bound morphemes.

<sup>35</sup>Indeed, experimental evidence suggests that seven and a half month old infants are already able to perform word segmentation (Jusczyk and Aslin 1995).

35 substrings is a prefix (any substring followed by at least one segment in the utterance could in principle be a prefix): *d, do, doi, doit, doita, doitag, doitaga, doitagai, o, oi, oit, oita, oitag, oitaga, oitagai, i, it, ita, itag, itaga, itagai, t, ta, tag, taga, tagai, a, ag, aga, agai, g, ga, gai, a, ai*. On the other hand, if the learner were looking for prefixes in the already segmented input *do it again*, she would have to consider only the following 6 potential prefixes: *d, i, a, ag, aga, agai*. It is likely that, in order to decide if a string is a prefix, the learner has to also consider whether what remains of a word when the string is stripped off from it is a plausible stem. Now: the learner faced with the unsegmented utterance *doitagain* has to evaluate 8 candidate stems for the potential prefix *d* (*o, oi, oit, oita, oitag, oitaga, oitagai, oitagain*); on the other hand, a learner considering the segmented input *do it again* will only have to decide whether *o* is a plausible stem for *d*.

The number of possible morphological parses to be evaluated is only one of the many aspects with respect to which performing morpheme discovery on words rather than unsegmented utterances appears to be a much easier task. However, it should be clear to anybody who has pondered similar issues that virtually all the semantic, syntactic, phonotactic and distributional cues relevant to morpheme discovery are also likely to be easier to extract from a list of segmented words than from a list of unsegmented utterances.

While the arguments presented here do not prove that morpheme discovery occurs after utterance segmentation, I believe that, at least, they make a good case for using this as a reasonable null hypothesis, to be modified only in the face of empirical evidence to the contrary.

### 3.2.1.2 Looking for different morphemes in different steps

Similar arguments can be applied to the search for different types of morphemes: it is easier for a learner to first look for one kind of morpheme, then another, then yet another, than to consider all possible combinations of morphemes at the same time.

First of all, in this way the number of possible parses of each word which a learner has to consider at one time (and in total) is drastically reduced. For example, if a learner is looking only for suffixes, she will have to consider only four potential morphological parses of the word *frost* (potential stems are underlined to avoid ambiguities): *f+rost*, *fr+ost*, *fro+st*, *fros+t*. Then, when looking for prefixes, the learner will have to consider four other potential parses of the same word: *f+rost*, *fr+ost*, *fro+st*, *fros+t*. The total number of parses explored during both analyses is eight. On the other hand, if a learner is looking even simply for prefixes *and* suffixes at the same time, the same word will have fourteen possible parses: *f+rost*, *fr+ost*, *fro+st*, *fros+t*, *f+rost*, *fr+ost*, *fro+st*, *fros+t*, *f+r+ost*, *f+ro+st*, *f+ros+t*, *fr+o+st*, *fr+os+t*, *fro+s+t*.

Moreover, what the learner knows about a certain class of morphemes can be helpful in the search for other types of morphemes. For example, a learner who already knows that plural is marked by the suffix *-s* in English, can, in the absence of strong evidence for some form of double marking, disregard the hypothesis that words with plural meaning ending in *-s* contain a plural-marking prefix. Knowing that *-s* is a plural marking suffix could also be useful for more general reasons to a learner looking for prefixes: For example, the learner could avoid counting singular and plural forms of the same noun as different word types for the purpose of a distribution-driven prefix search (e.g., when counting the number of occurrences in different words of a candidate prefix *ro-*, the forms *rose* and *roses* should really be counted as a single word type).

Looking for one kind of morpheme at a time may provide help of an even more general nature. Indeed, it is possible that learners look for rarer types of morphemes, such as prosodic or autosegmental morphemes, only if they notice large or unexpected gaps in the affixing (suffixing/prefixing) morphology of their language.<sup>36</sup>

All the arguments presented here in favor of the idea that learners should look for one kind of morpheme at a time were already suggested by Brent 1993, who phrased them in the following way:

Even when known universal constraints are taken into account, the number of possible hypotheses consistent with a given linguistic input is generally so large that evaluating them all is computationally intractable... One technique that might aid children in the identification of morphemes is search ordering, where the most likely hypotheses are explored first. For example, suffixation appears to be the most common effect of morphological processes in the world's languages, and all languages in Greenberg's survey that have non-affixal morphology also have prefixes, suffixes, or both (Greenberg 1966). Thus, it would make sense for children to look for suffixes and prefixes before looking for metatheses and truncations. Search ordering would clearly speed the identification of

---

<sup>36</sup>Starting with unsegmented sentences, one can imagine a hierarchy of searches ordered by how typologically common a certain unit is: children first look for words, since all languages have words, then, if necessary, for suffixes, then, if necessary, for prefixes, then, if necessary, for discontinuous morphological constituents. It seems plausible that children are innately aware of the existence of such a hierarchy.

suffixes, as compared to searching for all sorts of phonological effects at once, and since suffixes are so common, the average rate of acquisition would be improved too. But looking for suffixes first might actually speed the acquisition of non-affixal morphology as well. The rapid discovery of some suffixes might provide the child with a toe-hold on the language's morphology, making possible partial analysis of the input and thereby simplifying the search for other morphemes. (Brent 1993: 28-29.)

Again, although the arguments presented here do not prove that learners look for different types of morphemes in different steps, they do make a case for this as a reasonable null hypothesis.

Notice that, if the suggestion that learners look for more common morphemes first is correct, then learners should look for suffixes before prefixes, and could use the information gained during suffix discovery in their search for prefixes and prefixed forms. However, here I model prefix discovery assuming that learners do not know anything about suffixation. In this sense, the task faced by DDPL might actually be harder than the task faced by human learners.

### **3.2.1.3 Maximally binary parses**

DDPL evaluates only monomorphemic and binary (prefix+stem) parses of words, and, thus, it systematically misses all but the first prefix of words containing more than one prefix. Evaluating maximally binary parses greatly reduces the hypothesis space DDPL has to explore, and a strategy along these lines could also be helpful to human learners.

Children could first consider only binary parses of words and then, for the set of words that were treated as prefixed in the first pass, strip off the prefix and repeat the search (and then again repeat the search for the set of doubly prefixed words found in this way, etc.). In this way children will reduce the overall number of possible parses to be considered (this is true even in a language in which all the words contain at least one prefix, and of course the savings are really dramatic in languages, such as English, in which the large majority of words are not prefixed). Moreover, the list of prefixes found during the first pass could make the second and later passes considerably easier.

### **3.2.2 Why prefix discovery?**

In the previous section, I defended the view that it is legitimate to model prefix discovery as an independent subtask within the general process of morpheme discovery, which is in turn independent from utterance segmentation.

Now, I will discuss the reason why I decided to concentrate on prefix discovery, as opposed to suffix discovery (the reasons why I did not decide to start by modeling the discovery of non-affixing morphemes should be clear). Given that, for practical reasons, the only language on which DDPL was tested, until now, is English, the argument presented here concerns why it seemed a good idea to concentrate on *English* prefixes vs. *English* suffixes.

My interest in modeling the discovery of English prefixation derives from the fact that this is a domain in which it is unlikely that semantic, syntactic and phonological heuristics play a major role. As I discussed in section 1.2.3, the statistics presented by Schreuder and Baayen on the proportion of pseudo-prefixed forms corresponding to the

most frequent English prefixes suggest that semantic cues cannot be too helpful to learners looking for prefixes. Moreover, since English prefixation is entirely derivational and it does not change the syntactic category of words, syntactic cues can have only a marginal function. Furthermore, while phonological cues may help learners to a certain extent (for example, the vowel of the prefix of nonce formations and transparent forms tends to carry secondary stress), many prefixed words are not marked by special phonological characteristics (and, even if this were the case, it is not clear that learners could notice that certain special phonological features characterize prefixed words until they have performed at least part of the prefix discovery task).

The fact that semantic, syntactic and phonological cues are not likely to play a major role in morpheme discovery makes this a good testing ground for distribution-driven models for two reasons. First, if we find that our model is able to discover actual prefixes and assign correct morphological parses, it is plausible to hypothesize that English learners also relied on distributional cues such as the ones implemented in the model, given that they could not have extracted too much information from other kinds of cues.

Second, if learners do indeed use distributional strategies, English prefixation could be a domain in which these strategies are particularly prominent (given that other forms of evidence are weak), leaving obvious traces in the lexical representations of adult speakers. Thus, it should be easier to test a convergence between learners' intuitions and the output of the computational model in this domain.



### **3.3 The DDPL model**

I turn now to the description and discussion of the prefix learning model which constitutes the core of this study, i.e. the DDPL (Distribution-Driven Prefix Learner) model. Like the suffix discovering algorithm of Brent 1993 and the utterance segmentation model of Brent and Cartwright 1996, DDPL is based on a “generation and selection” strategy: a large number of lexica compatible with the input data are generated, a certain measure is computed for each lexicon, and the lexicon with the lowest value of this measure is selected. The formula used to compute this measure constitutes the conceptual core of the algorithm, and it is based on the idea that the best morphological analysis of the input is also the one allowing maximal data compression of the input, given certain assumptions about how the data compression process should work.

I will start by discussing examples illustrating the connection between the task of data compression and the task of morpheme discovery. In particular, I will present a data compression scheme which, as examples will show, favors the same lexical analyses which would have to be selected on the basis of the morpheme discovering heuristics discussed in 1.4.1 above:

- Substrings which occur in a high number of different words are likely to be morphemes;
- Substrings which tend to occur with other potential morphemes are more likely to be morphemes;
- All else being equal, low frequency words are more likely to be morphologically complex than high frequency words.

The approach presented here, as I will mention, can be interpreted as an implementation of the MDL principle of Rissanen 1978. Finally, I will show how the formula used by the DDPL to select the best lexicon derives from a computation of data compression based on the discussed compression scheme.

In later sections, I will present and discuss the lexicon generation algorithm used by DDPL. For now, I will concentrate on the problem of selecting the best one among a set of candidate lexica, without discussing how these candidate lexica were generated.

### **3.3.1 Data compression and morphological analysis: the shortest lexicon criterion**

The criterion used by DDPL to select the best lexicon is based on the idea that the lexicon generated by the most plausible morphological analysis is also the best lexicon for purposes of data compression, given certain restrictions on how the compressing procedure works. The rationale behind this intuition is the following: since morphemes are syntagmatically independent units (see 1.4.1) which occur in different words and combine with each other, a lexicon containing morphemes is going to be “shorter” (in the literal sense that it can be represented using a small number of characters) than a lexicon containing random substrings, or a lexicon in which no word is decomposed. The advantage of reducing the problem of morpheme discovery to a matter of (constrained) data compression is the following: There are no straightforward ways to decide which one, among a set of possible lexica, is the best one from the point of view of morphology, but it is relatively simple to estimate which lexicon allows maximal data compression.

Given that the connection between data compression and morphological analysis is not very intuitive, I will illustrate it through a simple example. Suppose that we are given the following list of words:

- (33) redo  
do  
remake  
undo  
make  
unmake

Our goal is to find the shortest possible lexicon which could be used to “reconstruct” this list (as I said, the shortest lexicon is, literally, the one which can be written using the smallest number of characters). A word can be reconstructed from the lexicon if the word is listed in the lexicon, so that it can be directly retrieved from it, or if the word can be formed by combining exactly two lexical units. It does not matter if other words, besides the ones in (33), could also be reconstructed by combining lexical units. Given these restrictions, here is a legitimate lexicon from which one could reconstruct the list in (33):

(34) redo  
do  
remake  
undo  
make  
unmake

*length of lexicon: 26*

The lexicon in (34) is identical to the word list. There is no attempt to split words into smaller units. It is easier to design a shorter lexicon. For example, we can exploit the fact that the final substrings *o* and *ke* occur in more than one word:

(35) red  
d  
o  
rema  
ke  
und  
ma  
unma

*length of lexicon: 20*

This lexicon is shorter than the one in (34), since it exploits the fact that the substrings *o* and *ke* occur in more than one word. Thus, instead of rewriting them for each of the words in which they occur, we can write them once as independent lexical units, and reuse them multiple times to reconstruct the words in the input (*rema + ke, ma + ke, unma + ke...*)

This example highlights a first connection between data compression and morpheme discovery: In 1.4.1, I suggested that morphemes are likely to be substrings occurring in a number of different words. The comparison of the lexica in (34) and (35) shows that treating strings which occur in a number of different words as independent lexical units is also a good data compression strategy.

While the lexicon in (35) is based on what could be seen as a plausible morpheme searching strategy, this lexicon does *not* correspond to a plausible morphological analysis of the input word list. However, the following lexicon is shorter than the one in (35) -- it is, probably, the *shortest* lexicon from which one could reconstruct the list in (33)<sup>37</sup> -- and it *does* correspond to a plausible morphological analysis of the input:

---

<sup>37</sup>Here and below, keep in mind that I am assuming that words in the input can maximally be composed of two lexical entries. Without this restriction, the lexicon containing only the list of all phonemes/letters occurring in the corpus would always be the shortest one (or, in case of ties, part of the set of shortest lexica).

(36) re  
un  
do  
make

*length of lexicon: 10*

As we observed in 1.4.1, true morphemes, unlike frequent but arbitrary substrings, have the property that they co-occur with other morphemes. Thus, by splitting a word at a morpheme boundary, we obtain not one, but two syntagmatically independent units, which probably also occur in other words of the input. By decomposing the word *redo* into the morphemes *re-* and *do*, and the word *unmake* into the morphemes *un-* and *make*, we obtain four constituents which can be used to reconstruct not only the two words *redo* and *unmake*, but also *do*, *make*, *undo* and *remake*, all words occurring in the input word list.

We see here another parallelism between data compression and morpheme discovery: In 1.4.1, I suggested that learners should be more willing to treat substrings as morphemes if they tend to co-occur with other potential morphemes. But to split a word into two units both occurring in other words is also a good compression strategy.

Before I move on to consider a different data compression task, which provides a better approximation to the set of morpheme searching heuristics presented in 1.4.1, notice the following, crucial point. As morphologists, we know that (36) derives from a better morphological analysis of the input than (35). However, it is very difficult to devise a way to quantify this intuition, to compute the objective “morphological plausibility” score of a lexicon. Indeed, unless we already know which strings correspond to the morphemes of a language, it is probably impossible to compute such measure. On the other hand, it is

extremely easy to check which of two or more lexica is the most compact one -- it is sufficient to count how many characters are needed to write down each lexicon. Thus, the parallelisms between morpheme searching heuristics and data compression strategies that we started exploring are important because, as long as we can show that our data compression strategy lead us to select morphologically plausible lexica, we found what, as a matter of fact, is an objective and easy-to-compute measure of “morphological goodness”.

### **3.3.2 Data compression and morphological analysis: the shortest lexicon + encoding criterion**

There are several problems with the idea of using the shortest lexicon criterion described in the previous section as a system to look for the best morphological analysis of an input. The most obvious problem is illustrated by the following example. Consider the input list in (37):

(37) dog  
tag  
mug

The most plausible lexicon from which this list could be reconstructed is the following:

(38) dog  
tag  
mug

*length of lexicon: 9*

However, the shortest lexicon from which the list in (37) could be reconstructed is:

(39) do  
ta  
mu  
g

*length of lexicon: 7*

Since word-final *g* occurs in all three input words, it is convenient to store it in the lexicon as a separate unit, rather than writing it three times, once for each word.

This example illustrates the main problem with the shortest lexicon criterion: it is sufficient for a string to occur in a few words -- as little as two words -- to make lexica in which the string is treated as a lexical unit better than lexica in which those words are not decomposed. Clearly, from the point of view of morpheme discovery, this is too lax an interpretation of the “high” type frequency criterion.

One way of dealing with this problem would be to impose a minimum frequency threshold below which substrings cannot be treated as independent lexical units. However, this solution is problematic in at least two respects. First, the threshold would be arbitrary



and it would have to be changed depending on the size of the input corpus (the larger the corpus, the higher the threshold should be). Moreover, more than the absolute number of times a substring occurs in a corpus, what should matter is how many times the string occurs in contexts in which it is preceded/followed by other potential morphemes.

An elegant solution to the problem of excessive decomposition, which does not require arbitrary thresholds, naturally emerges from a reformulation of the goal of data compression that, as we will see, also constitutes a closer approximation to morpheme discovery in other respects.

Let us suppose that we are given a list of words, and our goal is to find a compact format to store information from which the very same list can be reconstructed. In particular, we take a “lexicon and encoding” approach to this task. We construct a compact lexicon from which all the input words (plus, possibly, others) can be reconstructed. We associate an index to each lexical unit, and then we rewrite the corpus as a sequence of these indices. I will refer to the rewriting of the corpus as a sequence of indices with the term *encoding*.

The difference with the task we discussed in the previous section is that in the new scenario we are not simply looking for the shortest lexicon from which the words in the input corpus can be reconstructed, but we are actually trying to find the most compact format in which the corpus itself can be represented. In the lexicon and encoding approach, we try to achieve this by constructing a short lexicon from which the corpus can be reconstructed in an economical way (through the encoding). As we will see, there is typically a trade-off between ways in which the lexicon can be shortened and ways in which the encoding of the input can be shortened. From the point of view of morpheme

discovery, it is this trade-off which will ensure that only decompositions motivated by strong distributional evidence will be performed.<sup>38</sup>

As long as some lexical units occur very frequently in the input corpus (and/or many units occur relatively frequently), and the lexical indices are, on average, shorter than the units they represent, the lexicon + encoding strategy will allow us to represent the corpus in a shorter format than the original one. In order to make sure that the second requirement is satisfied (i.e., lexical indices are on average shorter than the input words they represent), I assume here that all indices are exactly one character long. I will return to this assumption in 3.3.5 and 3.3.7 below.

The following example, which has nothing to do with morphological decomposition, is presented to give a first, general idea of how and why the lexicon and encoding strategy works. Suppose that we are given the following input corpus:

(40) dog  
cat  
dog  
dog  
cat  
cat

---

<sup>38</sup>See 3.3.3 below for a discussion of the relationship of this approach to the MDL principle of Rissanen 1978. As I mentioned, the “shortest lexicon + encoding” approach presented here is closely modeled after Brent and Cartwright’s 1996 minimum representation length criterion.

In order to write this list, we need 18 characters. Following the compression method described above, we can instead write the words *dog* and *cat* only once, assigning a one-character index to each of them (this is the *lexicon* component of the compressed data), and then rewrite the words in the input as a sequence of indices (the encoding):

(41) *lexicon*

dog    1

cat    2

*length of lexicon: 8*

*encoding of (40)*

1       (= dog)

2       (= cat)

1       (= dog)

1       (= dog)

2       (= cat)

2       (= cat)

*length of encoding: 6*

*total length (lexicon + encoding): 14*

To store the word types *dog* and *cat* in a lexicon, and then rewrite the word tokens in the input as a sequence of indices is more economical (in the sense that it requires a smaller number of characters: 14 vs. 18) than writing down the list of input word tokens as it is. Notice that from the lexicon and the list of indices we can reconstruct the original input. Thus, we can store a corpus in the more economical lexicon and encoding format without any loss of information.

The reason why the lexicon and encoding format is more economical than the original list should be clear: even if lexical entries and indices require a certain number of characters, a “good” lexical entry, i.e. an entry corresponding to many tokens in the input, allows major savings in the encoding of the lexicon.

The lexical entry for a three segment word such as *dog* takes four characters (three letters to write the word, one digit for the index). If this word occurs three times in the corpus, the corresponding index will occur three times in the encoding, requiring three characters. Thus, the total number of characters taken by the lexical entry for *dog* and the three occurrences of the corresponding index in the encoded corpus is seven. In the unencoded corpus, three occurrences of a three segment word require nine characters. Thus, we see that it is sufficient for a three segment word to occur three times in the input to justify a lexicon and encoding approach to its representation. Longer and/or more frequent words allow bigger savings.

Now, exactly as in the scenario described above, suppose that we are allowed to decompose input words into two constituents, in order to further compress the data. We assume the following encoding scheme: if an input word is identical to a lexical entry, then the input word is encoded using the index associated with that lexical entry (as in the example above); however, if a word does not have a corresponding entry, and must be reconstructed by concatenating two lexical units, then the word is encoded as the sequence

of the index associated with the first component, a one-character concatenation operator (represented here by the symbol  $\circ$ ) and the index associated with the second component.

For example, suppose that a corpus contains the word *redo*. If this word is listed in the lexicon, for example associated with the index *1*, then the word is represented by a *1* in the encoded input. However, if *redo* is not listed in the lexicon, and it has to be reconstructed from the entries *re*, associated with the index *1*, and *do*, associated with the index *2*, then the word will be represented by the sequence  $1\circ 2$  in the encoded corpus.

While it can be convenient to store frequent substrings in the lexicon, in order to make it shorter, there is going to be a tradeoff between minimizing the length of the lexicon and minimizing the length of the encoding. On the one hand, as in the scenario above, treating substrings which occur in a number of words as independent lexical entries will make the lexicon shorter. On the other hand, since it takes three characters (two indices plus the concatenation operator), instead of one, to encode an input word not listed in the lexicon, any decomposition which makes the lexicon shorter will also make the encoding longer. Thus, only those decompositions which allow a major decrease in lexical length, more than compensating for the corresponding increase in encoding length, are worth performing.

We observed above that morphologically sensible decompositions allow major savings in lexical length because it is likely that both constituents occur in several different words. On the other hand, there is no reason to expect that morphologically arbitrary substrings, although frequent, will tend to combine with other frequent arbitrary strings to form words. Thus, in general, arbitrary substrings will allow less savings in lexical length than complex words, and given that in the shortest lexicon + encoding model each decomposition is penalized in the encoding component, they will be less likely to be treated as independent units than true morphological constituents.

To illustrate this point, let us consider efficient ways to compress the input lists presented in (33) and (37) above, using the lexicon + encoding approach. The list of input words from (33) is re-presented here as (42):

(42) redo  
do  
remake  
undo  
make  
unmake

The shortest lexicon + encoding representation of this list is the following:

(43) *lexicon*

re	1
un	2
do	3
make	4

*length of lexicon: 14*

*encoding of (42)*

1°3    (= re°do)

3       (= do)

1°4    (= re°make)

2°3    (= un°do)

4       (= make)

2°4    (= un°make)

*length of encoding: 14*

*total length (lexicon + encoding): 28*

In particular, this is a shorter representation than the one in which no decomposition is attempted:

(44) *lexicon*

redo	1	
do	2	
remake		3
undo	4	
make	5	
unmake	6	

*length of lexicon: 32*

*encoding of (42)*

- 1      (= redo)
- 2      (= do)
- 3      (= remake)
- 4      (= undo)
- 5      (= make)
- 6      (= unmake)

*length of encoding: 6*

*total length (lexicon + encoding): 38*

The representation in (43), which is based on a plausible morphological decomposition of the input, is ten characters shorter than the representation in (44), where no decomposition is attempted. The reason for this is that the analysis of the input upon which (43) is based provides a very compact lexical component, since the units *re*, *un*, *do* and *make* are all morphemes which occur in at least two input words. Thus, even if the encoding in (43) is longer than the encoding in (44), the lexicon in (43) is so much shorter than the one of (44) that, overall, (43) is the analysis to be selected on the basis of the shortest lexicon + encoding criterion.

However, consider now the case of the input in (45) (presented above as (39)):



(45) dog  
tag  
mug

Recall that the shortest lexicon criterion lead us to choose a lexicon in which *g* is treated as a separate unit. However, if we consider the length of both lexica and encodings, we must choose the lexical analysis of (46), in which no word is decomposed:

(46) *lexicon*

dog 1  
tag 2  
mug 3

*length of lexicon: 12*

*encoding of (45)*

1 (= dog)  
2 (= tag)  
3 (= mug)

*length of encoding: 3*

*total length (lexicon + encoding): 15*

The total length of lexicon and encoding in (46) (where the input words are not decomposed) is shorter than the one of (47), where *g* is treated as an independent lexical unit:

(47) *lexicon*

do 1

ta 2

mu 3

g 4

*length of lexicon: 11*

*encoding of (45)*

1°4 (= do°g)

2°4 (= ta°g)

3°4 (= mu°g)

*length of encoding: 9*

*total length (lexicon + encoding): 20*

While the lexicon in (47) is shorter than the one in (46), the small decrease in lexical length does not justify the increase in encoding length due to the fact that one needs three characters to represent each word not listed as an independent unit in the lexicon.

As the previous examples show, the shortest lexicon + encoding criterion favors the representation of substrings as lexical entries only when this approach leads to considerable savings in lexical length. True morphemes, unlike arbitrary substrings, are more likely to lead to such savings, and, thus, to be treated as independent lexical entries.

Notice how we are now able to avoid excessive decompositions without having to resort to an absolute type frequency threshold. Indeed, the substrings *re-* and *un-* occur only twice in (42). Still, in the shortest representation of this input these substrings are treated as independent units. On the other hand, even though the substring *g* occurs once more (three times) in (45), this substring is not treated as an independent unit in the shortest representation of the relevant input.

### **3.3.3 The shortest lexicon + encoding criterion as an interpretation of the MDL principle**

Before I move on to the illustration of how the shortest lexicon + encoding criterion favors the same analyses that are best from the point of view of distributional morpheme discovering heuristics, I want to make clear that this criterion, which constitutes the basis of the DDPL algorithm, is actually an interpretation of the MDL principle.

The MDL principle (first proposed by Rissanen 1978) has been applied as an inductive learning strategy in many different areas, ranging from computer vision to protein

structure analysis (see the references in Li and Vitányi 1997 ).<sup>39</sup> Within linguistics, the MDL principle has been applied to aspects of phonological structure learning (Ellison 1992), continuous speech segmentation (Brent and Cartwright 1996, de Marcken 1996), morphological segmentation (Brent 1993, Brent, Murthy and Lundberg 1995, Goldsmith *submitted*) and syntactic category acquisition (Cartwright and Brent 1997).

The MDL principle can be stated in the following way (adapted from Li and Vitányi 1997: 5.5): given a sample of data and a set of theories that can account for the data, the best theory is the one that minimizes the sum of the length of the description of the theory and the length of the data when encoded with the help of the theory (where both the length of the theory and the length of the data encoded with the help of the theory are measured in number of units of the description language adopted).

In the lexicon + encoding approach, the sample of data is represented by the list of morphologically unanalyzed input words. The set of theories is a set of lexica, where each lexicon is a list of entries associated with indices. The data are encoded with the help of the lexicon by representing them as a sequence of indices. Both the length of the theory and the length of the data encoded with the help of the theory are measured by counting how many characters (letters, digits, special symbols) are needed to write them.

From here on, I will often directly refer to the shortest lexicon + encoding criterion with the term “MDL principle (criterion)”.

---

<sup>39</sup>The MDL principle can be seen as an information-theoretic interpretation of the Occam’s Razor principle. See Li and Vitányi 1997 for a discussion of this principle in the larger context of the theory of computational complexity. General introductions to the MDL principle are presented in Ballard (1997: 2), Grünwald (1998: 1), Hutchinson (1994: 8.2), Li and Vitányi (1997: 5.5).

### **3.3.4 Data compression and morphological analysis: illustrating the relationship between the lexicon + encoding approach to data compression and distribution-based morpheme discovery heuristics**

In this section, I present a series of related examples which have the function of illustrating how the MDL principle favors lexical analyses that are also optimal from the point of view of the distributional morpheme discovery heuristics that I proposed in 1.4.1 above, and how interesting interactions between these heuristics directly follow from this approach.

#### **3.3.4.1 The high frequency heuristic**

The first example I present shows how the first heuristic is captured by the MDL criterion. Consider the data sample in (48):

- (48)    *disarray*  
          *disdain*  
          *disintegrate*  
          *disadvantage*  
          *disaster*

The analysis in which the word-initial substring *dis* is treated as an independent lexical entry (49.a) allows a more compact lexicon + encoding representation than the analysis in which the words are not decomposed (49.b)

(49) a. *lexicon*

dis	1
array	2
dain	3
integrate	4
advantage	5
aster	6

*length of lexicon: 40*

*encoding of (48)*

1°2 (= dis°array)

1°3 (= dis°dain)

1°4 (= dis°integrate)

1°5 (= dis°advantage)

1°6 (=dis°aster)

*length of encoding: 15*

*total length (lexicon + encoding): 55*

b. *lexicon*

disarray	1
disdain	2
disintegrate	3
disadvantage	4
disaster	5

*length of lexicon: 52*

*encoding of (48)*

1	(= disarray)
2	(= disdain)
3	(= disintegrate)
4	(= disadvantage)
5	(= disaster)

*length of encoding: 5*

*total length (lexicon + encoding): 57*

The analysis in (49.a) is shorter than the analysis in (49.b) (and, hence, it must be selected on the basis of the MDL principle) simply in virtue of the fact that *dis* is a frequent word-

initial string.<sup>40</sup> The lexicon which would be favored by the “frequent strings are morphemes” heuristic is also the lexicon favored by the MDL criterion.

However, notice also that the difference in length between the analyses is rather small (55 vs. 57). Indeed, it is sufficient to remove one word from the list in (48)...

- (50)   disarray  
          disdain  
          disintegrate  
          disadvantage

... and the analysis in which *dis* is treated as an independent entry is no longer shorter than the analysis in which the input words are not decomposed:

- (51)   a.     *lexicon*
- |           |   |
|-----------|---|
| dis       | 1 |
| array     | 2 |
| dain      | 3 |
| integrate | 4 |
| advantage | 5 |

---

<sup>40</sup>Of course, what counts as “frequent” depends in general on the size of the input corpus. In this and the following sections, we analyze very small corpora, and, as a consequence, strings occurring in a small number of words will behave as “frequent”.



*length of lexicon: 35*

encoding of (50)

1°2 (= dis°array)

1°3 (= dis°dain)

1°4 (= dis°integrate)

1°5 (= dis°advantage)

*length of encoding: 12*

*total length (lexicon + encoding): 47*

b. *lexicon*

disarray 1

disdain 2

disintegrate 3

disadvantage 4

*length of lexicon: 43*

*encoding of (50)*

- 1      (= disarray)
- 2      (= disdain)
- 3      (= disintegrate)
- 4      (= disadvantage)

*length of encoding: 4*

*total length (lexicon + encoding): 47*

Exactly as we observed that the “frequent strings are morphemes” heuristic is rather problematic in morpheme discovery, the “store frequent strings as independent units” principle plays a secondary role in data compression: as we will see below, treating substrings co-occurring with other syntagmatically independent substrings as independent entries leads to shorter representations than simply treating frequent substrings as independent units.

When I discussed the “frequent strings are morphemes” heuristic in 1.4.1, I also observed that, in order for this heuristic to be valid, string length has to be controlled for: a longer substring is more likely to be a morpheme than a shorter string of equal frequency. The weight-by-length principle naturally emerges from the shortest lexicon + encoding approach, as the following example illustrates. Consider this input:

(52)    allied  
          amber

asymptotic

arithmetic

adored

In this case, the lexicon in (53.b), where no decomposition is attempted, allows a shorter overall representation than the lexicon in (53.a), where the word-initial string *a* is treated as an independent entry:

(53) a. *lexicon*

a	1
llied	2
mber	3
symptotic	4
rithmetic	5
dored	6

*length of lexicon: 39*

*encoding of (52)*

1°2 (= a°llied)

1°3 (= a°mber)

1°4 (= a°symptotic)

1°5 (= a°rithmetic)

1°6 (=a°dored)

*length of encoding: 15*

*total length (lexicon + encoding): 54*

b. *lexicon*

allied 1

amber 2

asymptotic 3

arithmetic 4

adored 5

*length of lexicon: 42*

*encoding of (52)*

- 1      (= allied)
- 2      (= amber)
- 3      (= asymptotic)
- 4      (= arithmetic)
- 5      (= adored)

*length of encoding: 5*

*total length (lexicon + encoding): 47*

Compare (52)/(53) to (48)/(49) above. There, we saw that it is convenient, for the sake of data compression, to treat a word-initial three letter string (*dis*) as an independent unit, when it occurs in five words in which it is followed by “stems” of length 5 (*array*), 4 (*dain*), 9 (*integrate*), 9 (*advantage*) and 5 (*aster*), respectively. The example in (52)/(53) shows instead that it is not convenient to represent a one letter string as an independent entry even if the string occurs in a corpus which is, in all the other relevant respects, identical to the one in (48) (in (52), the string *a* occurs in five words in which it is followed by “stems” of the same length as the “stems” following *dis* in (48)).

The reason for this asymmetry is the following: the longer a substring is, the larger the savings that its storage as an independent unit will allow. If a three letter string occurs in five words and the string is represented once as an independent lexical unit, instead of repeated in the entry for each of the five words, we save twelve characters. On the other hand, if a one letter string occurs in five words, we save only four characters by

representing it as an independent unit. Thus, all else being equal, in the shortest lexicon + encoding model shorter substrings require higher frequencies in order to be treated as independent entries. But, as we observed, this is also a sensible principle from the point of view of morpheme discovery.

Here, I illustrated the interaction of the effect of substring length with the high frequency principle, but of course the same length effect will also affect the morpheme discovery heuristics / data compression strategies I discuss below (*en passant*: the length effect is also what guarantees that the analysis of (48) in which *dis* is a prefix is shorter than the analyses in which its substrings *di* and *d* are treated as prefixes).

### **3.3.4.2 Co-occurrence with other potential morphemes**

As we observed in 1.4.1, one of the main reasons why high type frequency constitutes a rather poor approximation to morphemic status is that morphemes, unlike arbitrary but frequent strings, occur in combination with other morphemes. Thus, we proposed that a sensible morpheme discovery strategy should not simply be based on absolute frequency, but on the number of times a string tends to co-occur with other “potential morphemes”, i.e. strings which also occur elsewhere in the corpus.

In the lexicon + encoding model, independent lexical entries for strings which tend to combine with other independently occurring strings lead to larger savings than simply treating frequent strings as lexical entries. Consider first the sample in (54):

(54) redo  
go  
remake  
replug  
dogs

Even if the string *re* occurs three times in this corpus, the representation in which this string is not represented as an independent lexical unit (55.b) is shorter than the one in which the words beginning with *re* are decomposed:

(55) a. *lexicon*

re	1
do	2
go	3
make	4
plug	5
dogs	6

*length of lexicon: 24*

*encoding of (54)*

1°2 (= re°do)

3 (= go)

1°4 (= re°make)

1°5 (= re°plug)

6 (dog)

*length of encoding: 11*

*total length (lexicon + encoding): 35*

b. *lexicon*

redo 1

go 2

remake 3

replug 4

dogs 5

*length of lexicon: 27*



*encoding of (54)*

- 1      (= redo)
- 2      (= go)
- 3      (= remake)
- 4      (= replug)
- 5      (= dogs)

*length of encoding: 5*

*total length (lexicon + encoding): 32*

Compare now the input in (54) with the input in (56):

- (56) redo
- do
- remake
- sprint
- make

The two inputs are similar in that they are both composed of six words of length 4, 2, 6, 6 and 4 respectively. Moreover, the two words containing *re* in (56) are also present in (55). However, on the one hand the string *re* only occurs two times in (56) (vs. three times in (54)); on the other hand, in (56) the string *re* occurs before strings which also occur elsewhere in the list (both *do* and *make* also occur as independent words). In this case, the

analysis in which *re* is treated as an independent lexical entry (57.a) is much shorter than the analysis in which the words beginning with *re* are not decomposed (57.b):

(57) a. *lexicon*

re 1

do 2

make 3

sprint 4

*length of lexicon: 18*

*encoding of (56)*

1°2 (= re°do)

2 (= do)

1°3 (= re°make)

4 (= sprint)

3 (= make)

*length of encoding: 9*

*total length (lexicon + encoding): 27*

**b.     *lexicon***

**redo    1**

**do       2**

**remake       3**

**sprint  4**

**make   5**

***length of lexicon: 27***

***encoding of (56)***

**1       (= redo)**

**2       (= do)**

**3       (= remake)**

**4       (= sprint)**

**5       (= make)**

***length of encoding: 5***

***total length (lexicon + encoding): 32***

The examples from (54) to (57) show how, in the MDL model, strings which co-occur with other strings that also independently occur in the corpus are more likely to be treated as independent units than strings which simply frequently occur in the corpus, even if the

latter are more frequent, in absolute terms, than the former.<sup>41</sup> The reason for this preference is that, if both elements forming a complex word are already stored in the lexicon, then there is no need to store any extra item in the lexicon in order to be able to reconstruct the word. On the other hand, if only one of the constituents of a word is already stored in the lexicon, we still need to create a lexical entry, complete with an index, for the remainder of the form.

As we observed in 1.4.1, looking for substrings which tend to co-occur with other independently occurring substring is also a plausible morpheme discovery strategy.

### **3.3.4.3 Word frequency and morphological complexity**

We observed that, all else being equal, learners should be more willing to treat words as morphologically complex if they rarely occur in the corpus than if they are frequent. Again, there is a parallel with the shortest lexicon + encoding approach to data compression, as the following examples show. Consider first the input in (58):

---

<sup>41</sup>To keep things simple, I presented here an example in which stems occur elsewhere in the corpus as independent words -- i.e., they are free stems. However, the same pattern takes place even if the relevant stems never occur in independent words, but are the product of the parse of other prefixed forms -- i.e., they are bound stems.

(58) disarray  
array  
disobey  
obey

Of course, the shortest analysis of this input is the one in which both *disarray* and *disobey* are decomposed (59.a):<sup>42</sup>

(59) a. *lexicon*

dis 1  
array 2  
obey 3

*length of lexicon: 15*

*encoding of (58)*

1°2 (= dis°array)  
2 (= array)  
1°3 (= dis°obey)  
3 (= obey)

---

<sup>42</sup>Both analyses presented here are shorter than the one in which *disobey* is not decomposed.

*length of encoding: 8*

*total length (lexicon + encoding): 23*

b. *lexicon*

disarray      1

array          2

dis            3

obey          4

*length of lexicon: 24*

*encoding of (58)*

1      (= disarray)

2      (= array)

3°4    (= dis°obey)

4      (= obey)

*length of encoding: 6*

*total length (lexicon + encoding): 30*

However, consider now the input in (60), which is identical to (58), except that I added four more tokens of the word *disarray*.

(60)    *disarray*  
         *array*  
         *disobey*  
         *obey*  
         *disarray*  
         *disarray*  
         *disarray*  
         *disarray*

Now, the best analysis becomes the one in which *disarray* is *not* decomposed into *dis* and *array* (61.b):<sup>43</sup>

(61)    a.    *lexicon*

*dis*    1  
         *array* 2  
         *obey* 3

*length of lexicon: 15*

---

<sup>43</sup>Both analyses are shorter than the one in which neither *disarray* nor *disobey* are decomposed.

*encoding of (60)*

1°2 (= dis°array)

2 (= array)

1°3 (= dis°obey)

3 (= obey)

1°2 (= dis°array)

1°2 (= dis°array)

1°2 (= dis°array)

1°2 (= dis°array)

*length of encoding: 20*

*total length (lexicon + encoding): 35*

b. *lexicon*

disarray 1

array 2

dis 3

obey 4

*length of lexicon: 24*



*encoding of (60)*

1      (= disarray)  
2      (= array)  
3°4    (= dis°obey)  
4      (= obey)  
1      (= disarray)  
1      (= disarray)  
1      (= disarray)  
1      (= disarray)

*length of encoding: 10*

*total length (lexicon + encoding): 34*

These examples show how, all else being equal, in the lexicon + encoding model more frequent words are more likely to be stored in the lexicon in non-decomposed format than less frequent words. The only difference between the distribution of *disarray* in (58) and (60) is that in (60) this word occurs five times, whereas in (58) it occurs only once. Because of this difference in frequency, *disarray* get its own lexical entry in the shortest analysis of (60), whereas in the shortest analysis of (58) it must be reconstructed from the components *dis* and *array*.

The reason why this model favors independent storage of frequent words, even when both of their components are also specified in the lexicon, is the following: Given that each occurrence of a decomposed word in the encoded corpus requires three indices

instead of one, if a word occurs frequently in the corpus, it is more convenient to use some characters to build a lexical entry for it, in order to have an economical way to encode it. On the other hand, if a word is rare, it is more convenient to save the characters required to represent the word in the lexicon, and encode the word in a costly way the few times in which it occurs in the corpus.

The example (60)/(61.b) also illustrates an interesting general property of the DDPL model: Not only is this model flexible enough to allow words to be treated as morphologically simple (i.e., represented in the lexicon as independent units), even if they begin with a substring which is specified as a prefix in the lexicon (in this case, the string *dis* is a prefix in the lexicon of (61.b), but the word *disarray* is represented in the lexicon in non-decomposed format), but words can be represented as units in the lexicon even if *both* their component parts are also lexical entries: in the case at hand, both the word *disarray* and its constituents *dis* and *array* are listed in the lexicon.

As it is a common feature of many recent models of lexical-morphological processing (see Schreuder and Baayen 1995 and the other models reviewed there) to assume that words can have an independent lexical representation even if they could be entirely derived from morphemic constituents also stored in the lexicon, I believe that it is a desirable property of our learning model that it is allowed to select lexica in which this situation arises.

#### **3.3.4.4 Type vs. token frequency**

The fact that, as we showed in the previous section, the MDL criterion disfavors lexica in which frequent words are not represented as whole units, also has an important

consequence for the high frequency heuristic. Intuitively, this heuristic should be based on *type* and not *token* frequency. If a substring occurs at the beginning of, say, 100 words, it is more likely that this substring is a prefix than if the substring occurs at the beginning of a single word repeated 100 times in the input.

Indeed, since in the lexica selected using the MDL criterion frequent words tend to be stored as independent units, substrings occurring in a number of different low frequency words are more likely to be treated as independent lexical entries than substrings which occur in a limited number of high frequency words (this can also be seen as an approximation to the *hapax legomena* heuristic suggested by Baayen's work on productivity).

For example, recall that the best analysis of the input in (48) above (re-presented here as (62.a)) was the one in (49.a) (re-presented here as (62.b)):

(62) a.     *input*

disarray

disdain

disintegrate

disadvantage

disaster

b. *lexicon*

dis	1
array	2
dain	3
integrate	4
advantage	5
aster	6

*length of lexicon: 40*

*encoding of (62.a)*

- 1°2 (= dis°array)
- 1°3 (= dis°dain)
- 1°4 (= dis°integrate)
- 1°5 (= dis°advantage)
- 1°6 (=dis°aster)

*length of encoding: 15*

*total length (lexicon + encoding): 55*

In this corpus, the word-initial string *dis* occurs in five different words, each occurring only once. In the shortest analysis of the corpus, *dis* is represented as an independent lexical unit. Consider now the following input:

(63)    *disarray*  
         *disarray*  
         *disarray*  
         *disarray*  
         *disintegrate*

Even though the string *dis* occurs five times in this input as well, the analysis of (63) in which *disarray* and *disintegrate* are not decomposed (64.b) is shorter than the analysis in which *dis* is treated as an independent entry (64.a):

(64)    a.     *lexicon*

<i>dis</i>	1
<i>array</i>	2
<i>integrate</i>	3

*length of lexicon: 20*

*encoding of (63)*

$1^{\circ}2$  (= dis°array)

$1^{\circ}2$  (= dis°array)

$1^{\circ}2$  (= dis°array)

$1^{\circ}2$  (= dis°array)

$1^{\circ}3$  (= dis°integrate)

*length of encoding: 15*

*total length (lexicon + encoding): 35*

b. *lexicon*

disarray      1

disintegrate    2

*length of lexicon: 22*

*encoding of (63)*

1      (= disarray)  
1      (= disarray)  
1      (= disarray)  
1      (= disarray)  
2      (= disintegrate)

*length of encoding: 5*

*total length (lexicon + encoding): 27*

All else being equal, in the MDL model type frequency is more important than token frequency. The string *dis* has the same token frequency (five occurrences) in (62.a) and (63). However, in the first input it also has a (relatively) high type frequency -- it occurs in five different words -- whereas in the second input its type frequency is only two. In the shortest analysis of the corpus in which *dis* has a high type frequency, the string is represented as an independent entry, whereas in the shortest analysis of the corpus in which the string has only a high token frequency, the words containing it are not decomposed.

### 3.3.4.5 An example of constraint interaction in the lexicon + encoding model

In the previous sections, we saw how the shortest lexicon + encoding (MDL) criterion favors analyses which are also sensible from the point of view of the distribution-driven morpheme discovery heuristics that we discussed in chapter 1. Of course, sometimes what is good from the point of view of one heuristic is not good from the point of view of another heuristic.

For example, given a word such as, say, *resist* in an English corpus, the high frequency heuristic could favor a parse such as *r+esist*, since *r* is an extremely frequent word-initial substring. On the other hand, the heuristic favoring splits in which both components also occur elsewhere will probably favor the parse *re+sist*. Finally, given that *resist* is a relatively frequent word, the heuristic favoring independent storage of frequent words probably favors the parse in which this word is not divided.

In the MDL model, which solution wins in cases of conflict among heuristics depends on the amount of savings that each solution allows given the characteristics of a particular input. The way in which the heuristics interact can be rather complex. As an example of how contrasting data compression (and morpheme searching) schemes can lead to interesting lexicon selection patterns, consider the following set of examples.

First, recall from (60)/(61) above that in the shortest analysis of an input such as (64.a), the word *disarray* is represented as an independent lexical unit (64.b):



(64) a. *input*

*disarray*

*array*

*disobey*

*obey*

*disarray*

*disarray*

*disarray*

*disarray*

b. *lexicon*

*disarray*      1

*array*          2

*dis*             3

*obey*          4

*length of lexicon: 24*

*encoding of (64.a)*

1      (= disarray)  
2      (= array)  
3°4    (= dis°obey)  
4      (= obey)  
1      (= disarray)  
1      (= disarray)  
1      (= disarray)  
1      (= disarray)

*length of encoding: 10*

*total length (lexicon + encoding): 34*

The representation in (64.b) respects the principle requiring frequent words to be stored as independent units in the lexicon, but it is marked in that *disarray* is not decomposed although both *dis* and *array* also occur elsewhere in the corpus (and are represented as independent units in the lexicon).

The reader can verify that, if we were to remove even just one occurrence of *disarray* from the input in (64.a), then the best analysis of the input would become the one in which this word is decomposed into *dis* and *array*. Thus, *disarray* must occur at least five times in this kind of input to gain its own independent lexical entry. Consider now the following list:

(65) disaster  
disobey  
obey

In the shortest analysis of this input, the word *disaster* is decomposed into *dis* and *aster*  
(66.a):<sup>44</sup>

(66) a. *lexicon*

dis 1  
aster 2  
obey 3

*length of lexicon: 15*

*encoding of (65)*

1°2  
1°3  
3

*length of encoding: 7*

---

<sup>44</sup>Both analyses reported here are shorter than the ones in which disobey is not decomposed. This is also true for (68).

*total length (lexicon + encoding): 22*

b. *lexicon*

disaster	1
dis	2
obey	3

*length of lexicon: 18*

*encoding of (65)*

1  
2°3  
3

*length of encoding: 5*

*total length (lexicon + encoding): 23*

Here, it is convenient to decompose *disaster* into *dis* and *aster* because the unit *dis* “comes for free” from the analysis of *disobey* and *obey*. However, this solution is somewhat marked, in that it requires us to build a lexical entry *aster* even if this substring occurs only once in the corpus.

Now, interestingly, it is sufficient to add *one* more instance of *disaster* to the corpus in (65)...

(67) disaster  
disobey  
obey  
disaster

... and the analysis in which *disaster* is *not* decomposed (68.b) becomes the shortest one:

(68) a. *lexicon*

dis 1  
aster 2  
obey 3

*length of lexicon: 15*

*encoding of (67)*

1°2  
1°3  
3  
1°2

*length of encoding: 10*

*total length (lexicon + encoding): 25*

b. *lexicon*

disaster      1

dis            2

obey          3

*length of lexicon: 18*

*encoding of (67)*

1

2°3

3

1

*length of encoding: 6*

*total length (lexicon + encoding): 24*

Both (64) and (67)/(68) illustrate the effect of the “store frequent words as independent entries” constraint on eight character words beginning with *dis*. However, the non-

decomposed representation of *disarray* in (64) violates the constraint favoring the decomposition of words that can be entirely parsed into constituents independently occurring in the input (*dis* and *array*). No such counterforce is active in (67), where the potential stem of *disaster* (*aster*) does not occur elsewhere.

As a result of the tension between two contrasting forces, *disarray* in (64) is more resistant against the “store frequent words as independent entries” constraint than *disaster* in (67). In the case of *disarray*, it is necessary to include at least five occurrences of this word in the input to make the analysis in which the word is not decomposed the shortest one. On the other hand, in the case of *disaster*, it is sufficient for this word to occur twice, and the analysis in which the word is not decomposed becomes the shortest one.

### **3.3.5 Prefixes, stems and prefix-stem asymmetries in the lexicon + encoding model**

In the previous sections, we saw how the MDL criterion provides an objective, easy-to-compute measure to select a lexicon which is also optimal from the point of view of morpheme discovery heuristics. Since our ultimate goal is to model morpheme discovery, we are not interested in what would be, in principle, the best way to maximally compress a data set. Rather, we are interested in a compression scheme which is constrained and biased towards morphologically sensible analyses.

In section 3.3.7, I review some of the differences between the shortest lexicon + encoding criterion implemented in the DDPL model and more efficient and realistic compression schemes. First, however, I will introduce some changes to the lexicon + encoding model as it was presented above. These changes are not justified from the point

of view of data compression; instead, they have the function of making the model sensitive to the fact that prefixes and stems are different types of units, with different distributional properties.

First of all, notice that in the lexica considered above there is no distinction between prefixes and stems. For example, this was the lexicon of (57.a), the shortest analysis of (56):

(69)    *re*        1  
         *do*        2  
         *make*    3  
         *sprint* 4

Nothing in this representation tells us that *re* is different from the other entries. While this is not a problem from the point of view of data compression, the ultimate purpose of DDPL is to model prefix discovery, and discovering which lexical entries are prefixes and which entries are stems is of course one of the basic aspects of this task (at the very least, learners should know which entries are prefixes and which entries are stems/words in order to be able to parse any new complex word they learn).

Thus, in the actual DDPL model, each lexical entry is marked by a one-character diacritic indicating whether the unit is a prefix or a stem.<sup>45</sup> Let us assume that the diacritic

---

<sup>45</sup>In the current version of the DDPL system, bound and free stems do not have a different status. However, even if we assigned a different one-character diacritic to bound stems in order to distinguish them from free stems, the formula in (76) below would not have to be modified.



marking prefixes is the symbol +, and the diacritic marking stems/words is the symbol #. Then, the lexicon in (69) can be rewritten as:

(70)	re+	1
	do#	2
	make#	3
	sprint#	4

This is an improvement, but it is not enough. Prefixes and stems have different distributional properties, and this should be reflected in the model.<sup>46</sup> In particular, the formula used to select the best lexicon should take into account the fact that affixes tend to be more frequent units than stems. Given a prefixed input word, it is very likely that its prefix also occurs in a number of other input words, whereas the stem probably only occurs in very few other words. For example, it is plausible that in a corpus of English containing the form *reconsider*, the prefix *re* also occurs in hundreds of other words, whereas the stem *consider* only occurs in this prefixed form and as an independent word.

In order to take the prefix-stem asymmetry into account, I introduce the following bias against prefix entries. I have assumed until now that both prefix and stem entries are associated with indices that are one character long. However, in the final version of the DDPL model prefix entries are associated with indices that are 1.25 characters long. The

---

<sup>46</sup>Part of the duties of the lexicon generation algorithm I discuss in 3.3.9 below is to ensure that only word-initial substrings are counted as occurrences of prefixes (e.g. that *re* as an independent word and *re* in *more* are not treated as instances of the prefix *re-*) and that only word-final stems and independent words are counted as stems.

particular amount of extra-length assigned to prefix indices (.25) in the current model was empirically determined, by running the algorithm with the same input and different values for this parameter.<sup>47</sup>

While this final modification is not very straightforward to represent, suppose that we use the symbol ' to mark "one fourth of character" (= .25). Then, the lexicon in (70) has to be rewritten as:

(71)	re+	1'
	do#	2
	make#	3
	sprint#	4

Besides making the lexical representation of a prefix one fourth longer than the representation of a stem of the same orthographic or phonetic length, this approach further penalizes prefixes because each occurrence of an index corresponding to a prefix in the

---

<sup>47</sup>In a series of simulations, I ran DDPL with the input described in chapter 4, but changing the value of the prefix index length parameter (from a minimum length of 1 to a maximum of 2, with .05 intervals). What emerged from these experiments was that the value of this parameter can range from 1.15 to 1.55 without creating major differences in the output generated by the model (the same prefixes are found, and the parses assigned to input words are almost identical). If the prefix index parameter is assigned a value higher than 1.55, DDPL finds only a subset of the prefixes found in the 1.15-1.55 range. More interestingly, if the prefix index parameter is assigned a value lower than 1.15, DDPL does not find more actual English prefixes, but just more "false positives" (strings that are not real English prefixes but are treated as prefixes by DDPL).

encoded corpus will be one fourth longer than the representation of a stem or a morphologically simple word.

The effect of this representational bias is that prefixes have to be more frequent (and/or more frequently co-occur with potential morphemes) than stems in order to be represented as independent lexical units in the shortest analysis of the input.

An important consequence of the bias against infrequent prefixes we just introduced is that it will in general disfavor analyses in which stems of suffixed forms are mistakenly treated as prefixes. Consider for example the word *lovely*. Given that, in a reasonably sized English corpus, both *love* and *ly* probably also occur in other forms, there is the risk that DDPL could treat *love* as a prefix and *ly* as a bound stem. However, compared to a real prefix, *love-* is likely to occur in a very limited number of forms: for example, in the PHLEX database the word-initial substring *love* only occurs in a total of 8 forms, whereas even a rare prefix such as *para-* occurs in 22 words. Thus, given the “anti-infrequent-prefix” bias, it is unlikely that in the shortest analysis of an input a string such as *love* will be actually treated as a prefix.

### **3.3.6 A note on boundaries**

For obvious reasons of clarity, in the previous sections I presented each lexicon + encoding pair exploiting the organization of lines and blank spaces in the page as a way to make the examples easier to read. However, a lexicon + encoding pair could be stored as a continuous string of symbols without any loss of information, since all the necessary boundaries can be recovered from this continuous string. Thus, there is no need to compute the length of

the formatting and of blank spaces, when calculating the overall length of a lexicon + encoding pair, as formatting and blank spaces do not carry any extra information.

Consider the following example. Given the input in (72.a) the analysis presented in (72.b,c) could also be presented as a continuous string, as in (72.d) (assuming, as before, that *l'* stands for a *single*, one-and-one-fourth characters long index):

(72) a. *input*

redo

do

make

make

remake

b. *lexicon*

re+            *l'*

do#            2

make#        3

c. *encoding of (71.a)*

1'°2 (= re°do)

2 (= do)

3 (= make)

3 (= make)

1'°3 (= re°make)

d. *lexicon + encoding*

re+1'do#2make#31'°22331'°3

Given (72.d), we can reconstruct the boundaries in the lexical component since we know that every time we encounter the symbols + or # an entry is over, the next symbol will be the index associated with that entry (recall that we are assuming that even prefix indices are single, although longer, symbols), and the next next symbol is the beginning of a new entry. Assuming (as we have, implicitly, until now) that there is no overlap between the alphabet used to represent the lexical entries and the alphabet used to represent the indices, then the end of lexicon / beginning of encoding boundary is marked by a sequence of two index symbols. In the part of the string corresponding to the encoding, we can reconstruct the boundary between input words in this way: whenever we encounter an index corresponding to a stem, we are at the end of a word; whenever we encounter an index corresponding to a prefix, we know that the next two symbols are the concatenation operator and the index corresponding to the stem of the same word.

### **3.3.7 Why DDPL is not a realistic data compression scheme**

In various respects, the lexicon + encoding model described above is *not* realistic and/or *not* economical as a data compression scheme.

First of all, I assumed that all indices associated with stems are one character long and that all indices associated with prefixes are one-and-one-fourth characters long. While justified by morphological considerations, the anti-prefix bias is of course absolutely unmotivated from the point of view of data compression, as it makes the representation of prefixes in the lexicon and encoded corpus less succinct. Moreover, the idea of “one fourth of a character” is rather abstract, and it is not clear to me how it could be actually implemented in a concrete representational system. Finally, in order to be able to assign a different one-character index to each stem in a reasonably sized corpus, we would need an alphabet of thousands of characters. This is not a very plausible scenario.

The standard approach to encoding (as described by Roman 1996, among others), of course, does not make use of a huge alphabet. Instead, indices of different length (but mostly longer than one character) are assigned to different entries on the basis of their frequency of occurrence in the corpus. The length of an index is inversely proportional to the frequency of the corresponding entry: frequent entries are assigned shorter indices; rare entries are assigned longer indices.

This approach to encoding is commonly adopted in applications of the MDL principle to linguistic problems (see, for example, Brent and Cartwright 1996). Indeed, an earlier version of DDPL (Baroni 2000) was based on this standard, frequency-dependent approach to index assignment. However, I later decided to opt for the alternative approach presented above for the following reasons.

First, frequency-dependent index assignment schemes favor lexical analyses in which few entries are extremely frequent and many entries are extremely rare. It is not clear to me that this corresponds to a sensible morpheme discovery heuristic.

Moreover, it would be harder to introduce an anti-prefix bias, such as the one I proposed in the previous section, if indices, independently of whether they represent prefixes or stems, had different lengths depending on their frequency.

Finally, as I will discuss below, the DDPL algorithm is based on a formula which allows us to estimate what would be the length of a lexicon plus the corresponding encoding of the input without having to perform the actual data compression procedure. While such a formula can also be derived for the model in which indices depend on frequency, in the latter model the formula is likely to require the computation of the entropy of the relative frequencies of the lexical indices in the corpus. This term is extremely inefficient to compute, and I am not aware of any efficient way to estimate it.

The DDPL lexicon + encoding scheme also differs from more plausible data compression strategies in that some of the information present in the proposed representations is redundant, and thus it should be removed to obtain maximal data compression. In particular, the concatenation mark is not necessary. Even in an encoding scheme that does not employ this symbol, whenever we encounter an index corresponding to a prefix we would know that we are dealing with a complex form, and that the next index represents the stem of the same prefixed word. Moreover, if the decoders of the corpus can tell one-and-one-fourth characters long indices from one-character indices (and assuming that there is no overlap between the alphabet used to represent entries and the alphabet used to represent indices), then even the diacritics marking prefixes and stems in the corpus become redundant, as the indices are already sufficient to mark the boundaries between entries and to signal which entries are prefixes and which entries are stems.

To conclude, in this section, I pointed out several aspects in which the lexicon + encoding model as presented above is not realistic / efficient as a data compression scheme. However, these aspects should not be seen as “problems” of the model, since, as we already remarked many times, the real testing ground for DDPL is not data compression, but morpheme discovery.

### **3.3.8 Computing the length of lexicon + encoding pairs: the DDPL lexicon selection formula**

One way to evaluate different lexica using the shortest lexicon + encoding criterion would be to actually construct alternative lexicon + encoding pairs, and check which one is the shortest. However, DDPL uses a different, more efficient approach, in which the length of a lexicon and the corresponding encoding is calculated without actually having to go through the steps of diacritic and index assignment and input encoding.

The data structures generated and compared by DDPL constitute analyses of the input that are different from the lexicon + encoding pairs considered above, but from which we can easily reconstruct how long lexicon + encoding pairs containing the same kind of information would be.

The data structures generated by DDPL contain a list of prefixes, a list of stems (which can also be independent, non-prefixed words) and, for each prefix and stem, the number of occurrences of that prefix or stem in the input corpus. Consider for example the following input:



(73) redo  
redo  
do  
remake  
remake  
make  
make

The lexicon + encoding representation of this input in which *re* is treated as a prefix is the following:

(74) *lexicon*

re+ 1'

do# 2

make# 3

*length of lexicon:* 14.25

*encoding of (73)*

1'°2    (= re°do)

1'°2    (= re°do)

2        (= do)

1'°3    (= re°make)

1'°3    (= re°make)

3        (= make)

3        (= make)

*length of encoding: 16*

*total length (lexicon + encoding): 30.25*

The corresponding DDPL data structure is:

(75) *prefix list*

re        4

*stem list*

do        3

make     4

In (75), the number following each lexical entry is not an index, but the number of times that the entry occurs in the corpus. Given (75), we can calculate the length of the same analysis as if it were represented in the lexicon + encoding format.

Notice that there are two aspects of (74) (the corresponding lexicon + encoding representation) which could not be reconstructed from the DDPL data structure in (75): the order in which indices occur in the input corpus, and the specific symbols used to represent indices. For example, from (75) we cannot tell that the first index in the encoding in (74) is *l'*, which corresponds to the prefix *re-* (however, we can tell that the index corresponding to *re-* occurs 4 times in the encoded input). Moreover, we cannot tell that the index assigned to *re-* is *l'*, and not, say, *7'* (however, we can tell that *re-*, being a prefix, is associated with an index that is one-and-one-fourth characters long).

From our perspective, this loss of information is not problematic, since neither the order in which indices occur in the encoded input nor the particular symbols representing indices play a role in the computation of the lexicon + encoding length: If the index corresponding to *re-* occurs four times in the encoded input, then it will contribute a total of 5 characters (1.25 times 4) to the encoding length, independently of where it occurs. If *re-* is a prefix, then its index will be one-and-one-fourth characters long, independently of the specific symbol used to represent it.

In other words, while there is a one-to-many relationship between DDPL data structures and lexicon + encoding representations, all the lexicon + encoding representations corresponding to a single DDPL data structure have the same length (they differ in the specific alphabet symbols assigned to indices and/or in the order of indices in the corpus, but these properties do not affect their total length). Thus, while a DDPL data structure does not contain all the information present in a lexicon + encoding representation, it does contain all the information necessary to calculate the length common

to all the corresponding lexicon + encoding representations. From our point of view, the alternative lexicon + encoding pairs corresponding to each DDPL data structure are equivalent, since they have the same length. Thus, I will from now on assume that there is only one such pair.

Also, from now on, I will refer to the total (lexicon + encoding) length of a lexicon + encoding pair with the term *description length*. The idea is that a lexicon + encoding pair constitutes a *description* of the input. This term is obviously related to the name of the MDL principle.

Let us see how the length of the corresponding lexicon + encoding description can be computed from a DDPL data structure, such as the one in (75) above.

For each entry in the prefix list of a DDPL data structure, the same entry also occurs in the lexicon of the corresponding lexicon + encoding pair, where it is followed by the one-character prefix-marking diacritic, and associated with a one-and-one-fourth characters long index. Thus, the contribution of each entry in the prefix list to the description length of the corresponding lexicon is given by the sum of length of the entry itself plus 2.25 (one character for the diacritic plus 1.25 characters for the index). For example, the entry *re* in (75) contributes 4.25 ( $= 2 + 2.25$ ) characters to the length of the lexical component.

Similarly, the contribution of each entry in the stem list to the description length of the corresponding lexicon is given by the sum of the length of the entry itself plus 2 (one character for the stem-marking diacritic and one character for the associated index).

Thus, the total length of the lexicon component corresponding to a DDPL data structure will be given by the length (in characters) of all the entries in both the prefix and the stem lists, plus 2.25 times the number of entries in the prefix list (for the prefix diacritics and indices), and 2 times the number of entries in the stem list (for the stem diacritics and indices).

In order to compute the length of the encoding component, notice that there are three classes of symbols occurring in it: 1.25-character-long prefix indices, one-character concatenation symbols and one-character stem indices.

Starting from the latter: Since each stem index is one character long, the total length contributed by all occurrences of all stem indices in the encoded corpus equals the total number of occurrences of all stems in the corpus. Thus, the total length contributed by stem indices can simply be computed by summing the number of occurrences in the input of each entry listed in the stem list (recall that the number following each entry in a DDPL data structure is the number of occurrences of that entry in the input). For example, the total length contributed by the stem indices in the encoding corresponding to (75) is given by 7 (3 occurrences of *do* plus 4 occurrences of *make*).

Since prefix indices are one-and-one-fourth characters long, the total length contributed by all occurrences of all prefix indices in the encoded corpus equals the total number of occurrences of all prefixes in the input times 1.25. Thus, the total length contributed by prefix indices is computed by summing the number of occurrences of each entry listed in the prefix list and multiplying this number by 1.25. For example, the total length contributed by prefix indices in the encoding corresponding to (75) is 5 (4 occurrences of *re* times 1.25).

Finally, observe that the concatenation symbol is always preceded by a prefix index. Thus, the total number of occurrences of the concatenation mark is the same as the total number of occurrences of all prefixes in the corpus. Since the concatenation symbol is one character long, the total length contributed by all concatenation marks equals the number of occurrences of prefixes in the corpus. For example, the length contributed by concatenation marks in the encoding corresponding to (75) is 4 (the only prefix in the prefix list, *re*, has four occurrences in the corpus).

Thus, the total length of the encoding component corresponding to a DDPL data structure will be given by the total number of occurrences in the input of all the stems in the stem list, plus 2.25 times the total number of occurrences in the input of all the prefixes in the prefix list (this term derives by summing the terms used to compute the total length of prefix indices and the total length of concatenation marks).

Putting the pieces together, the description length of a lexicon + encoding pair corresponding to a DDPL data structure can be computed using the following formula:

(76) *the description length formula*

$$dl = \sum_{ent \in entries} \text{length}(ent) + 2|stem\_entries| + 2.25|prefix\_entries| + |stem\_occurrences| + 2.25|prefix\_occurrences|$$

<i>dl</i> :	description length;
<i>ent</i> $\in$ <i>entries</i> :	any entry, prefix or stem, in the DDPL data structure;
<i>length(ent)</i> :	length in characters of an entry;
<i> stem_entries </i> :	total number of entries in the stem list;
<i> prefix_entries </i> :	total number of entries in the prefix list;
<i> stem_occurrences </i> :	total number of occurrences of all stem entries in the input;
<i> prefix_occurrences </i> :	total number of occurrences of all prefix entries in the input;

For example, if we calculate each of the terms of the formula in (76) for the data structure in (75), we obtain:

$$(77) \quad dl = (2 + 2 + 4) + 2*2 + 2.25*1 + 7 + 2.25*4 = 8 + 4 + 2.25 + 7 + 9 = 30.25$$

The description length computed from the DDPL data structure in (75) using the formula in (76) equals the length as calculated by actually representing the same data in the lexicon + encoding format, as in (73).

Given that the formula in (76) computes the length of a certain analysis as if it were represented in the lexicon + encoding format, the very same analyses that are selected by the shortest lexicon + encoding criterion are also selected when using (76) to determine the shortest description length. For this reason, I will not discuss how various patterns are favored/disfavored by (76).

In brief, the need to minimize the first three terms of (76) will favor distributionally motivated decompositions (as such decompositions will reduce the number of entries in the prefix and stem lists, and they will reduce the lengths of lexical entries), whereas the last two terms will disfavor decompositions (after a decomposition, some words in the input will have to be encoded with three indices instead of one, and thus the total number of occurrences of stems or prefixes in the encoding will increase). The extra weights added to the two prefix-specific factors will insure that, all else being equal, less distributional evidence is needed to postulate a stem than to postulate a prefix.

Henceforth, I will use the term *lexicon* to refer to DDPL data structures. In ambiguous cases, I will refer to these structures as *DDPL lexica*.

### 3.3.9 Searching for the best lexicon: the lexicon generation algorithm

We can use the formula in (76) to determine which, among alternative lexica, has the shortest description length, and should thus be selected as the best lexicon accounting for the input data. However, we still need a way to *find* the best lexicon, or more precisely, generate the set of alternative lexica from which to select the one allowing the shortest description length.

The brute force solution in which all possible combinations of all possible bi- or monomorphemic parses of each word in the input are evaluated is not feasible, as the number of possible combinations grows exponentially with the number of words in the corpus.<sup>48</sup> Therefore, it is necessary to substitute it with a search procedure that is likely to find the best solution, or something close to it. I will refer to this procedure as a “lexicon generation algorithm”. The DDPL lexicon generation algorithm is based on a greedy strategy inspired by the one proposed by Brent and Cartwright 1996.<sup>49</sup>

The DDPL starts by computing the description length of the lexicon in which no morphological analysis of the input is computed. Then, all possible lexica generated by

---

<sup>48</sup>This approach is not feasible even if, as in the DDPL model, we assume that tokens of the same word are always parsed in the same way. In this case, the number of analyses to be considered grows exponentially with the number of word types in the corpus.

<sup>49</sup>The algorithm can be classified as greedy since, at each stage, only lexica derived from the best lexicon of the previous stage are evaluated. For a general introduction to greedy algorithms, see Cormen, Leiserson and Rivest (1990: chapter 17). Cormen, Leiserson and Rivest discuss several examples in which it can be proven that a greedy algorithm will be able to find the best solution. This is not the case here. In future research, it would be interesting to design algorithms that explore a larger section of the hypothesis space.



splitting one single word (type) into morphemes are evaluated. The description length of these lexica is computed, and the lexicon with the shortest description length is selected.

Now, of all the lexica which could be generated by splitting two word (types) into morphemes, only those lexica are evaluated in which one of the two morphological splits is the same as the one performed to generate the shortest lexicon created by one morphological split. Then, the same procedure is repeated for lexica generated by splitting three words, and so on. Generalizing, of all lexica which could be generated by  $n$  morphological splits, only those in which  $n-1$  splits are identical to the ones performed to generate the shortest lexicon created by  $n-1$  morphological splits are evaluated.

Since we are only considering binary parses of words, we add the further condition that, if a stem/word occurs in a morphological split, then no further decomposition of that stem/word will be attempted.

All of this will probably become clearer by considering an example. Consider the following input:

(78) do  
redo  
cat  
redo  
do  
do

The first lexicon generated by the DDPL is simply listing word types, with no morphological analysis:

(79) *prefix list*

*(empty)*

*stem list*

do      3          redo    2          cat      1

*dl: 21*

Now, we consider all the possible lexica generated by decomposing one single word type of the input into one prefix and a stem (notice that only splits in which the prefix is followed by a word-final string are allowed; e.g. the lexicon in which the input word *do* is treated as a prefix in its entirety is not considered):

(80) a. *prefix list*

d      3

*stem list*

o      3          redo    2          cat      1

*dl: 30*

b. *prefix list*

r      2

*stem list*

do 3 edo 2 cat 1

*dl: 27.75*

c. *prefix list*

re 2

*stem list*

do 5 cat 1

*dl: 23.75*

d. *prefix list*

red 2

*stem list*

do 3 o 2 cat 1

*dl: 27.75*

e. *prefix list*

c 1

*stem list*

do      3      redo    2      at      1

*dl: 26*

f.      *prefix list*

ca      1

*stem list*

do      3      redo    2      t      1

*dl: 26*

Now, the crucial step. The shortest among the lexica in (80) is (80.c), i.e., the lexicon which was generated by dividing the word *redo* into *re* and *do*. Thus, of all the lexica which could be generated by dividing two words into morphemes, only those will be considered in which *redo* is one of the two words, and it is split into *re* plus *do*. This means that no lexicon in which *redo* is *not* split will be considered, and that no lexicon in which *redo* is parsed otherwise (as *r+edo* or *red+o*) will be considered.

Thus, only the following two, among the lexica which could be generated by two morphological splits, are evaluated:

(81)      a.      *prefix list*

re      2

c      1

*stem list*

do      5          at          1

*dl: 28.25*

b.      *prefix list*

re      2

ca      1

*stem list*

do      5          t          1

*dl: 28.25*

Further analysis of *do* is not allowed, or else the input word *redo* would have to be parsed into more than two units, violating one of the basic assumptions of the DDPL model. Thus, the algorithm stops. Among all the lexica considered, the one in (79) (no morphological analysis) is the one with the shortest description length, and thus the one that DDPL will select as its output.

The basic procedure I just illustrated guarantees that the number of lexica to be evaluated does not grow exponentially with the number of word types in the corpus. Moreover, other constraints are introduced in the model to further reduce the number of lexica that has to be evaluated.

First of all, only morphological splits which create a stem and/or a prefix that is already in the shortest lexicon of the previous level are considered. For example, of all the lexica in (80), the only one that DDPL would actually consider is the one in (80.c), because it is based on a split which produces the stem *do*, and this stem is already present in the lexicon of (79). None of the lexica in (81) would be considered by DDPL.<sup>50</sup>

Moreover, the following constraint dramatically reduces the number of lexica evaluated by the DDPL, while at the same time further reducing the risk that stem + affix parses are mistakenly treated as prefix + suffix parses. In very general term, the constraint can be phrased in the following way: Only word-initial strings that frequently occur in the corpus before relatively long word-final strings can be treated as prefixes in a morphological split.

Before I go on to describe how this is actually implemented in DDPL, I will quickly mention the rationale behind the constraint: Prefixes are always followed by stems; possibly, by stems and suffixes. Thus, it seems reasonable to expect that prefixes will occur in a certain number of words in which they are followed by relatively long word-final strings. Consequently, we can trim the search space by requiring, as I said, that only word-initial strings that “frequently” occur before “relatively long” word-final strings can be treated as prefixes in a morphological split.

---

<sup>50</sup>This constraint has the following side effect: if a language had a set of bound stems which only occur in combination with a set of prefixes all of which, in turn, never combine with free stems, DDPL is going to be able to find neither the bound stems nor the prefixes combining with them. Notice that it is sufficient for one of the prefixes in such a set to combine with at least one free stem, then all the other prefixes and bound stems in the set can also be discovered by DDPL.

In order to implement the constraint, we have to specify what we mean for “frequently” and what we mean for “relatively long”. Starting from the latter: In DDPL, a word-initial string is a potential prefix only if it frequently occurs before word-final strings of length  $l$  or longer, where  $l$  is the length of the average word in the corpus. This constraint could be problematic for prefixes occurring only before bound stems. However, I believe that it is reasonable to expect that, even for a prefix which only occurs before bound stems, some of these stems (or stem + suffix combinations) will be as long as independent words.

Now, I have to make the notion of “frequently occurring” explicit. Notice that what counts as frequent, rather than being constant, should depend on the length of the word-initial string. That a single segment word-initial string occurs before, say, one hundred word-final strings of length  $l$  is less significant than if a four-segment word-initial string occurs before the same number of word-final strings of length  $l$ .

Thus,  $f$ , the minimum number of times that a word-initial string of length  $k$  is required to occur before a word-final string of length  $l$  or longer is given by the following formula:

$$(82) \quad f = \lceil e^{2(4-k)} \rceil + 1$$

The formula in (82) requires a very high frequency of occurrence for one-segment strings, drastically lower frequencies for longer strings:

(83)  $k$   $f$

1 2982

2 56

3 9

4 4

> 4 3

While the constraint I just described is based on a plausible heuristic and makes the lexicon generation procedure a lot more efficient, by drastically reducing the number of possible morphological analyses to be evaluated, notice that the exact formula used to compute  $f$  in (82) is rather arbitrary, and it would have to be changed depending on the size of the input corpus.<sup>51</sup>

Finally, after the best lexicon is selected, the DDPL algorithm goes over the list of prefixes postulated in this lexicon and, if a prefix only occurs in a single word of the input corpus, the prefix is removed from the list, and the corresponding word is re-introduced in the list of stems.

---

<sup>51</sup>Given the constraints I just described, DDPL halts when one of these events takes place: 1) each word in the input word type list has either been split, or analyzed as the stem of another word; 2) no further morphological split generates a prefix and/or stem identical to an already existing prefix or stem; 2) no further morphological split generates a prefix occurring at least  $f$  times before strings of length  $l$  (or longer) in the input. Notice that the lexicon selected by DDPL is not necessarily the last lexicon evaluated, but the lexicon with the shortest description length, among all the lexica that were evaluated during the lexicon generation procedure.



To summarize: In the actual DDPL model, the procedure illustrated by the examples (78)-(81) above is further constrained in the following ways: first, only morphological splits producing either a stem or a prefix that is already in the lexicon are allowed; second, only those morphological splits are allowed in which the prefix is a word-initial string that occurs at least  $f$  times before strings of length  $l$  (or longer) in the input. Moreover, prefixes with a type frequency of one are trimmed from the final lexicon constituting the DDPL output.

### 3.3.10 The DDPL model: summary

At the core of the DDPL algorithm is the formula in (76), repeated here as (84):

$$(84) \quad dl = \sum_{ent \in entries} \text{length}(ent) + 2|stem\_entries| + 2.25|prefix\_entries| + \\ + |stem\_occurrences| + 2.25|prefix\_occurrences|$$

The analysis of the input (represented as a DDPL lexicon) which minimizes this formula is selected by DDPL as the most plausible one. The formula in (84) computes the length of the same analysis if it was represented in a specific data compression format, i.e., the lexicon + encoding format. As I argued, the optimal (most compact) lexicon + encoding analysis is also the best analysis from the point of view of distribution-driven morpheme discovery heuristics such as the ones I discussed in chapter 1.

While the connection between data compression and morpheme discovery (based on the observation that the property of syntagmatic independence is crucial to both data

compression and distributional morpheme learning) constitutes the conceptual core of the DDPL model, we need a way to generate a set of alternative analyses to be evaluated and compared using the formula in (84).

The specific algorithm implemented in DDPL is based on a greedy strategy similar to the one proposed by Brent and Cartwright 1996 for the sentence segmentation task. The basic idea adapted from Brent and Cartwright is that a lexicon constructed with  $n$  morphological splits is only evaluated if, of those  $n$  splits,  $n-1$  are identical to the ones that were used to generate the shortest lexicon constructed with  $n-1$  splits.

However, the DDPL lexicon generation strategy is further constrained. In particular, the number of lexica to be evaluated is strongly reduced by the requirement that only word-initial strings that frequently occur in the input before relatively long word-final strings can be treated as prefixes in a DDPL lexicon.

### **3.4 DDPL as a model of human morpheme discovery**

My ultimate purpose in designing the computational model presented in this chapter is, of course, to learn something about how human beings discover the morphemes of their language. By showing that DDPL can extract a fair amount of information about English prefixation from distributional cues, we can prove that, in principle, human beings could successfully use distributional cues in morpheme discovery. Moreover, as we discussed in the introduction and in chapter 1, if we can show that human beings assign morphological parses similar to the ones assigned by DDPL to words where their intuition is not likely to depend on semantics or other non-distributional factors, then we can make a case for the claim that humans rely on distributional cues such as the one used by DDPL.

Notice that these arguments are based on the nature of the evidence used by DDPL (distribution, frequency and length of words and substrings), and on the output generated by the model (does DDPL discover actual English prefixes? do the DDPL parses match speakers' intuitions?) However, I am making no claim about DDPL as a model of how humans process distributional information.

The purpose of DDPL is to show that it is in principle possible to generate a certain output (prefixes, stems, morphological parses) given a certain type of input (distributional cues); and I would like to argue that humans, like the DDPL algorithm, are indeed relying on the same kind of input information in order to come up with the same kind of output. The specific way in which the input data are actually manipulated by DDPL vs. by humans, in order to generate the output, could be radically different without invalidating my arguments.

Indeed, there are several aspects of the DDPL model that are probably implausible from a psychological point of view, the most obvious one being that DDPL takes a fixed list of words as its input, and analyzes them all together (and multiple times) in order to try to determine what are the morphemes of the language under consideration. While it is not implausible that even human learners use a certain set of words as their "benchmark" to perform morpheme discovery, it is not plausible that this set is determined at the beginning of morphological acquisition, and never augmented by new words the learners acquire. Moreover, it is not clear that the greedy algorithm used to generate candidate lexica in DDPL could have any plausible psychological counterpart.

On the other hand, if learners *do* rely on distributional heuristics such as the ones used by the DDPL, it is not completely implausible that they would use a strategy based on something like the lexicon + encoding scheme in order to implement these heuristics.

The most radical view along these lines would be that learners use the shortest lexicon + encoding criterion because they are indeed performing a task which is very similar to data compression. Suppose that, on the one hand, learners constructing their lexicon are trying to minimize the amount of information that they have to memorize, and that one of the functions of morphological analysis is to economize on lexical storage. On the other hand, suppose that there is an extra cost associated with the processing needed to reconstruct words from morphemes during lexical access (as opposed to directly retrieving whole words from the lexicon), and that learners are also trying to minimize this cost. In particular, since learners do not know how frequently they will need to retrieve a certain word in the future, suppose that they estimate the processing cost associated with retrieving a word in decomposed format on the basis of how often they heard that word in the past. It is not hard to imagine that, if learners are trying to find a balance between the need to minimize lexical storage and the need to minimize the amount of processing associated with the retrieval of words in a decomposed format, they could come up with a measure related to the shortest lexicon + encoding criterion.

An alternative theory which could explain why learners rely on something similar to the lexicon + encoding scheme, without committing us to the claim that learners are trying to optimize their lexical representations in the way I just suggested, is the following: The task of the learner is to discover which one, among the many lexica compatible with it, was the one that actually generated the input, i.e. the lexicon shared by the adult speakers surrounding the learner. Now, in the absence of better heuristics, the learner could assume that the lexicon of adults is optimal, in the sense that it allows a good trade-off between economy of storage and efficiency of retrieval. Thus, they use the shortest lexicon + encoding criterion not because they actually want to optimize their lexicon, but because they assume that this is what adults did.

One way to express the difference between the two views is this: In one scenario, the Occam Razor is a tool which children actually use to construct an optimal lexicon, whereas in the other scenario the Occam Razor is a theory that children developed about how the adult mind works.

These are very abstract speculations. As I remarked at the beginning of this section, the DDPL simulation can provide us with useful insights on human morphological acquisition independently of whether the shortest lexicon + encoding criterion, or other aspects of the model, correspond to ways in which human beings handle distributional data.

### **3.5 Summary**

In this chapter, I first explained why I believe that modeling prefix discovery as an independent stage of morpheme discovery is both interesting and legitimate. Then, I presented DDPL, an algorithm which performs prefix discovery taking a list of words as its input and relying on distributional evidence only. I have shown that the formula used by this model to select the best morphological analysis of the input favors the same kind of analysis that would be favored by the distributional heuristics discussed in chapter 1. Finally, I briefly presented some speculations on how certain aspects of the DDPL model could have a psychologically plausible counterpart as human learning strategies.

## **Chapter 4**

### **The DDPL model and the discovery of English prefixes**

#### **4.1 Introduction**

In this chapter, I discuss the results of a simulation in which DDPL was tested with a corpus of untagged orthographically transcribed English words from the PHLEX database as its input. After discussing the characteristics of this corpus and why I chose it, I will present the results of the simulation. First, I analyze the list of prefixes discovered by DDPL. Then, I present the results of two surveys in which the morphological parses assigned by DDPL were compared to morphological complexity ratings assigned by native English speakers to the same words. In particular, in the second survey DDPL and English speakers' morphological parses of a set of semantically opaque words were compared. Finally, I briefly discuss another simulation in which DDPL was run with a corpus of phonetically transcribed words as its input.

#### **4.2 The input corpus**

The PHLEX database (Seitz, Bernstein, Auer and MacEachern 1998) contains, among other word lists, a list of the 20,000 most common word types in the Brown corpus (Kucera and Francis 1967), in orthographic and phonetic transcription, together with their frequency of occurrence in the Brown corpus. I removed from this list all word-types

containing non-alphabetic symbols (digits and diacritics such as -).<sup>52</sup> After this trimming, a set composed of 18,460 word types was left. The input corpus for the DDPL simulation was generated by multiplying each of the types in this set by its frequency in the Brown corpus, and randomizing the resulting list.<sup>53</sup> For example, the word *kindergarten* has a frequency of 3 in the Brown corpus, and thus it occurs three times in the DDPL input used for the current simulation. In total, the corpus generated in this way contains 959,655 orthographically transcribed word tokens.

The primary reason why I decided to use the Brown list from the PHLEX database as the input to DDPL is that, when this project started, this was the first corpus that became available to me. However, I consider the fact that PHLEX only contains the 20,000 (18,460 after trimming) most frequent word types in the Brown corpus as a positive feature, for current purposes. Concretely, this means that the each word in the PHLEX-Brown list has a minimum token frequency of 3. Thus, the corpus I used does not contain *hapax legomena* or words with a token frequency of 2. Given that we are trying to model an aspect of language acquisition, this seems to be a plausible exclusion, as in general language learners are not likely to be exposed to such forms.<sup>54</sup>

---

<sup>52</sup>Notice that, while the exclusion of words containing - makes it less likely for DDPL to treat compounds as prefixed words, at the same time the same exclusion makes its task harder, because many productively formed prefixed words (such as *re-elected*, *re-enter*, *re-examine*) are transcribed with - in the original list, and were thus discarded from the DDPL input.

<sup>53</sup>Randomization has a purely esthetic function, as the order of tokens in the input is irrelevant to DDPL.

<sup>54</sup>Notice also that, by excluding words with a frequency lower than 3, we are probably making the task of DDPL more challenging, given that, as shown in the work of Baayen and collaborators (see, for instance,

One could argue that, from this point of view, it would have been more interesting to test the DDPL model using a corpus of child-directed speech. Besides the fact that I am not aware of the existence of any large corpus of English child-directed speech that would have been practical to use for our goal, I believe that English prefix discovery does not take place very early in morphological acquisition, but late enough that we would have to consider child-directed speech from an age in which the discrepancy between the adult and child-directed lexicon is not very large. Thus, I believe that the PHLEX-Brown corpus is not as bad an approximation to the type of input children are probably using to perform prefix discovery as it would be to the input to what are likely to be earlier acquisition tasks (e.g., sentence segmentation).

The reason why I believe that prefix discovery does not take place very early in English language acquisition is that English prefixation is entirely derivational and, in general, not very productive. As a consequence of this, prefixed forms are not very frequent, and it seems to me that children would simply lack enough evidence (distributional or of some other nature) to perform prefix discovery if they pursued this task early on, when their lexicon is very small, and, one would expect, mostly composed of high frequency words.

Consider for example the case of *re-*, which is probably the most productive and definitely the most common English prefix. The most frequent prefixed word with *re-* in the PHLEX-Brown corpus is *review*, a relatively cultivated word which, with a token frequency of 56, cannot be classified as a high frequency form (there are 1939 words with



higher frequency of occurrence in the same corpus).<sup>55</sup> And, of course, before children become familiar with at least *a few* forms containing a prefix, they will probably not notice its presence. Thus, English learners must develop a relatively large and sophisticated vocabulary in order to be able to discover even the most common of their prefixes.<sup>56</sup>

I decided to use orthographic transcriptions instead of phonetic transcriptions for several reasons, besides the fact that it is obviously easier to work with a corpus of orthographically transcribed words.

First, notice that some English prefixes display special prosodic properties, reflected in their segmental makeup, when they occur in prefixed nonce formations and transparent forms. For example, the vowel of the prefix *re-* is likely to be produced as /i:/ in a transparent form such *redo* but as /ʊ/ or /ə/ in an opaque form such as *resume*. Given that DDPL has no access to prosodic/phonological knowledge (for example, DDPL cannot tell that the distribution of stress in English determines alternations such as /i:/ vs. /ʊ/ or /i:/ vs. /ə/), the model has no way to tell that the /ri:/ of *redo* and the /rʊ/ of *resume* should be counted as instances of the same string. On the other hand, vowel reduction is such a basic

---

Baayen 1994, Baayen and Lieber 1991), productive affixes are characterized by their frequent occurrence in *hapax legomena*.

<sup>55</sup>Compare *re-* with an inflectional suffix such as *-ing*. In the PHLEX-Brown corpus, there are 55 words containing this suffix (counting only uncontroversially suffixed words) that have a higher frequency than *review* (the highest frequency form containing *-ing* being *being*, with a token frequency of 711).

<sup>56</sup>Notice that, while I am not attempting to model this pattern in DDPL, it is very likely that learners discover different prefixes in different stages. For example, English learners probably discover prefixes such as *re-* and *de-* much earlier than “fancy” prefixes such as *para-* or *meta-* (if they do discover such prefixes at all).

pattern of English that it seems reasonable to assume that children performing prefix discovery would already be aware of it.<sup>57</sup>

Moreover, many entries in the PHLEX-Brown database have multiple phonetic transcriptions. One common source of multiple transcriptions is the following: given that the PHLEX entries are untagged and not lemmatized, verbs and nouns that are only distinguished by their stress pattern (and by the effects of the stress pattern on their segments) have a single orthographic entry and frequency value, but two phonological transcriptions: one corresponding to the verb, one corresponding to the noun. For example, the PHLEX entry for the orthographic word *rebels* (fq = 17) has two transcriptions, corresponding to its interpretations as a plural noun ('*rebəlz*) or third person singular verb (*ri'belz*). In other cases, multiple transcriptions are due to the fact that the PHLEX transcribers report more than one pronunciation of a certain word. For example, the word *respiratory* (fq = 17) is transcribed as '*respəratɔri* and *res'paɪratɔri*.

In order to deal with multiple transcriptions, we have to either arbitrarily select one of the transcriptions of each ambiguous word, or divide the frequency of the corresponding orthographic entry among the alternative transcriptions. Both strategies would affect the

---

<sup>57</sup>In other respects as well, orthographic transcriptions make the task of finding prefixes and assigning morphological parses to words harder than phonetic transcriptions. For example, in orthographic transcription the *un-* (/ʌn/) of *undo* (which is a prefix) and the *un-* (/jun/) of *unit* (not a prefix) are *not* distinguished. Thus, given an orthographic input, a learner that discovered that *un-* is a prefix would have to decide whether *unit* is a prefixed word or not, whereas this would not be an issue for a learner that, given phonetic input, discovered that /ʌn/ is a prefix.

distribution of strings in the corpus in ways that would be hard to control and probably not motivated.<sup>58</sup>

Notice also that, as I am arguing that the discovery of English prefixes must occur relatively late, it is not completely unlikely that, at least for some prefixes, it actually takes place when children already learned to read. Thus, I would not completely exclude the hypothesis that the input to prefix discovery, for children literate in English, is in part constituted by written words.

Finally, given that the parses assigned by DDPL to input words will be compared to morphological complexity ratings assigned to the same words by English speakers, and these speakers were presented with orthographic transcriptions of the words, to the extent that there could be some discrepancy between morphological representations of spoken and written words, it made sense to test the DDPL on a written input.

While the main focus of this chapter is on a simulation in which the input to DDPL is composed of orthographically transcribed forms, I will briefly discuss in section 4.6 the results of a run in which the DDPL was tested with the same words in phonetic transcription.

---

<sup>58</sup>Observe also that the Brown corpus is a corpus of written English.

### 4.3 Assessing the performance of the distribution-driven model, part 1: prefixes postulated by DDPL

Given the input described in the previous section, DDPL generated an output lexicon containing the following 29 prefixes:<sup>59</sup>

(85) *prefixes postulated by DDPL*

ad-	auto-	co-	com-	con-	cor-	de-	dis-	ex-
extra-	juris-	in-	inter-	man-	mis-	non-	over-	para-
pre-	psycho-	radio-	re-	sub-	sup-	super-	sur-	tele-
un-	under-							

A first inspection of this list shows that DDPL was quite successful and accurate in finding (almost) only actual English prefixes.<sup>60</sup> In the next two subsections, I discuss the “false

---

<sup>59</sup>Notice that *com-*, *con-* and *cor-* are actually allomorphs of the same prefix, and so are *sub-* and *sup-* (cf. *suppress*). The purpose of the DDPL model is simply to find the list of strings that correspond to prefixes (and stems) of a language. The model does not attempt to group strings that are allomorphs of the same morpheme into the same entry. I do not think that allomorph grouping is a task which should be performed on the sole basis of distributional cues, as semantic and phonological cues would, obviously, be of great help.

<sup>60</sup>Notice, among other things, that the DDPL model was able to find prefixes that constitute substrings of other prefixes: *co-* is a substring of *com-/con-/cor-*; *ex-* is a substring of *extra-*; *in-* is a substring of *inter-*; *un-* is a substring of *under-*. This property (together with the fact that not all words containing a word-

positives” in the list (i.e., strings that were treated as prefixes by DDPL but are not actual English prefixes) and the “misses” (i.e., English prefixes that were not found by DDPL).

#### 4.3.1 False positives

The list contains only one obvious false positive: the string *man-*. As *man-* is a noun which often occurs as the first member of compounds, and DDPL does not have access to (morpho-)syntactic information, the model mistakenly treated forms such as *manservant* and *manslaughter* as prefixed.

Besides *man-*, there are three ambiguous cases: The strings *juris-*, *radio-* and *psycho-* are not classified as prefixes in standard references such as Marchand 1969 or Quirk, Greenbaum and Svartvik 1985.<sup>61</sup> Thus, we should probably count them as false positives. However, as these strings correspond to bound word-initial units associated with specific semantic features, even if they might not be prefixes under some definition of what a prefix is, they are “prefix-like” enough that I am reluctant to classify them as real false positives.

The DDPL did not treat any frequent but linguistically insignificant word-initial string (strings such as *pa-*, *pr-*...) as a prefix.

It seems legitimate, I think, to conclude that the DDPL was very accurate in its identification of English prefixes.

---

initial string identical to one of the postulated prefixes are treated as prefixed) constitutes, I believe, a major improvement with respect to Brent’s 1993 model.

<sup>61</sup>The Merriam-Webster’s dictionary classifies *radio-* and *psycho-* as “combining forms.”

### 4.3.2 Misses

The following 19 prefixes listed in Quirk *et al.* 1985 were not discovered by DDPL (notice that *il-*, *im-* and *ir-* are allomorphs of *in-*, a prefix that *was* found by DDPL):<sup>62</sup>

(86) *prefixes missed by DDPL*

a-	an-	anti-	arch-	contra-	counter-	fore-
hyper-	il-	im-	ir-	mal-	mini-	out-
post-	pro-	pseudo-	trans-	ultra-		

Of this set, the following 6 misses are due to the nature of the input corpus, which did not contain enough forms to motivate their treatment as prefixes:

(87) a-          an-          arch-          hyper-          mini-          pseudo-

In particular, the string *hyper-* never occurs in the corpus, whereas the other five strings in (87) only occur in two or fewer words in which they function as prefixes (even counting completely semantically opaque, highly lexicalized prefixed forms).

---

<sup>62</sup>The neo-classical prefixes (such as *hemi-* and *paleo-*) and conversion prefixes (such as *en-* and *be-*) listed in Quirk *et al.* 1985 but missed by DDPL are not reported in this list. I believe that misses from these classes are not problematic, as they concern prefixes that are very cultivated and/or not very productive.

Of the remaining 13 misses, the following 9 are due to the constraint on lexicon generation presented in 3.3.9, which requires that a word-initial string of length  $k$  occurs at least  $f$  times before a word-final string of length  $l$  or longer,<sup>63</sup> in order to be a candidate prefix:

(88) counter-      fore-   il-      im-      ir-      mal-   out-      post-   ultra-

None of these prefixes occurs frequently enough before word-final strings of length  $l$  in the input to justify their treatment as prefixes.

This leaves us with 4 unexplained misses:

(89) anti-      contra-      pro-      trans-

Inspection of the input suggests that the corpus does contain several truly prefixed forms displaying these prefixes in combination with independently occurring stems (but notice that, in the input used in this simulation, the prefixes *contra-* and *pro-* only occur in lexicalized formations with bound stems, such as *contraception* and *proceed*). Thus, these misses cannot be attributed to the nature of the input. I plan to explore the issue of why DDPL failed to discover these prefixes in future research.

---

<sup>63</sup>In this specific input,  $l = 7.5$ .

### **4.3.3 Prefixes found and missed by DDPL: concluding remarks**

The list of prefixes found by DDPL is more accurate than exhaustive. On the one hand, the false positives in the list are few and linguistically motivated. On the other hand, even if the prefixes in (87) are excluded from the count, as they hardly occur in the input, we must remark that DDPL missed 13 productive English prefixes.

In particular, most of these misses are due to the constraint on lexicon generation requiring word-initial strings to occur a certain number of times before “long” word-final strings. This constraint plays an important role in the current DDPL model, as it greatly reduces the number of lexica DDPL has to evaluate, and it helps (together with the anti-prefix bias) avoiding the parsing of stem+suffix combinations as prefix+stem sequences. It is likely that the misses that are not due to the mentioned constraint are also a product of the lexicon generation algorithm, rather than of the lexicon selection method.

These considerations suggest that the first step in future revisions of DDPL should concentrate on the lexicon generation component of the model.

## **4.4 Assessing the performance of the distribution-driven model, part 2: morphological parses assigned by DDPL compared to complexity ratings assigned by English speakers**

Besides finding a list of prefixes, DDPL assigns morphological parses (prefixed vs. monomorphemic parses) to all the words in the input corpus. Of course, only the parses assigned by DDPL to potentially prefixed words, i.e., words beginning with a string



identical to one of the prefixes found by the algorithm, are of interest, as all other input words are treated as monomorphemic.

Notice that assessing the plausibility of the parses assigned by DDPL to potentially prefixed words is not a trivial task, as the morphological status of many potentially prefixed words is not clear. While probably everybody would agree that the word *redo* is prefixed and the word *red* is not, there are many intermediate cases (such as *resume*, *recitation*, *remove*) about whose status morphologists would probably disagree.

Thus, rather than trying to decide on my own which of the parses assigned by DDPL were “right” and which ones were “wrong”, I conducted a survey in which I collected morphological complexity ratings from native English speakers, and then I compared these ratings to the parses assigned by DDPL to the same words, in a correlation analysis. The idea was to see if speakers’ intuitions on the prefixed status of such words would agree with the parses assigned by the algorithm.

All the words in the survey corpus began with a string corresponding to one of the prefixes postulated by DDPL, but half of the words were selected from the set of forms that were treated as complex by the model (for example, *disability*), the other half from the set of forms which, although they begin with a string identical to a prefix, were *not* treated as complex by the model (for example, *cocoon*).

In particular, the survey corpus contained 300 forms that were randomly selected from the words in the DDPL output that began with one of the prefixes postulated by the algorithm (excluding *man-*, *juris-* and *radio-*). 150 of these forms were randomly selected from the set of words that DDPL treated as prefixed. The other 150 forms were randomly selected from the set of words that DDPL treated as monomorphemic (non-prefixed).

Before I present the results of the DDPL/speakers correlation analysis, I will describe how the morphological complexity ratings were collected, and I will discuss some

issues related to the nature of this kind of data, which are rather unusual in morphological studies.<sup>64</sup>

#### **4.4.1 Collecting morphological complexity ratings**

##### **4.4.1.1 Methodology and data collection**

A group of eight native English speakers were asked to rate a set of potentially prefixed words (“potentially prefixed” in the sense that they begin with a word-initial string identical to a prefix) on a scale from 1 to 5, assigning 1 to words that they definitely feel to be non-prefixed, 5 to words that they definitely feel to be prefixed.

In the instructions (see appendix 1), the participants were presented with an example of a word that should receive a 1-rating (the word *cocoon*) and an example of a word that should receive a 5-rating (the word *coexist*).

I originally planned to use a 3-point scale (“non-prefixed”, “don’t know”, “prefixed”) but later opted for the 5-point scale after some of the participants in a pilot study reported that they felt they needed more than 3 rating options to satisfactorily perform the rating task.

The 300 words in the survey corpus were presented in a different random order to each participant. Words were presented in a list format. To avoid possible ambiguities due to the fact that some of the prefixes under investigation are substrings of other prefixes

---

<sup>64</sup>The only other study I am aware of in which this kind of data is presented is Smith 1988.

(e.g. *in-* is a substring of *inter-*), each word in the list was followed by the potential prefix that participants were to consider.

Participants were given unlimited time to complete the task, but they were asked to write down their ratings as quickly as possible, and to avoid revising a rating once they had written it down.

Clearly, in order to take part in the survey, participants had to be familiar with basic notions of morphology (at least, the notions of prefix and prefixed form). Thus, I selected the participants among undergraduate and graduate linguistics students: 5 participants were UCLA undergraduate linguistics majors; 3 participants were graduate students in the UCLA linguistics program.

#### **4.4.1.2 Are morphological complexity ratings valid data?**

The first issue we have to consider is whether the participants were able to perform the task, or whether they randomly assigned ratings to the words in the survey.

If participants assigned ratings randomly, then we would expect that their ratings would not be correlated with each other, but this is not the case. I computed Pearson and Spearman correlation coefficients for the ratings of each pair of speakers.<sup>65</sup> This set of analyses showed that the rating patterns of all eight participants were highly correlated (the Pearson correlation coefficients computed pairwise for each pair of raters were always

---

<sup>65</sup>For the notions of statistics employed in this chapter, see, for example, Woods, Fletcher and Hughes 1986, in particular chapter 10.

higher than .55; the Spearman correlation coefficients computed pairwise for each pair of raters were always higher than .6).

This inter-speaker agreement indicates that the participants did not assign ratings randomly. However, this by itself does not entail that the ratings were indeed based on the way in which the speakers represent the relevant words in their lexicon.

First of all, it could be that speakers based their judgment on explicit linguistic knowledge. In particular, they could have assigned ratings on the basis of the morphological theory of their choice or on the basis of whether a certain word is etymologically complex or not.<sup>66</sup>

As I checked in informal interviews with the participants, none of the undergraduate students who took part in the survey had a serious background in Latin, Greek or the history of English. While it was harder to verify this, it also seems that the knowledge of morphology of the undergraduate students who participated in the survey was relatively limited, and none of them seemed to be familiar with a morphological theory detailed enough to predict the morphological status of a list of words.<sup>67</sup> Thus, the hypothesis that ratings were assigned on the basis of etymology or morphological theory can be excluded at least for the majority of participants.

Still, one could object that, even if speakers did not rely on their explicit knowledge of morphology to assign ratings, the kind of implicit linguistic knowledge accessed in a

---

<sup>66</sup>Unfortunately, morphological complexity as determined by distributional cues and etymology will tend to be highly correlated: strings such as *-sume* and *-sist* occur in combination with a number of strings identical to prefixes, in English, because they correspond to real Latin stems.

<sup>67</sup>What is said here also holds for the undergraduate linguistics majors who participated in the survey discussed in 4.5 below.

rating task is distinct from (and not determined by) the implicit linguistic knowledge that speakers access during real communication. Notice that, for this objection to be problematic, we have to assume not only that when speakers assign ratings they access an implicit metalinguistic knowledge component, and not their lexical representations, but also that the knowledge stored in this component does not derive from the speakers' actual linguistic competence, but from some other, as-yet-mysterious source.

While I cannot exclude this hypothesis, I believe that, as it is not clear what the status of this implicit metalinguistic component unrelated to actual linguistic knowledge would be, the burden of proof is on the supporters of the hypothesis of its existence. Until such evidence is provided, it is reasonable to maintain the simpler working hypothesis that speakers' morphological ratings derive from their direct or indirect access to lexical representations of the words under analysis.

#### **4.4.2 DDPL parses and speakers' complexity ratings: results and discussion**

As I discussed in the previous section, the rating patterns of all eight participants were highly correlated. Thus, I computed the per-word average rating across all participants, and I compared the resulting variable to the parses assigned by DDPL to the same words (coded by assigning 1 to words treated by DDPL as prefixed and 0 to words treated by DDPL as non-prefixed) in a correlation analysis.

The Spearman coefficient of this analysis was .62 ( $p < .000$ ), i.e., speakers' morphological ratings and DDPL parses are correlated.<sup>68</sup> Thus, we can conclude that, besides being able to find a number of actual English prefixes, DDPL also assigned plausible morphological parses to potentially prefixed words.

Of course, while the correlation between DDPL and the speakers' ratings is significant, it is by no means "perfect", as there are some discrepancies between the speakers' ratings and the DDPL parses (see appendix 3, where I present the list of words in the survey, reporting for each word how it was parsed by DDPL and the average speakers' rating).

Several reasons can explain these discrepancies. First, as there are individual differences in morphological intuitions (and probably representations) among speakers (indeed, the ratings of some of the speakers were less correlated with each other than the DDPL parses and the average ratings), I think that, even if DDPL were a perfect model of human morpheme discovery, we should not expect a 100% correlation between its parses and an average of human intuitions.

More importantly, the DDPL is intended to model a hypothesized early stage of morpheme discovery in which learners rely entirely on distributional cues (see discussion in chapter 1). However, the speakers who participated in the survey are adults who

---

<sup>68</sup>Similar results were obtained in an ANOVA in which the DDPL parses were used to predict the speakers' ratings. Notice that, with Spearman coefficients,  $r^2$  cannot be interpreted as a value indicating how much of the variance in one variable can be explained by the other variable (see discussion in Woods, Fletcher and Hughes 1986: 10.7).

successfully completed the task of morphological acquisition, and are aware of the semantic and syntactic properties associated with prefixes and stems.<sup>69</sup>

From this perspective, it is actually surprising that the purely distributionally-driven parses assigned by DDPL are as correlated to adult speakers' ratings as the results indicate.

Interestingly, the discrepancies between DDPL and English speakers appear to be attributable to the fact that DDPL is too "conservative", i.e., DDPL was more likely to treat obviously prefixed forms as simple than obviously simple forms as complex. The reader can verify this by inspecting the data in appendix 3: while it is easy to find obvious misses among the forms treated as simple by DDPL (*unconsciously*, *distrust*, *subgroups*, *unavoidable...*), I would judge that only two of the forms treated as complex by DDPL are obviously non-prefixed (*comin*,<sup>70</sup> *constable*).

The same point is made by the following analysis: The average mean rating across all forms that were treated as complex by DDPL is a rather high 4.05 (recall that speakers had to rate forms on a scale from 1 to 5, assigning 5 to clearly prefixed forms). This indicates that in general speakers largely agree with DDPL on the status of forms that the algorithm treated as complex. On the other hand, the average mean rating across all forms that were treated as simple by DDPL is 2.11. This is still lower than the chance level, but does suggest that there was less overlap between DDPL parses and speakers' intuitions in the domain of forms that are simple for the computational model.

---

<sup>69</sup>Indeed, I suspect that the reason why speakers feel more comfortable with a 5-point scale than with more categorical parses is that they try to accommodate cases in which distributional and semantic cues are in conflict by assigning intermediate ratings.

<sup>70</sup>This form is probably a spelling of the colloquial form of *coming*.

Thus, as with the list of prefixes found by DDPL, what emerges here is that the analysis generated by the model is quite accurate (very few “false positives”) but not exhaustive (many “misses”).

Notice that, as I discussed in the introduction and in chapter 1, the purpose of assigning parses to potentially complex words in morpheme discovery is to have a set of forms to analyze in order to discover the semantic and grammatical properties of affixes. In this perspective, it seems that morpheme discovery should indeed favor accuracy over exhaustiveness: a relatively small set of words containing a certain prefix is probably more helpful, in identifying the properties of that prefix, than a larger set that also includes many pseudo-prefixed forms.

#### **4.4.3 Assessing the performance of the distribution-driven model: concluding remarks**

The analysis of the results of the DDPL simulation shows that distribution-driven heuristics such as the ones implemented by this model can be quite helpful in morpheme discovery, both in terms of finding the prefixes of a language and in terms of assigning morphological parses to words.

As I discussed in the introduction of this study, the success of this computational simulation constitutes evidence against the claim that children cannot *in principle* learn something about morphology from distributional evidence, since distributional evidence does not provide enough useful cues. Clearly, even the relatively simple distributional cues used by DDPL could be of great help to language learners.



We observed in both the analysis of the prefixes and the analysis of the parses that DDPL is better in terms of accuracy than in terms of exhaustiveness. At least with respect to the misses in the list of prefixes, the excessive conservativeness of DDPL is in part due to the lexicon generation component of the model, which should probably be revised.

#### **4.5 Evidence from the morphological treatment of semantically opaque words: a second survey**

As I discussed at length in the introduction and in chapter 1 (see in particular section 1.4.3.1), the treatment of semantically opaque but potentially prefixed forms constitutes a potential source of evidence on the “psychological relevance” of distributional cues in morpheme discovery.

Without going into details, the argument goes like this: If we show that DDPL assigns parses matching speakers’ intuitions to potentially prefixed but semantically opaque words, then it seems reasonable to conclude that speakers relied on distributional cues such as the ones used by DDPL when determining the morphological status of those words, as they could not have relied on semantics (nor other grammatical cues -- see 3.2.2 above).

To test this, I designed another survey, using the same methodology described in section 4.4.1, but with a corpus composed entirely of semantically opaque words.

#### 4.5.1 Constructing the semantically opaque word list

Following a standard practice in morphological processing studies (see, for example, Marslen-Wilson, Tyler, Waksler and Older 1994), I first conducted a survey in which three judges were asked to rate a set of forms from the DDPL output for semantic transparency, and then I selected forms that received a low average semantic transparency rating to construct the survey corpus.

As the DDPL output contains a total of 3,651 forms beginning with one of the prefixes postulated by the model, it was not feasible to ask the semantic transparency judges to assign a rating to all the forms. Thus, the corpus presented to the judges was constructed in the following way.

First, I made a preliminary division of the 382 words treated as prefixed by DDPL into two categories: words that I judged to be obviously prefixed (productively formed, semantically transparent), and words that may or may not be prefixed (this preliminary list included a wide range of types, from obviously monomorphemic words such as *adage* to only slightly lexicalized forms such as *inhumane*). The first list was composed of 101 words, the second list of 181 words. I randomly selected 10 words from the first list, and I kept all the 181 words from the second list.

I then randomly selected, from the list of the remaining 3269 words that begin with a string identical to a prefix but are treated as simple by DDPL, 10 more words that were obviously prefixed and completely transparent, and 200 words that may or may not be prefixed.

The corpus presented to the three judges was composed of the 20 completely transparent words and 381 “ambiguous” words selected in this way. The 20 completely transparent words served both as a form of control and, more importantly, to make sure

that the judges were not going to rate some semantically opaque forms as transparent merely to make use of the whole scale.

The judges were two graduate students and one postdoctoral fellow in the UCLA Linguistics Department, and were selected because of their strong background in morphology/processing.<sup>71</sup> Judges were asked to rate the words in the corpus on a scale from 1 to 5, assigning 1 to completely opaque words and 5 to completely transparent words. The instruction sheet given to the judges is presented in appendix 2.

A series of correlation analyses showed that the judges' ratings were highly correlated (both Pearson and Spearman correlation coefficients in all pairwise comparisons were higher than .7). Thus, I computed the average cross-judge rating for each word in the corpus.

As expected, the 20 transparent words received very high ratings (the mean rating for this set of words was 4.89). Of the remaining forms, 97 out of the 181 words treated as prefixed by DDPL received an average rating lower than 2.5; 183 out of the 200 words treated as simple by DDPL received an average rating lower than 2.5.<sup>72</sup>

---

<sup>71</sup>I selected "expert" judges because I wanted to make sure that they would understand the task, and in particular that they would understand the distinction between rating forms on the basis of semantic transparency vs. morphological complexity.

<sup>72</sup>Notice the asymmetry between the two sets: just a little more than half of the complex (for DDPL) words that were pre-selected as potentially opaque are indeed semantically opaque, whereas 90% of the simple (for DDPL) words that were pre-selected as potentially opaque are indeed semantically opaque. This suggests that, although DDPL did not have access to semantic information, the model did show a preference for treating semantically opaque words as simple. This is good from the point of view of a general assessment of the DDPL performance, but it made it harder to design the survey presented here.

The corpus for the morphological complexity survey was thus composed of the 97 prefixed (for DDPL) forms that had a semantic rating lower than 2.5, and 97 randomly selected words from the 183 simple (for DDPL) words with a semantic rating lower than 2.5.<sup>73</sup> The average semantic rating across the prefixed (for DDPL) forms in this corpus was 1.54; the average rating across the simple (for DDPL) forms in this corpus was 1.21.<sup>74</sup> The survey corpus, together with the average complexity rating and DDPL parse of each form, is presented in appendix 4.

---

<sup>73</sup>I decided not to add a control set of semantically transparent forms, as I wanted to maximize the participants' sensitivity to differences in morphological status among opaque words. If some semantically transparent words had been inserted, speakers would have probably reserved the high values of the rating scale for such forms, "squeezing" the ratings of semantically opaque words within a narrow range at the bottom of the scale.

<sup>74</sup>One of the judges was also asked to rate the 194 forms in the corpus by assigning ratings on a 5 point scale on the sole basis of the degree of semantic transparency of the potential *prefix* of each form. The average prefix transparency rating across forms treated as complex by DDPL was 1.86; the average prefix transparency rating across forms treated as simple by DDPL was 1.46. Thus, while there is a noticeable and slightly worrisome difference in the degree of prefix transparency between the two sets, it seems safe to state that not only the forms in both sets are semantically opaque when considered as wholes, but also that the potential prefixes occurring in them tend to be opaque.

#### **4.5.2 Methodology and data collection/analysis**

The same methodology described in section 4.4.1 was followed in this second survey. The words were presented in random list format, followed by the corresponding potential prefixes, and the participants were asked to rate them on a 5-point scale, assigning 1 to clearly non-prefixed forms and 5 to clearly prefixed forms. The instructions given to the participants in this survey were identical to the ones given to the participants in the previous survey.

A group of eight English native speakers took part in the survey. None of them had participated in the previous study. Of these eight speakers, one was a postdoctoral fellow and two were graduate students in the UCLA linguistics department. The other five participants were UCLA undergraduate linguistics majors.

Pairwise Pearson and Spearman correlation coefficients were computed for the ratings of all pairs of participants. The patterns of three participants were poorly correlated with those of the other participants (and with each other). Thus, their data were discarded.<sup>75</sup>

As the ratings of the remaining participants were highly correlated (all pairwise Pearson and Spearman coefficients were higher than .5), the per-word average rating value across these participants was computed, and the resulting variable was compared to the parses assigned by DDPL to the same words (coded by assigning 1 to words treated by DDPL as prefixed and 0 to words treated by DDPL as non-prefixed) in a correlation analysis.

---

<sup>75</sup>For each of these three participants, the correlation coefficient between her/his ratings and those of a majority of other speakers was lower than .4.

The Spearman coefficient of this analysis was .46 ( $p < .000$ ), i.e., the speakers' morphological ratings of the list of semantically opaque words in the survey were significantly correlated to the parses assigned by DDPL to the same words.<sup>76</sup>

#### **4.5.3 Morphological complexity of semantically opaque words: discussion**

If the participants in the survey had mostly relied on semantic cues when assigning ratings to the words in the list, they should have assigned uniformly low ratings to all words.

However, this was not the case: as shown by the correlation between the average ratings and DDPL parses, in general speakers assigned higher ratings to words that DDPL treated as complex, lower ratings to words that DDPL treated as simple. The average mean rating across all words that were complex for DDPL was 3.78; the average mean rating across all words that were simple for DDPL was 2.81.

The most plausible explanation for this asymmetry is that the way in which speakers represent potentially complex words is affected by distributional factors such as the ones implemented in DDPL.<sup>77</sup> In turn, a plausible hypothesis about why distributional

---

<sup>76</sup>Similar results were obtained in an ANOVA in which the DDPL parses were used to predict the speakers' ratings.

<sup>77</sup>Donca Steriade pointed out a possible confound in the survey word list: The speakers could have decided that some of the words in the list are simple or complex on the basis of phonological (rather than distributional) cues. For example, speakers could have unanimously rated *cod* as non-prefixed simply because English stems must contain a syllabic segment; and they could have been more inclined to treat *suppress(ed)* as complex because it is unusual for non-prefixed disyllabic verbs ending in CVC (with short

factors have an effect on speakers' morphological intuitions is that speakers relied on distributional cues during morpheme discovery.

On the other hand, adult speakers are obviously also sensitive to semantic cues, when rating words for morphological complexity. As all the words in the survey corpus were semantically opaque, it is not surprising that the results of this second survey are less clear-cut than those of the previous survey (as shown by the lower correlation coefficient and by the fact that there is less difference between the average mean ratings assigned to DDPL simple and complex words).

I suspect that semantics influenced the results both directly and indirectly. First, the morphological representations of adult speakers are almost certainly affected by the semantic structure of words. Thus, while speakers seem to be able to distinguish words that are complex on purely distributional grounds from simple words, still it is likely that

---

V) to be stressed on the last syllable. I asked an experienced phonologist to list all the words in the survey corpus for which speakers could have relied on phonological cues (either to decide that they are simple, or to decide that they are complex). The phonologist assigned marks according to the following criteria: A form was marked as "simple from a phonological point of view" if the potential stem of the form did not contain a syllabic segment; and/or the potential stem was stressless; and/or the potential stem contained a phonotactically or orthographically impossible stem-initial sequence, and/or the potential prefix or stem contained other phonologically impossible structures. A form was marked as "complex from a phonological point of view" if it was a disyllabic verb ending in CVC (with short V) stressed on the last syllable (or a derivative of such form). In total, the phonologist marked 33 forms. I ran a Spearman correlation analysis after removing these forms. The results were very similar to the ones reported in the text (correlation coefficient = .44,  $p < .000$ ). This preliminary analysis suggests that the results reported here are not due to phonological confounds.

such words are not as straightforwardly complex as semantically transparent forms. Hence, we expect that speakers will be less inclined to assign very high ratings to distributionally complex but semantically opaque words.<sup>78</sup>

Moreover, as a consequence of the fact that the distinction between semantically opaque but complex forms and simple forms is probably not as clear-cut as the distinction between complex and transparent words and simple words, the participants in the second survey had to provide ratings based on more subtle judgments, requiring more sophisticated metalinguistic introspection skills. Thus, as this was a harder task, it is likely that the participants in the second survey had more difficulty with their task than the participants in the first survey, and that the lower correlation coefficient is in part due to “noise” in the ratings.

However, beyond these considerations, what is truly important from our point of view is that, still, there *is* a high correlation between DDPL parses and some speakers’ ratings of semantically opaque words. Thus, the survey results provide support for the hypothesis that humans are sensitive to distributional cues to morphological constituency such as the ones used by DDPL.

---

<sup>78</sup>Indeed, if no correlation between DDPL and the speakers had emerged, we could not have been sure that the negative result was due to the fact that speakers do not rely on distributional cues such as the ones employed by DDPL during morpheme discovery. The negative result could have instead been due to the fact that, once speakers acquire sufficient evidence about the semantic properties associated with morphemes, they revise their morphological representation of forms, and they change (from complex to simple) the representation of those forms that were originally treated as complex on distributional grounds, but whose complex representation is not supported by semantic evidence.



Notice that this is a weaker claim than the one we are truly interested in, i.e. that humans use distributional cues such as the ones used by DDPL during morpheme discovery. However, I believe that it is reasonable to hypothesize that humans are sensitive to distributional cues *because* they relied on them to discover morphemes.

#### 4.6 Testing DDPL with a phonetically transcribed input

I ran DDPL with the same list of words from the PHLEX database used for the analyses described above, but using phonetic transcriptions of the input words.<sup>79</sup> In the cases in which a word had multiple phonetic transcriptions, the first of the transcriptions was selected.

Given this input, DDPL came up with the list of prefixes in (90):

(90) eks    ekstr   imp   in   instru   intər   ʌn   ʌndər   dɪ   dis   di  
kən   kəm   kən   mæn   nan   pɛrə   pri   rɛ   reɪdiə   ri   sʌb  
supər   træn

Notice that *ekstr* is an allomorph of *extra-* (occurring before vowels in forms such as *extraordinarily*) and *træn* is an allomorph of *trans-* (occurring before s in forms such as *transcript*).

This list contains the false positives *mæn* (*man-*) and *reɪdiə* (*radio-*), which are also in the list of orthographic prefixes in (85). Moreover, the list in (90) contains the false positives *imp* and *instru*, which are not in (85), and which do not correspond to any plausible linguistic constituent.

The following prefixes from the list in (85) were missed in the simulation with phonetic transcriptions:

(91) ad-    auto-   co-   cor-   mis-   over-   sup-   sur-   tele-

---

<sup>79</sup>Stress and syllable boundary marking symbols were removed from the transcriptions.

With the exception of *co-* and *sup-*, these prefixes were missed in this run because they did not meet the constraint requiring word-initial strings to occur before a certain number of “long” word-final strings (something probably due to the fact that phonetically transcribed words tend to be in general shorter than orthographically transcribed words, with a different distribution of length across forms).<sup>80</sup>

*Trans-* (more precisely, its allomorph *træn*) was the only prefix found in the simulation with phonetically transcribed input but not in the simulation with orthographically transcribed input.

Clearly, future research should address issues related to assessing the performance of DDPL with a phonetically transcribed input in more detail.

## 4.7 Summary

In this chapter, I presented the results of a simulation in which the DDPL model was tested with a list of English words from the Brown-PHLEX corpus as its input. The results of the simulation suggest that DDPL is, to a large extent, successful at finding prefixes and assigning morphological parses to words. The problems the model encountered appear to be due to its lexicon generation component, rather than to the MDL-based lexicon selection formula at its core.

Moreover, I presented the results of a survey that shows that the morphological complexity ratings assigned by English speakers to semantically opaque but potentially

---

<sup>80</sup>In this simulation,  $l = 6.4$ .

prefixed forms are correlated with the parses assigned by DDPL to the same words. As I argued in previous chapters, this constitutes evidence that humans are sensitive to distributional cues such as the ones implemented in DDPL.

## **Chapter 5**

### **Conclusion**

The results of the simulation reported in the previous chapter provide support for the general hypothesis that distributional information of the kind encoded in the DDPL model can in principle be helpful in morpheme discovery. Moreover, the reported convergence between the DDPL parses and speakers' ratings of a set of semantically opaque words provides some preliminary support for the hypothesis that humans rely on distributional cues such as the ones employed by the automated learner when assigning morphological parses to some words. A plausible explanation of this finding is that speakers are sensitive to such cues because they employed them in order to assign morphological parses during morpheme discovery.

Clearly, while I believe that the results presented are encouraging, many questions are still open, and much more research has to be done before we can reach safe conclusions about the nature and role of distributional evidence in morpheme discovery. In particular, I will conclude this study by discussing some of the future directions that the computational and empirical work I am reporting could take.

In terms of improving the DDPL model, the first step one should take would be to design alternative lexicon generation algorithms, which explore a larger (or, rather, a morphologically more sensible) area of the hypothesis space. I observed that the lexicon selection formula based on the shortest lexicon + encoding criterion constitutes the conceptual core of the computational model. However, in order for a lexicon to be evaluated by the selection component of the model, that lexicon must have been generated by the generation component.

As I discussed in chapter 4, the lexicon generation component fails to generate some crucial analyses that should be evaluated (and, one hopes, selected) by the lexicon selection component. For example, the lexicon generation component fails to generate analyses in which the string *post-* is treated as a prefix.

A very simple way to assess the impact of this problem would be to *force* the lexicon generation component to generate some of the analyses that we know to be plausible (for example, the analysis in which *post-* is a prefix). In this way, we could at least check whether, once it is artificially presented with a plausible analysis of the input, the lexicon selection component is able to select it (for example, if the lexicon selection component would select a lexicon in which *post-* is a prefix, once it is presented with such a lexicon).

However, while this approach could allow us to assess the performance of the shortest lexicon + encoding model in a more satisfactory way, it would be of course crucial, in the long term, to come up with a better lexicon generation algorithm, which generates better analyses without requiring *ad hoc* interventions.

The DDPL model should be extended and revised in many other respects as well. For example, it would be interesting to design and test variations of the model in which the weights assigned to different heuristics are changed, and to compare the results in order to assess the role that each of the heuristics is playing in the model.

Moreover, it would be interesting to extend the model to suffixation, and possibly to design algorithms in which the distributional information used by DDPL is integrated with other types of information (such as syntactic category information).

From the point of view of testing the model, we should first of all test DDPL in simulations with other English corpora, both in orthographic and phonetic transcriptions. Furthermore, DDPL should be tested using input corpora from other languages.

In terms of collecting empirical evidence, we should first of all collect data from more speakers, possibly re-designing the survey task in order to make it feasible for speakers with no linguistics background. Furthermore, it would be interesting to collect data using other methods (for example, using a morphological priming paradigm), to make sure that the results we obtained are not task-specific. Finally, it would of course be important to collect developmental data from children, to have a more concrete idea of when and how human learners perform morpheme discovery.

While all these lines of research should be pursued in the near future, and I am sure that readers will raise other important issues that were not dealt with here, I believe that this study (together with the work of Brent and Goldsmith reviewed in chapter 2) constitutes an encouraging starting point for the investigation of morpheme discovery in general, and of the role of distributional cues in this domain in particular.

## Appendix 1

### *MORPHOLOGICAL COMPLEXITY RATING INSTRUCTIONS*

Each of the words in the attached list is potentially prefixed, in the sense that it begins with a string identical to an English prefix.

Probably, you will have the intuition that some of these words are actually prefixed, others are not. For example, both the words *coexist* and *cocoon* begin with the string *co*. However, it is likely that you will have the intuition that, while the word *coexist* is prefixed (i.e. it is composed of the prefix *co* plus the stem *exist*), the word *cocoon* is not prefixed (i.e. it is not composed of *co* plus *coon*). There are intermediate cases in which the morphological status of words (prefixed vs. non-prefixed) is not so obvious, and different speakers may have different intuitions.

Your task is to rate the degree of prefixedness of each word, on the basis of your native speaker intuitions (and NOT on the basis of what you learned in linguistics classes or elsewhere!) You should rate these words on a 5 point scale, assigning 1 to words that are clearly non-prefixed (such as *cocoon*) and 5 to words that are clearly prefixed (such as *coexist*). Use the intermediate values for less clear-cut cases.



Since the words of the list were randomly selected from a corpus of written English, the list may contain last names, misspellings, technical terms etc. If you do not have intuitions about such forms, please assign them a rating of 1.

To avoid ambiguities, I entered next to each word the potential prefix I would like you to consider. This means that, for example, even if a word begins with the string *inter*, if the word is followed by *in*, you should decide whether the word contains the prefix *in*, and not the prefix *inter*.

Please, complete the task in one session. Try to assign ratings quickly, and do not change ratings once you wrote them down.

*NB: in the version of these instructions presented to graduate student and post-docs (and NOT on the basis of what you learned in linguistics classes or elsewhere!) was replaced with (and NOT on the basis of their etymology or the morphological theory of your choice!).*

## **Appendix 2**

### ***SEMANTIC TRANSPARENCY SURVEY INSTRUCTIONS***

Each of the words in the attached list is potentially prefixed, in the sense that it begins with a string identical to an English prefix.

Your task is to rate the degree of semantic transparency of each (potentially) prefixed word on a 5 point scale, using 1 for completely semantically opaque words (no relation between the meaning of the word and the meanings of the potential components) and 5 for completely semantically transparent words (the meaning of the word is entirely predictable from the meanings of the components).

Notice that there could be mismatches between your semantic analysis and your morphological intuitions. In particular, you may have the intuition that certain words are morphologically complex (prefixed) even if they are (almost) completely opaque from a semantic point of view. In such cases, you should assign ratings on the basis of your semantic analysis, and not on the basis of your morphological intuitions.

Since the words of the list were randomly selected from a corpus of written English, the list may contain last names, misspellings, technical terms etc. If you do not have intuitions about the semantic structure of such forms, please assign them a rating of 1.

To avoid ambiguities, I entered next to each word the potential prefix I would like you to consider.

## Appendix 3

### *RESULTS OF FIRST SURVEY*

(*ddpl*: parses assigned by DDPL, coded as 0 = non-prefixed, 1 = prefixed; *avg\_rating*: average per word complexity rating across the participants in the survey).

<i>word</i>	<i>prefix</i>	<i>ddpl</i>	<i>avg_rating</i>
ada	ad	0	1
additional	ad	0	1.625
adelia	ad	0	1
administered	ad	0	2.125
administration	ad	0	2.125
adolescent	ad	0	1.25
adopted	ad	0	1.375
automobile	auto	0	3.375
coastal	co	0	1
cocoon	co	0	1
cohesive	co	0	2.5
coincidence	co	0	4.125
college	co	0	1
collusion	co	0	1.625
colonel	co	0	1

columnist	co	0	1
commissions	com	0	2.125
commotion	com	0	1.5
communication	com	0	1.625
communist	com	0	1.5
company	com	0	1.25
competition	com	0	1.25
components	com	0	1.625
conceal	con	0	1.625
conclusion	con	0	2.625
conferred	con	0	2.125
confide	con	0	1.75
congregational	con	0	1.875
conjugates	con	0	1.625
conservatory	con	0	2
conspired	con	0	1.875
constituents	con	0	2
constitution	con	0	1.5
consultation	con	0	1.5
contends	con	0	1.75
contingencies	con	0	2
contradiction	con	0	1.75
convoy	con	0	1.375
coop	co	0	1.5

corporations	cor	0	1.25
corso	cor	0	1
costume	co	0	1
couple	co	0	1.5
coupler	co	0	1
couplers	co	0	1
courteously	co	0	1
courting	co	0	1
debonnie	de	0	1.375
debris	de	0	1.25
deductible	de	0	2.375
deemed	de	0	1
definition	de	0	1.5
delegation	de	0	1.5
delivers	de	0	1.25
delphine	de	0	1
denied	de	0	1.375
denominations	de	0	3.25
depot	de	0	1.25
describe	de	0	2.375
destinies	de	0	1
destroyers	de	0	1.375
destroying	de	0	1.375
detachment	de	0	3.375

detergents	de	0	1.5
dewey	de	0	1
disarmament	dis	0	4.75
discharges	dis	0	3.5
discs	dis	0	1
disparate	dis	0	2.375
dissuade	dis	0	2.875
distinctions	dis	0	1.5
distinctly	dis	0	1.625
distrust	dis	0	5
examination	ex	0	1.375
exchanged	ex	0	4
exclamation	ex	0	1.875
excluded	ex	0	2.375
execute	ex	0	1.375
executives	ex	0	1.375
experts	ex	0	1.25
explaining	ex	0	1.625
incestuous	in	0	2
inch	in	0	1
increasingly	in	0	2.25
incredibly	in	0	3.375
incumbent	in	0	2.5
indebted	in	0	3.5

indulge	in	0	2
inform	in	0	1.875
initiation	in	0	1.5
inna	in	0	1.375
instance	in	0	1.5
insulation	in	0	1.875
intensify	in	0	2
intensity	in	0	1.625
intentions	in	0	1.75
internal	inter	0	1.875
intimately	in	0	1.75
intrusion	in	0	2.25
investigating	in	0	2.25
precincts	pre	0	2.125
prescription	pre	0	3.5
presiding	pre	0	2.125
pretense	pre	0	2.875
psychological	psycho	0	3.875
reached	re	0	1
reactors	re	0	2.125
realizes	re	0	1.125
rear	re	0	1
reason	re	0	1
reckoning	re	0	1



reconciled	re	0	2.75
recordings	re	0	1.25
reddish	re	0	1
redoute	re	0	2.5
reed	re	0	1
referral	re	0	1.625
register	re	0	1.125
registration	re	0	1.25
reid	re	0	1
repeal	re	0	2.375
repertory	re	0	1.5
replacing	re	0	4.25
reporters	re	0	1.5
resolve	re	0	3.375
respiratory	re	0	1.25
response	re	0	1.625
retained	re	0	2.5
retaining	re	0	2.375
returned	re	0	2.875
reversed	re	0	2.375
subgroups	sub	0	5
subscription	sub	0	2.75
subsequent	sub	0	3.25
subtracting	sub	0	2.875

supernaturalism	super	0	4.625
supersonic	super	0	4.375
supremely	sup	0	1.25
surrounded	sur	0	1.875
survival	sur	0	1.875
telegrapher	tele	0	3.625
unavoidable	un	0	5
unconsciously	un	0	4.375
underground	under	0	4.25
underlying	under	0	4.125
underworld	under	0	4.625
unemployment	un	0	5
unloaded	un	0	5
unorthodox	un	0	5
unwanted	un	0	5
administering	ad	1	2.25
admissions	ad	1	2
autobiographical	auto	1	4.375
autobiography	auto	1	4.875
autofluorescence	auto	1	3.875
comin	com	1	1
complains	com	1	1.375
composing	com	1	1.75
compresses	com	1	2.5

conducts	con	1	2.125
conformation	con	1	2.75
conformed	con	1	2.25
connotation	con	1	2.875
conserve	con	1	2.25
constable	con	1	1.5
convent	con	1	1.5
corresponded	cor	1	2.375
defender	de	1	1.75
detested	de	1	1.125
disability	dis	1	5
disadvantage	dis	1	5
disappearing	dis	1	4.375
disapprove	dis	1	5
disapproved	dis	1	5
disarmed	dis	1	4.875
disfigured	dis	1	4.625
disgrace	dis	1	3.125
dislikes	dis	1	5
dismounted	dis	1	4.75
disobeyed	dis	1	5
disorganized	dis	1	5
disprove	dis	1	5
dissection	dis	1	2.75

dissolving	dis	1	2.875
disunity	dis	1	4.5
excite	ex	1	1.5
exclaiming	ex	1	1.875
extracts	ex	1	2.375
extraordinarily	extra	1	4.5
extraterrestrial	extra	1	4.75
inaccurate	in	1	5
incite	in	1	1.875
inconsistent	in	1	4.875
inconvenience	in	1	4.875
indisposed	in	1	4
ineligible	in	1	4.875
inflexible	in	1	5
informally	in	1	3.375
infrequent	in	1	5
inholdings	in	1	2.75
inhumane	in	1	5
inroads	in	1	3.25
insanity	in	1	4
insecure	in	1	5
insensitive	in	1	4.5
insides	in	1	2.5
interfaces	inter	1	3.625

internationally	inter	1	4.875
interrelation	inter	1	4.75
interrelations	inter	1	4.75
interstage	inter	1	3.75
intertwined	inter	1	4.25
inviolate	in	1	3.75
involuntary	in	1	4.75
misled	mis	1	4.25
nonexistent	non	1	5
nonfiction	non	1	4.625
nonspecifically	non	1	5
nonverbal	non	1	5
nonwhite	non	1	5
overlap	over	1	3.625
overload	over	1	4.375
overlook	over	1	4.875
overlooks	over	1	4.5
overpayment	over	1	5
overreach	over	1	4.625
overtones	over	1	4.5
parapsychology	para	1	5
parasites	para	1	2.375
predetermined	pre	1	5
premature	pre	1	4.5

premix	pre	1	4.875
presumptuous	pre	1	2.25
psychoanalysis	psycho	1	4.5
psychologically	psycho	1	3.875
psychotherapy	psycho	1	4.375
reagents	re	1	2.625
rearrange	re	1	4.875
reassured	re	1	5
recollection	re	1	4.25
recollections	re	1	3.625
reconsideration	re	1	4.625
recounting	re	1	4.375
recovering	re	1	2.125
recurrent	re	1	4.25
redecorating	re	1	4.875
reformer	re	1	2.625
regaining	re	1	4.125
renamed	re	1	4.875
reorganized	re	1	5
repressed	re	1	3.375
resided	re	1	1.25
resides	re	1	1.25
restatement	re	1	5
resuspended	re	1	4.5

rewards	re	1	1.5
subcommittee	sub	1	5
subconsciously	sub	1	5
subsections	sub	1	4.75
subtitled	sub	1	4.625
supplant	sup	1	2
supporter	sup	1	1.375
unaffected	un	1	4.875
unanalyzed	un	1	5
unarmed	un	1	4.875
unawareness	un	1	4.5
unconditional	un	1	4.875
uncontrolled	un	1	5
undercurrent	under	1	4.375
underfoot	under	1	4.375
undershirt	under	1	4.125
underwrite	under	1	4.5
underwriters	under	1	4.125
undeveloped	un	1	5
undisciplined	un	1	5
undisturbed	un	1	4.75
undressing	un	1	4.875
uneconomical	un	1	4.875
unending	un	1	5

unexplained	un	1	5
unfitting	un	1	4.125
unfolds	un	1	4.625
unheard	un	1	4.875
unhurried	un	1	5
unkind	un	1	5
unlined	un	1	5
unmoved	un	1	4.875
unnamed	un	1	5
unnoticed	un	1	4.875
unpaired	un	1	4.5
unreasonable	un	1	4.875
unreliable	un	1	5
unrelieved	un	1	4.375
unsigned	un	1	4.875
unspoken	un	1	5
unstressed	un	1	5
unsuitable	un	1	4.875
unveiled	un	1	4.75
unwarranted	un	1	5
unwise	un	1	5



## Appendix 4

### *RESULTS OF SECOND SURVEY*

(*ddpl*: parses assigned by DDPL, coded as 0 = non-prefixed, 1 = prefixed; *avg\_rating*: average per word complexity rating across the 5 participants in the survey whose response patterns were highly correlated).

<i>word</i>	<i>prefix</i>	<i>ddpl</i>	<i>avg_rating</i>
administration	ad	0	4
admissible	ad	0	3.8
cod	co	0	1
collapsed	co	0	3.2
colors	co	0	1
commodity	com	0	2.2
commuting	com	0	3.8
comptroller	com	0	1.4
compute	com	0	3.2
comrade	com	0	3
concrete	con	0	3.6
concur	con	0	4.4
confronting	con	0	3.6
congressional	con	0	3.6

consensus	con	0	3.8
conspicuous	con	0	3.2
consultants	con	0	4
contemplate	con	0	3.8
contracting	con	0	3.8
contributes	con	0	3.6
conventions	con	0	3.8
conversations	con	0	3.8
copied	co	0	1
copper	co	0	1
cossack	co	0	1
cossacks	co	0	1
coupled	co	0	1
debts	de	0	1
decency	de	0	1.4
decrees	de	0	2.8
deeds	de	0	1
defends	de	0	3.6
delegate	de	0	2.8
denied	de	0	2
depravity	de	0	3
depth	de	0	1
destinies	de	0	1.4
detectable	de	0	2.8

developer	de	0	3
deviant	de	0	3.6
discipline	dis	0	2.4
dispensation	dis	0	4.2
distinct	dis	0	4
distinguishes	dis	0	3.2
distortion	dis	0	3.6
excerpts	ex	0	3.2
exciting	ex	0	3.2
executions	ex	0	2
executives	ex	0	2.6
exemption	ex	0	3
expenditure	ex	0	2.8
experimenter	ex	0	3.2
expressive	ex	0	3.8
extant	ex	0	2.6
extruded	ex	0	4.2
incepting	in	0	4
inclination	in	0	4
industrialized	in	0	2.6
infestations	in	0	3.6
initials	in	0	1.4
injuries	in	0	2.2
insults	in	0	3.4

intellectuals	in	0	2
intense	in	0	3
interesting	inter	0	2
introduces	in	0	2.4
inverse	in	0	3.6
involve	in	0	3.8
involvement	in	0	4
overlapped	over	0	4.8
parameter	para	0	4
prevot	pre	0	1
reader	re	0	1
rebellling	re	0	2.4
rebels	re	0	1.8
received	re	0	3.6
reflex	re	0	4.2
region	re	0	1
registration	re	0	2.6
regrets	re	0	2.4
reich	re	0	1
rejects	re	0	3.4
remain	re	0	3
remainder	re	0	3.2
remained	re	0	2.6
repeal	re	0	4.4

reporter	re	0	4.2
resin	re	0	1
respectable	re	0	2.6
respected	re	0	2.4
responsibility	re	0	2.6
revelation	re	0	3.8
reward	re	0	3.2
subsequent	sub	0	4.4
supported	sup	0	3.6
uniquely	un	0	1
unit	un	0	1
adage	ad	1	1.6
addiction	ad	1	3.6
administer	ad	1	4.2
administering	ad	1	4
admissions	ad	1	4
adsorbed	ad	1	3
coefficient	co	1	4.6
comin	com	1	1
compassion	com	1	4.4
compiling	com	1	3.6
complains	com	1	2.8
composing	com	1	3.6
compresses	com	1	3.8

concave	con	1	4.2
conducts	con	1	3.8
confirms	con	1	3.6
conformation	con	1	4.4
conformed	con	1	4.2
confronts	con	1	4
connotation	con	1	3.6
conserve	con	1	4.2
consoles	con	1	3.2
constable	con	1	3
consummation	con	1	4.2
contested	con	1	3.4
convent	con	1	3.6
convince	con	1	3.6
convocation	con	1	4.6
corresponded	cor	1	4.4
correspondents	cor	1	3.8
defender	de	1	3.2
defoe	de	1	1
delights	de	1	1.4
depositions	de	1	3.4
depressing	de	1	4.2
detested	de	1	2.8
discounts	dis	1	4.2

dismissing	dis	1	4.2
dispatched	dis	1	3.4
dispatches	dis	1	3
dissection	dis	1	4.2
dissolution	dis	1	4.6
dissolving	dis	1	4.4
exchanges	ex	1	4
exchanging	ex	1	4.2
excite	ex	1	3.4
exclaiming	ex	1	4
exposing	ex	1	3.6
extracts	ex	1	4.4
incite	in	1	3.8
incited	in	1	4
incorporation	in	1	4.6
infamous	in	1	4.8
informing	in	1	4
inholdings	in	1	4
injunction	in	1	4.2
inlets	in	1	4.4
insides	in	1	4.6
invest	in	1	3.8
overhaul	over	1	4.6
overlap	over	1	4.6

parades	para	1	1
parasites	para	1	3.6
parasol	para	1	3.6
preface	pre	1	4.4
prescribe	pre	1	4.6
presumptuous	pre	1	4
pretext	pre	1	4.2
recitation	re	1	3.8
recollection	re	1	4
recollections	re	1	4.6
recounting	re	1	4.6
recounts	re	1	4.4
recovering	re	1	4
recurrent	re	1	4.6
refine	re	1	4.4
reformed	re	1	4.2
reformer	re	1	4
repetitions	re	1	3.8
repressed	re	1	4.2
reserving	re	1	4
resided	re	1	2.8
resides	re	1	2.2
resolving	re	1	4.2
retailing	re	1	3.4



retreating	re	1	3.2
rewards	re	1	2.8
sublime	sub	1	3
submission	sub	1	4
supplant	sup	1	3.6
supporter	sup	1	3.6
suppositions	sup	1	4
suppressed	sup	1	3.8
surname	sur	1	4.8
surrendering	sur	1	3
underwrite	under	1	4.8
underwriters	under	1	4.8

## References

- Albro, D. 1998. POSCLASS: an automated morphological analyzer, manuscript, UCLA.
- Anderson, S. 1992. A-Morphous Morphology, Cambridge: Cambridge University Press.
- Aronoff, M. 1994. Morphology by itself, Cambridge: MIT Press.
- Baayen, H. 1994. Productivity in language production, Language and Cognitive Processes 9: 447-469.
- Baayen, H. and R. Lieber 1991. Productivity and English derivation: A corpus-based study, Linguistics 29: 801-843.
- Baayen, R., R. Piepenbrock and F. van Rijn 1993. The CELEX lexical database (CD-ROM), Philadelphia: Linguistic Data Consortium.
- Baayen, H., R. Schreuder and C. Burani *submitted*. Parsing and semantic opacity in morphological processing.
- Ballard, D. 1997. An introduction to natural computation, Cambridge: MIT Press.

- Baroni, M. 2000. Using distributional information to discover morphemes: A distribution-driven prefix learner, paper presented at the LSA Meeting, Chicago.
- Baroni, M. *in press* The representation of prefixes in the Italian lexicon: Evidence from the distribution of [s] and [z], Yearbook of Morphology.
- Bentin, S. and L. Feldman 1990. The contribution of morphological and semantic relatedness to repetition priming at short and long lags: Evidence from Hebrew, Quarterly Journal of Experimental Psychology 42A: 693-711.
- Brent, M. 1993. Minimal generative explanations: A middle ground between neurons and triggers, Proceedings of the 15th Annual Conference of the Cognitive Science Society: 28-36.
- Brent, M. and T. Cartwright 1996. Distributional regularity and phonotactic constraints are useful for segmentation, Cognition 61: 93-125.
- Brent, M., S. Murthy and A. Lundberg 1995. Discovering morphemic suffixes: A case study in minimum description length induction, paper presented at the Fifth International Workshop on AI and Statistics.
- Cartwright, T. and M. Brent 1997. Syntactic categorization in early language acquisition: Formalizing the role of distributional analysis, Cognition 63: 121-170.

- Cormen, T., C. Leiserson and R. Rivest 1990. Introduction to algorithms, Cambridge: MIT Press.
- de Marcken, C. 1996. Unsupervised language acquisition, MIT doctoral dissertation.
- Ellison, T. 1992. The machine learning of phonological structure, University of Western Australia doctoral dissertation.
- Emmorey, K. 1989. Auditory morphological priming in the lexicon, Language and Cognitive Processes 4: 73-92.
- Feldman, L. (ed.) 1995. Morphological aspects of language processing, Hillsdale: LEA.
- Goldsmith, J. *submitted*. Unsupervised learning of the morphology of a natural language.
- Gonnerman, L. and E. Andersen 2000. Graded semantic and phonological similarity effects in processing morphologically complex words, paper presented at the 9th International Morphology Meeting, Vienna.
- Greenberg, J. 1966. Some universals of grammar with particular reference to the order of meaningful elements, Cambridge: MIT Press.
- Grünwald, P. 1998. The minimum description length principle and reasoning under uncertainty, Amsterdam: ILLC.

- Harris, Z. 1955. From phoneme to morpheme, Language 31: 190-222.
- Hutchinson, A. 1994. Algorithmic learning, Oxford: Clarendon Press.
- Jusczyk, P. and R. Aslin 1995. Infants' detection of the sound patterns of words in fluent speech, Cognitive Psychology 29: 1-23.
- Kucera, H. and W. Francis 1967. Computational analysis of present-day American English, Providence, RI: Brown University Press.
- Laudanna, A. and C. Burani 1995. Distributional properties of derivational affixes: implications for processing, in Feldman 1995: 345-364.
- Li, M. and P. Vitányi 1997. An introduction to Kolmogorov complexity and its applications, New York: Springer.
- Marchand, H. 1969, The categories and types of present-day English word-formation: A synchronic-diachronic approach, Munich: Beck.
- Marslen-Wilson, W., L. Tyler, R. Waksler and L. Older 1994. Morphology and meaning in the English mental lexicon, Psychological Review 101: 3-33.
- McCarthy, J. and A. Prince 1986. Prosodic morphology, manuscript.
- McCarthy, J. and A. Prince 1993. Prosodic Morphology I, manuscript.

Mikheev, A. 1997. Automatic rule induction for unknown-word guessing, Computational Linguistics 24: 405-423.

Nespor, M. and I. Vogel 1986. Prosodic phonology, Dordrecht: Foris.

Nusbaum, H., D. Pisoni and C. Davis 1984. Sizing up the Hoosier Mental Lexicon: Measuring the familiarity of 20,000 words, Research on Spoken Language Processing PR 10: 357-376.

Pinker, S. 1984. Language learnability and language development, Cambridge: Harvard University Press.

Quirk, R., S. Greenbaum, G. Leech and J. Svartvik 1985. A comprehensive grammar of the English language, London: Longman.

Redington, M. and N. Chater 1998. Connectionist and statistical approaches to language acquisition: A distributional perspective, Language and Cognitive Processes 13: 129-191.

Rissanen, J. 1978. Modeling by shortest data description, Automatica 14: 456-471.

Roelofs, A. and H. Baayen *submitted*. Semantic transparency in producing polymorphemic words.

- Roman, S. 1996. Introduction to coding and information theory, New York: Springer.
- Saffran, J., R. Aslin and E. Newport 1996. Statistical learning by 8-month-old infants, Science 274: 1926-1928.
- Schreuder, R. and H. Baayen 1994. Prefix stripping re-revisited, Journal of Memory and Language 33: 357-375.
- Schreuder, R. and H. Baayen 1995. Modeling morphological processing, in Feldman 1995, 131-154.
- Seitz, P., L. Bernstein, E. Auer and M. MacEachern 1998. The PHLEX Database, Los Angeles: House Ear Institute.
- Smith, P. 1988. How to conduct experiments with morphologically complex words, Linguistics 26: 699-714.
- Spencer, A. 1991. Morphological theory, Oxford: Blackwell.
- Stolz, J. and L. Feldman 1995. The role of orthographic and semantic transparency of the base morpheme in morphological processing, in Feldman 1995: 109-129.
- Woods, A., P. Fletcher and A. Hughes 1986. Statistics in language studies, Cambridge: Cambridge University Press.