PARAMOR:

FROM PARADIGM STRUCTURE TO NATURAL LANGUAGE MORPHOLOGY INDUCTION

Christian Monson

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Thesis Committee

Jaime Carbonell	(Co-Chair)
Alon Lavie	(Co-Chair)
Lori Levin	
Ron Kaplan	(CSO at PowerSet)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

For Melinda

Abstract

Most of the world's natural languages have complex morphology. But the expense of building morphological analyzers by hand has prevented the development of morphological analysis systems for the large majority of languages. Unsupervised induction techniques, that learn from unannotated text data, can facilitate the development of computational morphology systems for new languages. Such unsupervised morphological analysis systems have been shown to help natural language processing tasks including speech recognition (Creutz, 2006) and information retrieval (Kurimo and Turunen, 2008). This thesis describes ParaMor, an unsupervised induction algorithm for learning morphological paradigms from large collections of words in any natural language. Paradigms are sets of mutually substitutable morphological operations that organize the inflectional morphology of natural languages. ParaMor focuses on the most common morphological process, suffixation.

ParaMor learns paradigms in a three-step algorithm. First, a recall-centric search scours a space of candidate partial paradigms for those which possibly model suffixes of true paradigms. Second, ParaMor merges selected candidates that appear to model portions of the same paradigm. And third, ParaMor discards those clusters which most likely do not model true paradigms. Based on the acquired paradigms, ParaMor then segments words into morphemes. ParaMor, by design, is particularly effective for inflectional mor-

phology, while other systems, such as Morfessor (Creutz, 2006), better identify derivational morphology. This thesis leverages the complementary strengths of ParaMor and Morfessor by adjoining the analyses from the two systems.

ParaMor and its combination with Morfessor participated in Morpho Challenge, a peer operated competition for morphology analysis systems (Kurimo, Turunen, and Varjokallio, 2008). The Morpho Challenge competitions held in 2007 and 2008 evaluated each system's morphological analyses in five languages, English, German, Finnish, Turkish, and Arabic. When ParaMor's morphological analyses are merged with those of Morfessor, the resulting morpheme recall in all five languages is higher than that of any system which competed in either year's Challenge; in Turkish, for example, ParaMor's recall of 52.1% is twice that of the next highest system. This strong recall leads to F_1 scores for morpheme identification above that of all systems in all languages but English.

Table of Contents

Acknowledgements

I am as lucky as a person can be to obtain a Ph.D. from the Language Technologies Institute (LTI) at Carnegie Mellon University. While I started this program with little more than an interest in languages and computation, the people I met at the LTI have brought natural language processing to life. From the unparalleled teaching and mentoring of professors like Pascual Masullo, Larry Wasserman, and Roni Rosenfeld I learned to appreciate, and begin to quantify natural language. And from fellow students including Katharina Probst, Ariadna Font Llitjós, and Erik Peterson I found that natural language processing is a worthwhile passion.

Still, whatever my desire for a degree in computational linguistics, this thesis never would have happened without the support, encouragement, and work of my three faculty advisors: Jaime Carbonell, Alon Lavie, and Lori Levin. Many people informed me that having three advisors was unworkable—one person telling you what to do is enough, let alone three! But in my particular situation, each advisor brought unique and indispensable strengths: Jaime's broad experience in cultivating and pruning a thesis project, Alon's hands-on detailed work on algorithmic specifics, and Lori's linguistic expertise were all invaluable. Without input from each of my advisors, ParaMor would not have been born.

Looking beyond the technical aspects, I was utterly unable to complete this thesis were it not for the support and patience of my family. Thank you Mom and Dad for being my final line of support; Thank you Kurt and Mitzi for not kicking me out of the house.

And without the encouragement of my wife, without her kind words and companionship, I would have thrown in the towel long ago. Thank you, Melinda, for standing behind me when the will to continue was beyond my grasp. And James, Liesl, and Edith: it was your needs and smiles that kept me trying.

— Christian

List of Figures

Chapter 1: Introduction

Most natural languages exhibit inflectional morphology, that is, the surface forms of words change to express syntactic features—I *run* vs. She *runs*. Handling the inflectional morphology of English in a natural language processing (NLP) system is fairly straightforward. The vast majority of lexical items in English have fewer than five surface forms. But English has a particularly sparse inflectional system. It is not at all unusual for a language to construct tens of unique inflected forms from a single lexeme. And many languages routinely inflect lexemes into hundreds, thousands, or even tens of thousands of unique forms! In these inflectional languages, computational systems as different as speech recognition (Creutz, 2006), machine translation (Goldwater and McClosky, 2005; Oflazer and El-Kahlout, 2007), and information retrieval (Kurimo and Turunen, 2008) improve with careful morphological analysis.

Computational approaches for analyzing inflectional morphology categorize into three groups. Morphology systems are either:

1. Hand-built,
2. Trained from examples of word forms correctly analyzed for morphology, or
3. Induced from morphologically unannotated text in an unsupervised fashion.

Presently, most computational applications take the first option, hand-encoding morphological facts. Unfortunately, manual description of morphology demands human expertise in a combination of linguistics and computation that is in short supply for many of the world's languages. The second option, training a morphological analyzer in a supervised fashion, suffers from a similar knowledge acquisition bottleneck: Morphologically analyzed training data must be specially prepared, i.e. segmented and labeled, by human experts. This thesis seeks to overcome the difficulties of knowledge acquisition through language independent unsupervised induction of morphological structure from readily available unannotated machine-readable natural language text.

1.1 The Structure of Morphology

Natural language morphology supplies many language independent structural regularities which unsupervised induction algorithms can exploit. This thesis intentionally leverages three such regularities to discover the morphology of individual languages. The first regularity is the *paradigmatic* opposition found in inflectional morphology. Paradigmatically opposed inflections are mutually substitutable and mutually exclusive. Take, for example, the Spanish word **hablar** '*to speak*', which belongs to the class of Spanish **ar**-verbs. Spanish **ar**-verbs inflect for the feature combination *2nd Person Present Indicative* with the suffix **as**, as in **hablas**; but mark *1st Person Present Indicative* with a mutually exclusive suffix **o**: **hablo**. The **o** suffix substitutes in for **as**, and no verb form can occur with both the **as** and the **o** suffixes simultaneously, ***hablaso**. Every set of paradigmatically opposed inflectional suffixes is said to fill a *paradigm*. In Spanish, the **as** and the **o** suffixes fill a portion of the verbal paradigm. Because of its direct appeal to paradigmatic opposition, the unsupervised morphology induction algorithm described in this thesis is dubbed *ParaMor*.

The second morphological regularity leveraged by ParaMor to uncover morphological structure is the *syntagmatic* relationship of lexemes. Natural languages with inflectional morphology invariably possess classes of lexemes that can each be inflected with the same set of paradigmatically opposed morphemes. These lexeme classes are in a syn-

tagmatic relationship. Returning to Spanish, *all* regular **ar**-verbs (**hablar**, **andar**, **cantar**, **saltar**, ...) use the **as** and **o** suffixes to mark *2nd Person Present Indicative* and *1st Person Present Indicative* respectively. Together, a particular set of paradigmatically opposed morphemes and the class of syntagmatically related stems adhering to that paradigmatic morpheme set forms an *inflection-class* of a paradigm of a language—in this case the **ar** inflection class of the Spanish verbal paradigm.

The third morphological regularity exploited by ParaMor follows from the paradigmatic-syntagmatic structure of natural language morphology. The repertoire of morphemes and stems in an inflection class constrains the *phoneme sequences* that occur within words. Specifically, while the phoneme sequence within a morpheme is restricted, a range of possible phonemes is likely at a morpheme boundary: A number of morphemes, each with possibly distinct initial phonemes, might follow a particular morpheme.

Spanish non-finite verbs illustrate paradigmatic opposition of morphemes, the syntagmatic relationship between stems, inflection classes, paradigms, and phoneme sequence constraints. In the schema of Spanish non-finite forms there are three paradigms, depicted as the three columns of Figure 1.1. The first paradigm marks the ***Type*** of a particular surface form. A Spanish verb can appear in exactly one of three ***Non-Finite Types***: as a *Past Participle*, as a *Present Participle*, or in the *Infinitive*. The three rows of the ***Type*** columns in Figure 1.1 represent the suffixes of these three paradigmatically opposed forms. If a Spanish verb occurs as a *Past Participle*, then the verb takes additional suffixes from two paradigms. First, an obligatory suffix marks ***Gender***: an **a** marks *Feminine*, an **o** *Masculine*. Following the suffix of the ***Gender*** paradigm, either a *Plural* suffix, **s**, appears or else there is no suffix at all. The lack of an explicit plural suffix marks *Singular*. The ***Gender*** and ***Number*** columns of Figure 1.1 represent these additional two paradigms. In the left-hand table the feature values for the ***Type***, ***Gender***, and ***Number*** paradigms are given. The right-hand table presents surface forms of suffixes realizing the corresponding feature values in the left-hand table. Spanish verbs which take the exact suffixes appearing in the right-hand table belong to the syntagmatic **ar** inflec-

Type	***Gender***	***Number***
Past Participle	*Feminine*	*Singular*
	Masculine	*Plural*
Present Participle		
Infinitive		

Type	***Gender***	***Number***
ad	a	Ø
	o	s
ando		
ar		

Figure 1.1 Left: A fragment of the morphological structure of Spanish verbs. There are three paradigms in this fragment. Each paradigm covers a single morphosyntactic category: first, Type; second, ***Gender***; and third, Number. Each of these three categories appears in a separate column; and features within one feature column, i.e. within one paradigm, are mutually exclusive. Right: The suffixes of the Spanish inflection class of ar verbs which fill the cells of the paradigms in the left-hand figure.

tion class of Spanish verbs. Appendix A gives a more complete summary of the paradigms and inflection classes of Spanish morphology.

To see the morphological structure of Figure 1.1 in action, we need a particular Spanish lexeme: a lexeme such as **administrar**, which translates as **to administer or manage**. The form **administrar** fills the *Infinitive* cell of the ***Type*** paradigm in Figure 1.1. Other forms of this lexeme fill other cells of Figure 1.1. The form filling the *Past Participle* cell of the ***Type*** paradigm, the *Feminine* cell of the ***Gender*** paradigm, and the *Plural* cell of the ***Number*** paradigm is **administradas**, a word which would refer to a group of *Feminine* ***Gender*** nouns under administration. Each column of Figure 1.1 truly constitutes a paradigm in that the cells of each column are mutually exclusive—there is no way for **administrar** (or any other Spanish lexeme) to appear simultaneously in the *Infinitive* and in a *Past Participle* form: ***administradasar**, ***administrardas**.

The phoneme sequence constraints implied by these Spanish paradigms emerge when considering the full set of surface forms for the lexeme **administrar**. Among the many inflected forms of **administrar** are *Past Participles* in all four combinations of ***Gender***

and ***Number***: **administrada**, **administradas**, **administrado**, and **administrados**; the *Present Participle* and *Infinitive* non-finite forms described in Figure 1.1: **administrando**, **administrar**; and the many finite forms such as the *1st Person Singular Indicative Present Tense* form **administro**. Figure 1.2 shows these forms (as in Johnson and Martin, 2003) laid out graphically as a finite state automaton (FSA). Each state in this FSA represents a character boundary, while the arcs are labeled with characters from the surface forms of the lexeme **administrar**. Morpheme-internal states are open circles in Figure 1.2, while states at word-internal morpheme boundaries are solid circles. Most morpheme-internal states have exactly one arc entering and one arc exiting. In contrast, states at morpheme boundaries tend to have multiple arcs entering or leaving, or both—the character (and phoneme) sequence is constrained within morpheme, but more free at morpheme boundaries.

This discussion of the paradigmatic, syntagmatic, and phoneme sequence structure of natural language morphology has intentionally simplified the true range of morphological phenomena. Three sources of complexity deserve particular mention. First, languages employ a wide variety of morphological processes. Among others, the processes of suf-

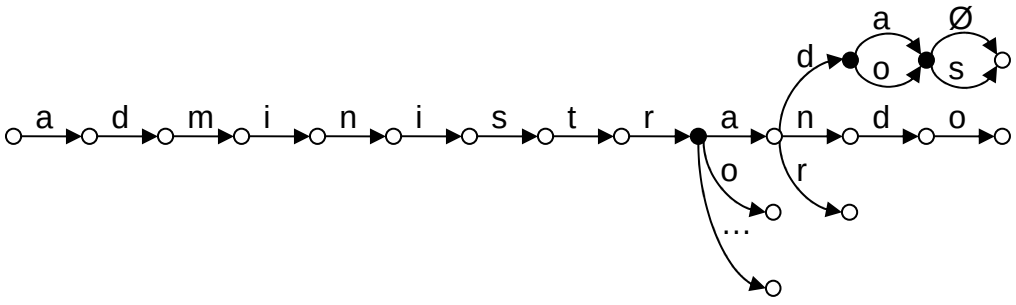

Figure 1.2: A Finite State Automaton (FSA) representing surface forms of the lexeme administrar. Arcs represent characters; States are character boundaries. States at morpheme boundaries typically have multiple arcs entering and/or exiting, while states at character boundaries internal to morpheme boundaries typically have a single entering and a single exiting arc.

fixation, prefixation, infixation, reduplication, and template filling all produce surface forms in some languages. Second, the application of word forming processes often triggers phonological (or orthographic) change. These phonological changes can obscure a straightforward concatenative treatment of morphology. And third, the morphological structure of a word can be inherently ambiguous—that is, a single surface form of a lexeme may have more than one legitimate morphological analysis.

Despite the complexity of morphology, this thesis holds that a large caste of morphological structures can be represented as paradigms of mutually substitutable substrings. In particular, sequences of affixes can be modeled by paradigm-like structures. Returning to the example of Spanish verbal paradigms in Figure 1.1, the ***Number*** paradigm on past participles can be captured by the alternating pair of strings **s** and **Ø**. Similarly, the ***Gender*** paradigm alternates between the strings **a** and **o**. Additionally, and crucially for this thesis, the ***Number*** and ***Gender*** paradigms combine to form an emergent cross-product paradigm of four alternating strings: **a**, **as**, **o**, and **os**. Carrying the cross-product further, the past participle endings alternate with the other verbal endings, both non-finite and finite, yielding a large cross-product paradigm-like structure of alternating strings which include: **ada**, **adas**, **ado**, **ados**, **ando**, **ar**, **o**, etc. These emergent cross-product paradigms each identify a single morpheme boundary within the larger paradigm structure of a language.

And with this brief introduction to morphology and paradigm structure we come to the formal claims of this thesis.

1.2 Thesis Claims

The algorithms and discoveries contained in this thesis automate the morphological analysis of natural language by inducing structures, in an unsupervised fashion, which closely correlate with inflectional paradigms. Additionally,

1. The discovered paradigmatic structures improve the word-to-morpheme segmentation performance of a state-of-the-art unsupervised morphology analysis system.
2. The unsupervised paradigm discovery and word segmentation algorithms improve this state-of-the-art performance for a diverse set of natural languages, including German, Turkish, Finnish, and Arabic.
3. The paradigm discovery and improved word segmentation algorithms are computationally tractable.
4. Augmenting a morphologically naïve information retrieval (IR) system with the induced morphological segmentations improves performance on an IR task. The IR improvements hold across a range of morphologically concatenative languages.

1.3 ParaMor: Paradigms across Morphology

The paradigmatic, syntagmatic, and phoneme sequence constraints of natural language allow ParaMor, the unsupervised morphology induction algorithm described in this thesis, to first reconstruct the morphological structure of a language, and to then deconstruct word forms of that language into constituent morphemes. The structures that ParaMor captures are sets of mutually replaceable word-final strings which closely model emergent paradigm cross-products—each paradigm cross-product identifying a single morpheme boundary in a set of words.

This dissertation focuses on identifying suffix morphology. Two facts support this choice. First, suffixation is a concatenative process and 86% of the world's languages use concatenative morphology to inflect lexemes (Dryer, 2008). Second, 64% of these concatenative languages are predominantly suffixing, while another 17% employ prefixation and suffixation about equally, and only 19% are predominantly prefixing. In any event, concentrating on suffixes is not a binding choice: the methods for suffix discovery detailed in this thesis can be straightforwardly adapted to prefixes, and generalizations could likely capture even non-concatenative morphological processes such as infixation.

To reconstruct the cross-products of the paradigms of a language, ParaMor defines and searches a network of paradigmatically and syntagmatically organized *schemes* of candidate suffixes and candidate stems. ParaMor's search algorithms are motivated by the paradigmatic, syntagmatic, and phoneme sequence constraints discussed in Section 1.1. Figure 1.3 depicts a portion of a morphology scheme network automatically derived from 100,000 words of the Brown Corpus of English (Francis, 1964). Each box in Figure 1.3 is a scheme, which lists in **bold** a set of candidate suffixes, or c-suffixes, together with a list, in *italics*, of candidate stems, or c-stems. Each of the c-suffixes in a scheme concatenates onto each of the c-stems in that scheme to form a word found in the input text. For instance, the scheme containing the c-suffix set **Ø.ed.es.ing**, where **Ø** signifies a null suffix, is derived from the words **address**, **addressed**, **addresses**, **addressing**, **reach**, **reached,** etc. In Figure 1.3, the two highlighted schemes, **Ø.ed.es.ing** and **e.ed.es.ing**, represent valid paradigmatically opposed sets of suffixes that head (orthographic) inflection classes of the English verbal paradigm. The other candidate

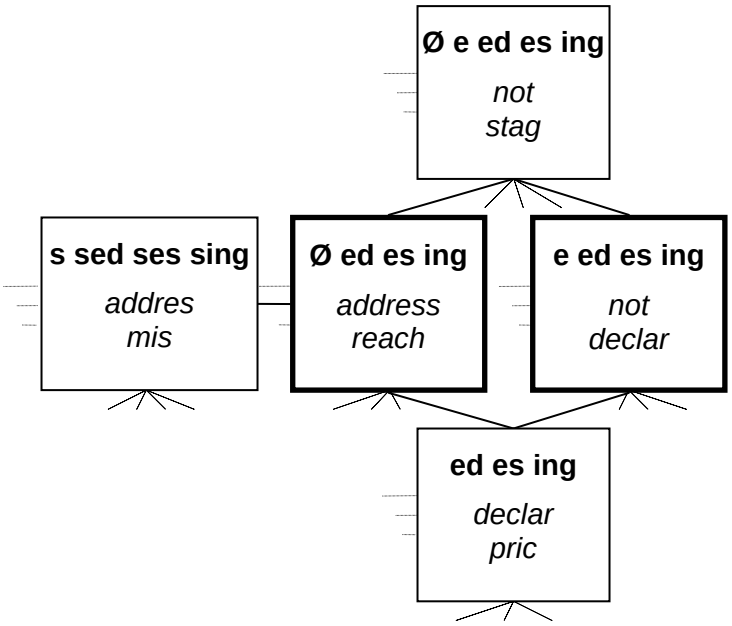

Figure 1.3: A portion of a morphology scheme network generated from 100,000 words of the Brown corpus of English (Francis, 1964). The two schemes which model complete verbal sub-classes are outlined in bold.

schemes in Figure 1.3 are wrong or incomplete. Crucially note, however, that as an unsupervised induction system ParaMor is not informed which schemes represent true paradigms and which do not—separating the good scheme models from the bad is exactly the task of ParaMor's paradigm induction algorithms.

Chapter 3 details the construction of morphology scheme networks over suffixes and describes a network search procedure that identifies schemes which contain in aggregate 91% of all Spanish inflectional suffixes when training over a corpus of 50,000 types. However, many of the initially selected schemes do not represent true paradigms; And of those that do represent paradigms, most capture only a portion of a complete paradigm. Hence, Chapter 4 describes algorithms to first merge candidate paradigm pieces into larger groups covering more of the affixes in a paradigm, and then to filter out those candidates which likely do not model true paradigms.

With a handle on the paradigm structures of a language, ParaMor uses the induced morphological knowledge to segment word forms into likely morphemes. Recall that, as

models of paradigm cross-products, each scheme models a single morpheme boundary in each surface form that contributes to that scheme. To segment a word form, ParaMor simply matches the c-suffixes of each discovered scheme against that word and proposes a single morpheme boundary at any match point. A pair of examples:

1. Assume ParaMor correctly identifies the English scheme **Ø.ed.es.ing** from Figure 1.3. When requested to segment the word **reaches**, ParaMor finds that the **es** c-suffix in the discovered scheme matches the word-final string **es** in **reaches**. Hence, ParaMor segments **reaches** as **reach +es**.
2. Since more than one paradigm cross-product may match a particular word, a word may be segmented at more than one position. The Spanish word **administradas** from Section 1.1 contains three suffixes, **ad**, **a**, and **s**. Presuming that ParaMor correctly identifies three separate schemes, one containing the cross-product c-suffix **adas**, one containing **as**, and one containing **s**, ParaMor will match in turn each of these c-suffixes against **administradas**, and will ultimately produce the correct segmentation: **administr +ad +a +s**.

To evaluate morphological segmentation performance, ParaMor competed in two years of the Morpho Challenge competition series (Monson et al., 2008a; 2008b). The Morpho Challenge competitions pit against one another algorithms designed to discover the morphological structure of natural languages from nothing more than raw text (Kurimo, Turunen, and Varjokallio, 2008). Unsupervised morphology induction systems were evaluated in two ways during the 2007 and 2008 Challenges. First, a linguistically motivated metric measured each system at the task of morpheme identification (Kurimo, Creutz, and Varjokallio, 2008; Kurimo and Varjokallio, 2008). Morpho Challenge 2007 evaluated systems' morpheme identification over four languages: English, German, Turkish, and Finnish; while the 2008 Challenge added Arabic. Second, in a Task-Based evaluation, the organizers of Morpho Challenge augmented an information retrieval (IR) system with the morphological segmentations that each system proposed and measured mean average precision of the relevance of returned documents (Kurimo, Creutz, and Turunen, 2007; Kurimo and Turunen, 2008).

As a stand-alone system, ParaMor performed on par with state-of-the-art unsupervised morphology induction systems at the Morpho Challenge competitions. Evaluated for F_1 at morpheme identification, in English ParaMor outperformed an already sophisticated reference induction algorithm, Morfessor-MAP (Creutz, 2006); placing third overall out of the eight participating algorithm families from the 2007 and 2008 competitions. In Turkish, ParaMor identified a significantly higher proportion of true Turkish morphemes than any other participating algorithm. This strong recall placed the solo ParaMor algorithm first in F_1 at morpheme identification for this language.

But ParaMor particularly shines when ParaMor's morphological analyses are adjoined to those of Morfessor-MAP. Where ParaMor focuses on discovering the paradigmatic structure of inflectional suffixes, the Morfessor algorithm identifies linear sequences of inflectional and derivational affixes—both prefixes and suffixes. With such complementary algorithms, it is not surprising that combining segmentations from the ParaMor and Morfessor systems improves performance over either algorithm alone. In all language tracks of the Challenge but English, the joint ParaMor-Morfessor system placed first at morpheme identification. In English the joint system moved to second. And in Turkish, morpheme identification of the ParaMor-Morfessor system is 13.5% higher absolute than the next best submitted system, excluding ParaMor alone. In the IR competition, which only covered English, German, and Finnish, the combined ParaMor-Morfessor system not only placed first in English and German, but also consistently outperformed, in all three languages, a baseline IR algorithm of no morphological analysis.

1.4 A Brief Reader's Guide

The remainder of this thesis is organized as follows: Chapter 2 situates the ParaMor algorithm in the field of prior work on unsupervised morphology induction. Chapters 3 and 4 present ParaMor's core paradigm discovery algorithms. Chapter 5 describes ParaMor's word segmentation models. And Chapter 6 details ParaMor's performance in the Morpho Challenge 2007 competition. Finally, Chapter 7 summarizes the contributions of

ParaMor and outlines future directions both specifically for ParaMor and more generally for the broader field of unsupervised morphology induction.

Chapter 2:
A Literature Review of ParaMor's Predecessors

The challenging task of unsupervised morphology induction has inspired a significant body of work. This chapter highlights unsupervised morphology systems that influenced the design of or that contrast with ParaMor, the morphology induction system described in this thesis. Two induction techniques have particularly impacted the development of ParaMor:

1. Finite State (FS) techniques, and
2. Minimum Description Length (MDL) techniques.

Sections 2.1 and 2.2 present, respectively, FS and MDL approaches to morphology induction, emphasizing their influence on ParaMor. Section 2.3 then describes several morphology induction systems which do not neatly fall in the FS or MDL camps but which are nevertheless relevant to the design of ParaMor. Finally, Section 2.4 synthesizes the findings of the earlier discussion.

2.1 Finite State Approaches

In 1955, Zellig Harris proposed to induce morphology in an unsupervised fashion by modeling morphology as a finite state automaton (FSA). In this FSA, the characters of each word label the transition arcs and, consequently, states in the automaton occur at character boundaries. Coming early in the procession of modern methods for morphology induction, Harris-style finite state techniques have been incorporated into a number of unsupervised morphology induction systems, ParaMor included. ParaMor draws on finite state techniques at two points within its algorithms. First, the finite state structure of morphology impacts ParaMor's initial organization of candidate partial paradigms into a search space (Section 3.1.2). And second, ParaMor identifies and removes the most unlikely initially selected candidate paradigms using finite state techniques (Section 4.4.2).

Three facts motivate finite state automata as appropriate models for unsupervised morphology induction. First, the topology of a morphological FSA captures phoneme sequence constraints in words. As was presented in Section 1.1, phoneme choice is usually constrained at character boundaries internal to a morpheme but is often more free at morpheme boundaries. In a morphological FSA, a state with a single incoming character arc and from which there is a single outgoing arc is likely internal to a morpheme, while a state with multiple incoming arcs and several competing outgoing branches likely occurs at a morpheme boundary. As described further in Section 2.1.1, it was this topological motivation that Harris exploited in his 1955 system, and that ParaMor draws on as well.

A second motivation for modeling morphological structure with finite state automata is that FSA succinctly capture the recurring nature of morphemes—a single sequence of states in an FSA can represent many individual instances, in many separate words, of a single morpheme. As described in Section 2.1.2 below, the morphology system of Altun and Johnson (2001) particularly builds on this succinctness property of finite state automata.

The third motivation for morphological FSA is theoretical: Most, if not all, morphological operations are finite state in computational complexity (Roark and Sproat, 2007). Indeed, state-of-the-art solutions for building morphological systems involve hand-

writing rules which are then automatically compiled into finite state networks (Beesley and Karttunen, 2003; Sproat 1997).

The next two sub-sections (2.1.1 and 2.1.2) describe specific unsupervised morphology induction systems which use finite state approaches. Section 2.1.1 begins with the simple finite state structures proposed by Harris, while Section 2.1.2 presents systems which allow more complex arbitrary finite state automata.

2.1.1 The Character Trie

Harris (1955; 1967) and later Hafer and Weiss (1974) were the first to propose and then implement finite state unsupervised morphology induction systems—although they may not have thought in finite state terms themselves. Harris (1955) outlines a morphology analysis algorithm which he motivated by appeal to the phoneme succession constraint properties of finite state structures. Harris' algorithm first builds character trees, or tries, over corpus utterances. Tries are deterministic, acyclic, but un-minimized FSA. In tries, Harris identifies those states for which the finite state transition function is defined for an unusually large number of characters in the alphabet. These branching states represent likely word and morpheme boundaries.

Although Harris only ever implemented his algorithm to segment words into morphemes, he originally intended his algorithms to segment sentences into words, as Harris (1967) notes, word-internal morpheme boundaries are much more difficult to detect with the trie algorithm. The comparative challenge of word-internal morpheme detection stems from the fact that phoneme variation at morpheme boundaries largely results from the interplay of a limited repertoire of paradigmatically opposed inflectional morphemes. In fact, as described in Section 1.1, word-internal phoneme sequence constraints can be viewed as the phonetic manifestation of the morphological phenomena of paradigmatic and syntagmatic variation.

Harris (1967), in a small scale mock-up, and Hafer and Weiss (1974), in more extensive quantitative experiments, report results at segmenting words into morphemes with the trie-based algorithm. Word-to-morpheme segmentation is an obvious measure of the

correctness of an induced model of morphology. And a number of natural language processing tasks, including machine translation, speech recognition, and information retrieval, could potentially benefit from an initial simplifying step of segmenting complex surface words into smaller recurring morphemes. Hafer and Weiss detail word segmentation performance when augmenting Harris' basic algorithm with a variety of heuristics for determining when the number of outgoing arcs is sufficient to postulate a morpheme boundary at a trie node. Hafer and Weiss measure recall and precision performance of each heuristic when supplied with a corpus of 6,200 word types. The variant which achieves the highest F_1 measure of 75.4%, from a precision of 81.8% and recall of 70.0%, combines results from both forward and backward tries and uses entropy to measure the branching factor of each node. Entropy captures not only the number of outgoing arcs but also the fraction of words that follow each arc.

A number of systems, many of which are discussed in depth later in this chapter, embed a Harris style trie algorithm as one step in a more complex process. Demberg (2007), Goldsmith (2001; 2006), Schone and Jurafsky (2000; 2001), and Déjean (1998) all use tries to construct initial lists of likely morphemes which they then process further. Bordag (2008) extracts morphemes from tries built over sets of words that occur in similar contexts. And Bernhard (2008) captures something akin to trie branching by calculating word-internal letter transition probabilities. Both the Bordag (2008) and the Bernhard (2008) systems competed strongly in the Morpho Challenge competition of 2007, alongside the unsupervised morphology induction system described in this thesis, ParaMor. Finally, the ParaMor system itself examines trie structures to identify likely morpheme boundaries. ParaMor builds local tries from the last characters of candidate stems which all occur in a corpus with the same set of candidate suffixes attached. Following Hafer and Weiss (1974), ParaMor measures the strength of candidate morpheme boundaries as the entropy of the relevant trie structures.

2.1.2 Unrestricted Finite State Automata

From tries it is not a far step to modeling morphology with more general finite state automata. A variety of methods have been proposed to induce FSA that closely model morphology. The ParaMor algorithm of this thesis, for example, models the morphology of a language with a non-deterministic finite state automaton containing a separate state to represent every set of word final strings which ends some set of word initial strings in a particular corpus (see Section 3.1.2).

In contrast, Johnson and Martin (2003) suggest identifying morpheme boundaries by examining properties of the minimal finite state automaton that exactly accepts the word types of a corpus. The minimal FSA can be generated straightforwardly from a Harris-style trie by collapsing trie states from which precisely the same set of strings is accepted. Like a trie, the minimal FSA is deterministic and acyclic, and the branching properties of its arcs encode phoneme succession constraints. In the minimal FSA, however, incoming arcs also provide morphological information. Where every state in a trie has exactly one incoming arc, each state, q, in the minimal FSA has a potentially separate incoming arc for each trie state which collapsed to form q. A state with two incoming arcs, for example, indicates that there are at least two strings for which exactly the same set of final strings completes word forms found in the corpus. Incoming arcs thus encode a rough guide to syntagmatic variation, see Section 1.1.

Johnson and Martin combine the syntagmatic information captured by incoming arcs with the phoneme sequence constraint information from outgoing arcs to segment the words of a corpus into morphemes at exactly:

1. Hub states—states which possess both more than one incoming arc and more than one outgoing arc, Figure 2.1, left.
2. The last state of stretched hubs—sequences of states where the first state has multiple incoming arcs and the last state has multiple outgoing arcs and the only available path leads from the first to the last state of the sequence, Figure 2.1, right.

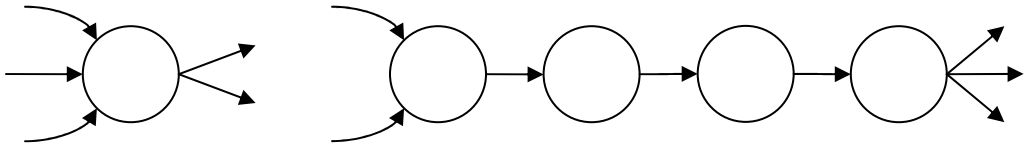

Figure 2.1: A hub, left, and a stretched hub, right, in a finite state automaton

Stretched hubs likely model the boundaries of a paradigmatically related set of morphemes, where each related morpheme begins (or ends) with the same character or character sequence. Johnson and Martin (2003) report that this simple Hub-Searching algorithm segments words into morphemes with an F_1 measure of 0.600, from a precision of 0.919 and a recall of 0.445, over the text of *Tom Sawyer*; which, according to Manning and Schütze (1999, p. 21), has 71,370 tokens and 8,018 types.

To improve segmentation recall, Johnson and Martin extend the Hub-Searching algorithm by introducing a morphologically motivated state merge operation. Merging states in a minimized FSA generalizes or increases the set of strings the FSA will accept. In this case, Johnson and Martin merge all states that are either accepting word final states, or that are likely morpheme boundary states by virtue of possessing at least two incoming arcs. This technique increases F_1 measure over the same *Tom Sawyer* corpus to 0.720, by bumping precision up slightly to 0.922 and significantly increasing recall to 0.590.

State merger is a broad technique for generalizing the language accepted by a FSA, used not only in finite state learning algorithms designed for natural language morphology, but also in techniques for inducing arbitrary FSA. Much research on FSA induction focuses on learning the grammars of artificial languages. Lang, Pearlmutter, and Price (1998) present a state-merging algorithm designed to learn large randomly generated deterministic FSA from positive and negative data. Lang, Pearlmutter, and Price (1998) also provides a brief overview of other work in FSA induction for artificial languages. Since natural language morphology is considerably more constrained than random FSA, and since natural languages typically only provide positive examples, work on inducing formally defined subsets of general finite state automata from positive data may be a bit

more relevant here. Work in constrained FSA induction includes Miclet (1980), who extends finite state k-tail induction, first introduced by Biermann and Feldman (1972), with a state merge operation. Similarly, Angluin (1982) presents an algorithm, also based on state merger, for the induction of k-reversible languages.

Altun and Johnson (2001) present a technique for FSA induction, again built on state merger, which is specifically motivated by natural language morphological structure. Altun and Johnson induce finite state grammars for the English auxiliary system and for Turkish Morphology. Their algorithm begins from the forward trie over a set of training examples. At each step the algorithm applies one of two merge operations. Either any two states, q_1 and q_2, are merged, which then forces their children to be recursively merged as well; or an ϵ-transition is introduced from q_1 to q_2. To keep the resulting FSA deterministic following an ϵ-transition insertion, for all characters a for which the FSA transition function is defined from both q_1 and q_2, the states to which *a* leads are merged, together with their children recursively.

Each arc (q,a) in the FSA induced by Altun and Johnson (2001) is associated with a probability, initialized to the fraction of words which follow the (q,a) arc. These arc probabilities define the probability of the set of training example strings. The training set probability is combined with the prior probability of the FSA to give a Bayesian description length for any training set-FSA pair. Altun and Johnson's greedy FSA search algorithm follows the minimum description length principle (MDL)—at each step of the algorithm, that state merge operation or ϵ-transition insertion operation is performed which most decreases the weighted sum of the log probability of the induced FSA and the log probability of the observed data given the FSA. If no operation results in a reduction in the description length, grammar induction ends.

Being primarily interested in inducing FSA, Altun and Johnson do not actively segment words into morphemes. Hence, quantitative comparison with other morphology induction work is difficult. Altun and Johnson do report the behavior of the negative log probability of Turkish test set data, and the number of learning steps taken by their algorithm, each as the training set size increases. Using these measures, they compare a version of their algorithm without ϵ-transition insertion to the version that includes this op-

eration. They find that their algorithm for FSA induction with ϵ-transitions achieves a lower negative log probability in less learning steps from fewer training examples.

2.2 MDL and Bayes' Rule: Balancing Data Length against Model Complexity

The minimum description length (MDL) principle employed by Altun and Johnson (2001) in a finite-state framework, as discussed in the previous section, has been used extensively in non-finite-state approaches to unsupervised morphology induction. The MDL principle is a model selection strategy which suggests to choose that model which minimizes the sum of:

1. The size of an efficient encoding of the model, and
2. The length of the data encoded using the model.

In morphology induction, the MDL principle measures the efficiency with which a model captures the recurring structures of morphology. Suppose an MDL morphology induction system identifies a candidate morphological structure, such as an inflectional morpheme or a paradigmatic set of morphemes. The MDL system will place the discovered morphological structure into the model exactly when the structure occurs sufficiently often in the data that it saves space overall to keep just one copy of the structure in the model and to then store pointers into the model each time the structure occurs in the data.

Although ParaMor, the unsupervised morphology induction system described in this thesis, directly measures neither the complexity of its models nor the length of the induction data given a model, ParaMor's design was, nevertheless, influenced by the MDL morphology induction systems described in this section. In particular, ParaMor implicitly aims to build compact models: The candidate paradigm schemes defined in Section 3.1.1 and the partial paradigm clusters of Section 4.3 both densely describe large swaths of the morphology of a language.

Closely related to the MDL principle is a particular application of Bayes' Rule from statistics. If d is a fixed set of data and m a morphology model ranging over a set of pos-

sible models, *M*, then the most probable model given the data is: $\text{argmax}_{m \in M} P(m \mid d)$. Applying Bayes' Rule to this expression yields:

$$\underset{m \in M}{\text{argmax}}\, P(m \mid d) = \underset{m \in M}{\text{argmax}}\, P(d \mid m) P(m),$$

And taking the negative of the logarithm of both sides gives:

$$\underset{m \in M}{\text{argmin}}\{-\log P(m \mid d)\} = \underset{m \in M}{\text{argmin}}\{[-\log P(d \mid m)] + [-\log P(m)]\}.$$

Reinterpreting this equation, the $[-\log P(m)]$ term is a reasonable measure of the length of a model, while $[-\log P(d \mid m)]$ expresses the length of the induction data given the model.

Despite the underlying close relationship between MDL and Bayes' Rule approaches to unsupervised morphology induction, a major division occurs in the published literature between systems that employ one or the other methodology. Sections 2.2.1 and 2.2.2 reflect this division: Section 2.2.1 describes unsupervised morphology systems that apply the MDL principle directly by devising an efficient encoding for a class of morphology models, while Section 2.2.2 presents systems that indirectly apply the MDL principle in defining a probability distribution over a set of models, and then invoking Bayes' Rule.

In addition to differing in their method for determining model and data length, the systems described in Sections 2.2.1 and 2.2.2 differ in the specifics of their search strategies. While the MDL principle can evaluate the strength of a model, it does not suggest how to *find* a good model. The specific search strategy a system uses is highly dependent on the details of the model family being explored. Section 2.1 presented search strategies used by morphology induction systems that model morphology with finite state automata. And now Sections 2.2.1 and 2.2.2 describe search strategies employed by non-finite state morphology systems. The details of system search strategies are relevant to this thesis work as Chapters 3 and 4 of this dissertation are largely devoted to the specifics of ParaMor's search algorithms. Similarities and contrasts with ParaMor's search procedures are highlighted both as individual systems are presented and also in summary in Section 2.4.

2.2.1 Measuring Morphology Models with Efficient Encodings

This survey of MDL-based unsupervised morphology induction systems begins with those that measure model length by explicitly defining an efficient model encoding. First to propose the MDL principle for morphology induction was Brent, Murthy, and Lundberg (1995; see also Brent, 1993). These authors use MDL to evaluate natural language morphology models of a simple, but elegant form. Their models describe a vocabulary as a set of three lists:

1. A list of stems
2. A list of suffixes, and
3. A list of valid (stem, suffix) pairs

Each of these three lists is efficiently encoded. The sum of the lengths of the first two encoded lists constitutes the model length, while the length of the third list yields the size of the data given the model. Consequently the sum of the lengths of all three encoded lists is the full description length to be minimized. As the morphology model in Brent, Murthy, and Lundberg (1995) only allows for pairs of stems and suffixes, each model can propose at most one morpheme boundary per word.

Using this list-model of morphology to describe a vocabulary of words, *V*, there are $\prod_{w \in V} |w|$ possible models—far too many to exhaustively explore. Hence, Brent, Murthy, and Lundberg (1995) describes a heuristic search procedure to greedily explore the model space. First, each word final string, *f*, in the corpus is ranked according to the ratio of the relative frequency of *f* divided by the relative frequencies of each character in *f*. Each word final string is then considered in turn, according to its heuristic rank, and added to the suffix list whenever doing so decreases the description length of the corpus. When no suffix can be added that reduces the description length further, the search considers removing a suffix from the suffix list. Suffixes are iteratively added and removed until description length can no longer be lowered.

To evaluate their method, Brent, Murthy, and Lundberg (1995) examine the list of suffixes found by the algorithm when supplied with English word form lexicons of various sizes. Any correctly identified inflectional or derivational suffix counts toward accu-

racy. Their highest accuracy results are obtained when the algorithm induces morphology from a lexicon of 2000 types: the algorithm hypothesizes twenty suffixes with an accuracy of 85%.

Baroni (2000; see also 2003) describes DDPL, an MDL inspired model of morphology induction similar to the Brent, Murthy, and Lundberg (1995) model. The DDPL model identifies prefixes instead of suffixes, uses a heuristic search strategy different from Brent, Murthy, and Lundberg (1995), and treats the MDL principle more as a guide than an inviolable tenet. But most importantly, Baroni conducts a rigorous empirical study showing that automatic morphological analyses found by DDPL correlate well with human judgments. He reports a Spearman correlation coefficient of 0.62 $(p << 0.001)$ for the correlation of average human morphological complexity rating to the DDPL analysis on a set of 300 potentially prefixed words of.

Goldsmith (2001; 2006), in a system called Linguistica, extends the promising results of MDL morphology induction by augmenting the basic model of Brent, Murthy, and Lundberg (1995) to incorporate the paradigmatic and syntagmatic structure of natural language morphology. As discussed in Chapter 1, natural language inflectional morphemes belong to paradigmatic sets where all the morphemes in a paradigmatic set are mutually exclusive. Similarly, natural language lexemes belong to syntagmatic classes where all lexemes in the same syntagmatic class can be inflected with the same set of paradigmatically opposed morphemes. While previous approaches to unsupervised morphology induction, including Déjean (1998), indirectly drew on the paradigmatic-syntagmatic structure of morphology, Goldsmith's Linguistica system was the first to intentionally model this important aspect of natural language morphological structure. The paradigm based algorithms of the ParaMor algorithm, as described in this thesis, were directly inspired by Goldsmith's success at unsupervised morphology induction when modeling the paradigm.

The Linguistica system models the paradigmatic and syntagmatic nature of natural language morphology by defining the *signature*. A Goldsmith signature is a pair of sets (T, F), T a set of st**e**ms and F a set of su**f**fixes, where T and F satisfy the following three conditions:

1. For any stem t in T and for any suffix f in F, $t.f$ must be a word in the vocabulary,
2. Each word in the vocabulary must be generated by exactly one signature, and
3. Each stem t occurs in the stem set of at most one signature

As in Brent, Murthy, and Lundberg (1995), a morphology model in Goldsmith's work consists of three lists. The first two are, as for Brent, a list of stems and a list of suffixes. But, instead of a list containing each valid stem-suffix pair, the third list in a Linguistica morphology consists of signatures. Replacing the list of all valid stem-suffix pairs with a list of signatures allows a signature model to potentially represent natural language morphology with a reduced description length. A description length decrease can occur because it takes less space to store a set of syntagmatically opposed stems with a set of paradigmatically opposed suffixes than it does to store the cross-product of the two sets.

Following the MDL principle, Goldsmith efficiently encodes each of the three lists that form a signature model; and the sum of the encoded lists is the model's description length. Notice that, just as for Brent, Murthy, and Lundberg (1995), Goldsmith's implemented morphology model can propose at most one morpheme boundary per word type—although Goldsmith (2001) does discuss an extension to handle multiple morpheme boundaries.

To find signature models, Goldsmith (2001; see also 2006) proposes several different search strategies. The most successful strategy seeds model selection with signatures derived from a Harris (1955) style trie algorithm. Then, a variety of heuristics suggest small changes to the seed model. Whenever a change results in a lower description length the change is accepted.

Goldsmith (2001) reports precision and recall results on segmenting 1,000 alphabetically consecutive words from:

1. The more than 30,000 unique word forms in the first 500,000 tokens of the Brown Corpus (Francis, 1964) of English: Precision: 0.860, Recall: 0.904, F_1: 0.881.
2. A corpus of 350,000 French tokens: Precision: 0.870, Recall: 0.890, F_1: 0.880.

Goldsmith (2001) also gives qualitative results for Italian, Spanish, and Latin suggesting that the best signatures in the discovered morphology models generally contain coherent sets of paradigmatically opposed suffixes and syntagmatically opposed stems.

2.2.2 Measuring Morphology Models with Probability Distributions

This sub-section contains an in-depth description of two morphology induction systems which exemplify the Bayes' Rule approach to unsupervised morphology induction: Snover (2002) and Creutz (2006) each build morphology induction systems by first defining a probability distribution over a family of morphology models and then searching for the most probable model. And, as described below, both the Snover system and that built by Creutz directly influenced the development of ParaMor.

The Morphology Induction System of Matthew Snover

Snover (2002; c.f.: Snover and Brent, 2002; Snover, Jarosz, and Brent, 2002; Snover and Brent, 2001) discusses a family of morphological induction systems which, like Goldsmith's Linguistica and like the ParaMor algorithm presented in this thesis, directly model the paradigmatic and syntagmatic structure of natural language morphology. But, where Goldsmith measures the quality of a morphological model as its encoded length, Snover invokes Bayes' Rule—defining a probability function over a space of morphology models and then searching for the highest probability model (see the introduction to Section 2.2).

Snover (2002) leverages paradigmatic and syntagmatic morphological structure both in his probabilistic morphology models and in the search strategies he employs. To define the probability of a model, Snover (2002) defines functions that assign probabilities to:

1. The stems in the model
2. The suffixes in the model
3. The assignment of stems to sets of suffixes called paradigms

Assuming independence, Snover defines the probability of a morphology model as the product of the probabilities of the stems, suffixes, and paradigms.

Like Goldsmith (2001; 2006), Snover only considers models of morphology where each word and each stem belong to exactly one paradigm. Hence, the third item in the above list is identical to Goldsmith's definition of a signature. Since Snover defines probabilities for exactly the same three items that Goldsmith computes description lengths for, the relationship of Snover's models to Goldsmith's is quite tight.

To find strong models of morphology Snover proposes two search procedures: *Hill Climbing Search* and *Directed Search*. Both strategies leverage the paradigmatic structure of language in defining data structures similar to the morphology networks proposed for this thesis in Chapter 3.

The Hill Climbing Search follows the same philosophy as the MDL based algorithms of Brent, Murthy, and Lundberg (1995) and Goldsmith (2001; 2006): At each step, Snover's Hill Climbing Search algorithm proposes a new morphology model, which is only accepted if it improves the model score—But in Snover's case, model score is probability. The Hill Climbing search uses an abstract suffix network defined by inclusion relations on sets of suffixes. Initially, the only network node to possess any stems is that node containing just the null suffix, **Ø**. All vocabulary items are placed in this **Ø** node. Each step of the Hill Climbing Search proposes adjusting the current morphological analysis by moving stems in batches to adjacent nodes that contain exactly one more or one fewer suffixes. At each step, that batch move is accepted which most improves the probability score.

Snover's probability model can only score morphology models where each word contains at most a single morpheme boundary. The Hill Climbing Search ensures this single boundary constraint is met by forcing each individual vocabulary word to only ever contribute to a single network node. Whenever a stem is moved to a new network node in violation of the single boundary constraint, the Hill Climbing Search simultaneously moves a compensating stem to a new node elsewhere in the network.

Snover's second search strategy, Directed Search, defines an instantiated suffix network where each node, or, in the terminology of Chapter 3, each scheme, in the network

inherently contains all the stems which form vocabulary words with each suffix in that node. In this fully instantiated network, a single word might contribute to two or more scheme-nodes which advocate different morpheme boundaries—a situation which Snover's probability model cannot evaluate. To build a global morphology model that the probability model *can* evaluate, the Directed Search algorithm visits every node in the suffix network and assigns a one-time probability score just for the segmentations suggested by that node. The Directed Search algorithm then constructs a globally consistent morphology model by first discarding all but the top *n* scoring nodes; and then, whenever two remaining nodes disagree on the segmentation of a word, accepting the segmentation from the better scoring node.

To evaluate the performance of his morphology induction algorithm, while avoiding the problems that ambiguous morpheme boundaries present to word segmentation, Snover (2002) defines a pair of evaluation metrics to separately: 1. Identify pairs of related words, and 2. Identify suffixes (where any suffix allomorph is accepted as correct). Helpfully, Snover (2002) supplies not only the results of his own algorithms using these metrics but also the results of Goldsmith's (2001) Linguistica. Snover (2002) achieves his best overall performance when using the Directed Search strategy to seed the Hill Climbing Search. This combination outperforms Linguistica on both the suffix identification metric as well as on the metric designed to identify pairs of related words, and does so for both English and Polish lexicons of up to 16,000 vocabulary items.

Hammarström (2006b; see also 2006a, 2007) presents a non-Bayes stemming algorithm that involves a paradigm search strategy that is closely related to Snover's. As in Snover's Directed Search, Hammarström defines a score for any set of candidate suffixes. But where Snover scores a suffix set according to a probability model that considers both the suffix set itself and the set of stems associated with that suffix set, Hammarström assigns a non-probabilistic score that is based on counts of stems that form words with pairs of suffixes from the set. Having defined an objective function, Hammarström's algorithm searches for a set of suffixes that scores highly. Hammarström's search algorithm moves from one set of suffixes to another in a fashion similar to Snover's Hill Climbing Search—by adding or subtracting a single suffix in a greedy fashion. Hammarström's

search algorithm is crucially different from Snover's in that Hammarström does not instantiate the full search space of suffix sets, but only builds those portions of the power set network that his algorithm directly visits.

Hammarström's paradigm search algorithm is one step in a larger system designed to stem words for information retrieval. To evaluate his system, Hammarström constructed sets of words that share a stem and sets of words that do not share a stem in four languages: Maori (an isolating language), English, Swedish, and Kuku Yalanji (a highly suffixing language). Hammarström finds that his algorithm is able to identify words that share a stem with accuracy above 90%.

ParaMor's search strategies, described in Chapter 3 of this thesis, were directly inspired by Snover's work and have much in common with Hammarström's. The most significant similarity between Snover's system and ParaMor concerns the search space they examine for paradigm models: the fully instantiated network that Snover constructs for his Directed Search is exactly the search space that ParaMor's initial paradigm search explores (see Chapter 3). The primary difference between ParaMor's search strategies and those of Snover is that, where Snover must ensure his final morphology model assigns at most a single morpheme boundary to each word, ParaMor intentionally permits individual words to participate in scheme-nodes that propose competing morpheme boundaries. By allowing more than one morpheme boundary per word, ParaMor can analyze surface forms that contain more than two morphemes.

Other contrasts between ParaMor's search strategies and those of Snover and of Hammarström include: The scheme nodes in the network that is defined for Snover's Directed Search algorithm, and that is implicit in Hammarström's work, are organized only by suffix set inclusion relations and so Snover's network is a subset of what Section 3.1.2 proposes for a general search space. Furthermore, the specifics of the strategies that Snover, Hammarström, and the ParaMor algorithm use to search the networks of candidate paradigms are radically different. Snover's Directed Search algorithm is an exhaustive search that evaluates each network node in isolation; Hammarström's search algorithm also assigns an intrinsic score to each node but searches the network in a greedy fashion; and ParaMor's search algorithm, Section 3.2, is a greedy algorithm like Ham-

marström's, but explores candidate paradigms by comparing each candidate to network neighbors. Finally, unlike Snover's Directed Search algorithm, neither Hammarström nor ParaMor actually instantiate the full suffix network. Instead, these algorithms dynamically construct only the needed portions of the full network. Dynamic network construction allows ParaMor to induce paradigms over a vocabulary nearly three times larger than the largest vocabulary Snover's system handles. Snover, Jarosz, and Brent (2002) discusses the possibility of using a beam or best-first search strategy to only search a subset of the full suffix network when identifying the initial best scoring nodes, but does not report results.

Mathias Creutz's Morfessor

In a series of papers culminating in a Ph.D. thesis (Creutz and Lagus, 2002; Creutz, 2003; Creutz and Lagus, 2004; Creutz, 2006) Mathias Creutz builds a probability-based morphology induction system he calls Morfessor that is tailored to agglutinative languages. In languages like Finnish, Creutz's native language, long sequences of suffixes agglutinate to form individual words. Morfessor's ability to analyze agglutinative structures inspired the ParaMor algorithm of this thesis to also account for suffix sequences—although the ParaMor and Morfessor algorithms use vastly different mechanisms to address agglutination.

To begin, Creutz and Lagus (2002) extend a basic MDL morphology model (Brent, Murthy, and Lundberg, 1995) to account for the morpheme sequences that are typical of agglutinative languages. The extension of Creutz and Lagus (2002) defines an MDL model that consists of just *two* parts:

1. A list of morphs, character strings that likely represent morphemes, where a morpheme could be a stem, prefix, or suffix; and
2. A list of morph sequences that result in valid word forms

By allowing each word to contain many morphs, Creutz and Lagus' Morfessor system neatly defies the single-suffix-per-word restriction found in so much work on unsupervised morphology induction.

The search space of agglutinative morphological models is large. Each word type can potentially contain as many morphemes as there are characters in that word. To rein in the number of models actually considered, Creutz and Lagus (2002) use a greedy search strategy where each word is recursively segmented into two strings, or morphs, as long as segmentation lowers the global description length.

Creutz (2003) improves Morfessor's morphology induction by moving from a traditional MDL framework, where models are evaluated according to their efficiently encoded size, to a probabilistic one, where model scores are computed according to a generative probability model (and implicitly relying on Bayes' theorem). Where Snover (2002) defines a paradigm-based probability model, Creutz (2003) probability model is tailored for agglutinative morphology models, and does not consider paradigm structure. Creutz (2003) does not modify the greedy recursive search strategy Morfessor uses to search for a strong morphology model.

Finally, Creutz and Lagus (2004) refine the agglutinative morphology models that Morfessor selects by introducing three categories: prefix, stem, and suffix. The Morfessor system assigns every identified morph to each of these three categories with a certain probability. Creutz and Lagus then define a simple Hidden Markov Model (HMM) that describes the probability of outputting any possible sequence of morphs that conforms to the regular expression: `(prefix* stem suffix*)+`.

The morphology models described in this series of three papers each quantitatively improves upon the previous. Creutz and Lagus (2004) compares the full Morfessor system that uses morph categories to Goldsmith's Linguistica using precision and recall scores for word-to-morpheme segmentation. They report results over both English and Finnish with a variety of corpus sizes. When the input is a Finnish corpus of 250,000 tokens or 65,000 types, the Morfessor category model achieves an F_1 of 0.64 from a precision of 0.81 and a recall of 0.53, while Linguistica only attains an F_1 of 0.56 from a precision of 0.76 and a recall of 0.44. On the other hand, Linguistica does not fare so poorly on a similarly sized corpus of English (250,000 tokens, 20,000 types): Creutz and Lagus' Morfessor Category model: F_1: 0.73, precision: 0.70, recall: 0.77; Linguistica: F_1: 0.74, precision: 0.68, recall: 0.80.

2.3 Other Approaches to Unsupervised Morphology Induction

Sections 2.1 and 2.2 presented unsupervised morphology induction systems which directly influenced the design of the ParaMor algorithm in this thesis. This section steps back for a moment, examining systems that either take a radically different approach to unsupervised morphology induction or that solve independent but closely related problems.

Let's begin with Schone and Jurafsky (2000), who pursue a very different approach to unsupervised morphology induction from ParaMor. Schone and Jurafsky notice that in addition to being orthographically similar, morphologically related words are similar semantically. Their algorithm first acquires a list of pairs of potential morphological variants (PPMV's) by identifying, in a trie, pairs of vocabulary words that share an initial string. This string similarity technique was earlier used in the context of unsupervised morphology induction by Jacquemin (1997) and Gaussier (1999). Schone and Jurafsky apply latent semantic analysis (LSA) to score each PPMV with a semantic distance. Pairs measuring a small distance, those pairs whose potential variants tend to occur where a neighborhood of the nearest hundred words contains similar counts of individual high-frequency forms, are then proposed as true morphological variants of one another. In later work, Schone and Jurafsky (2001) extend their technique to identify not only suffixes but also prefixes and circumfixes. Schone and Jurafsky (2001) report that their full algorithm significantly outperforms Goldsmith's Linguistica at identifying sets of morphologically related words.

Following a logic similar to Schone and Jurafsky (2000; 2001), Baroni, Matiasek, and Trost (2002) marry a mutual information derived semantic-based similarity measure with an orthographic similarity measure to induce the citation forms of inflected words. And in the information retrieval literature, where stemming algorithms share much in common with morphological analysis, Xu and Croft (1998) describe an unsupervised stemmer induction algorithm that also has a flavor similar to Schone and Jurafsky's morphology induction system. Xu and Croft start from sets of word forms that, because they share the same initial three characters, likely share a stem. They then measure the signifi-

cance of word form co-occurrence in windows of text. Word forms from the same initial string set that co-occur unusually often are placed in the same stem class.

Finally, this discussion concludes with a look at some work which begins to move beyond simple word-to-morpheme segmentation. All of the unsupervised morphology induction systems presented thus far, including the ParaMor algorithm of this thesis, cannot generalize beyond the word forms found in the induction corpus to hypothesize unseen inflected words. Consequently, the induction algorithms described in this chapter are suitable for morphological analysis but not for generation. Chan (2006) seeks to close this generation gap. Using Latent Dirichlet Allocation, a dimensionality reduction technique, Chan groups suffixes into paradigms and probabilistically assigns stems to those paradigms. Over preprocessed English and Spanish texts, where each individual word has been morphologically analyzed with the correct segmentation and suffix label, Chan's algorithm can perfectly reconstruct the suffix groupings of morphological paradigms.

In other work that looks beyond word segmentation, Wicentowski and Yarowsky (Wicentowski, 2002; Yarowsky and Wicentowski, 2000; Yarowsky, Ngai, and Wicentowski, 2001) iteratively train a probabilistic model that identifies the citation form of an inflected word from several individually unreliable measures including: relative frequency ratios of stems and inflected word forms, contextual similarity of the candidate forms, the orthographic similarity of the forms as measured by a weighted Levenstein distance, and in Yarowsky, Ngai, and Wicentowski (2001) a translingual bridge similarity induced from a clever application of statistical machine translation style word alignment probabilities.

Wicentowski's work stands out in the unsupervised morphology induction literature for explicitly modeling two important but rarely addressed morphological phenomena: non-concatenative morphology and morphophonology. Wicentowski and Yarowsky's probabilistic model explicitly allows for non-concatenative stem-internal vowel changes as well as phonologic stem-boundary alterations that may occur when either a prefix or a suffix attaches to a stem.

More recent work has taken up the thread of both non-concatenative morphological processes and morphophonology. Xanthos (2007) builds an MDL-based morphology in-

duction system specifically designed to identify the interstitial root-and-pattern morphology of Semitic languages such as Arabic and Hebrew. While in other work, Demberg (2007) takes a close look at morphophonology. Demberg's system extracts, from forward and backward tries, suffix clusters similar to Goldsmith's signatures. Demberg then measures the edit distance between suffixes in a cluster and notes that suffixes with a small distance are often due to morphophonologic changes.

Probst (2003) pursues another less-studied aspect of unsupervised morphology induction: Unlike any other recent proposal, Probst's unsupervised morphology induction system can assign morphosyntactic features (i.e. ***Number***, ***Person***, ***Tense***, etc.) to induced morphemes. Like Yarowsky, Ngai, and Wicentowski (2001), Probst uses machine translation word alignment probabilities to develop a morphological induction system for the second language in the translation pair. Probst draws information on morphosyntactic features from a lexicon that includes morphosyntactic feature information for the first language in the pair and projects this feature information onto the second language. Probst's work, as well as Chan's and that of Yarowsky and Wicentowski's, take small steps outside the framework of unsupervised morphology induction by assuming access to limited linguistic information.

2.4 Discussion of Related Work

ParaMor, the unsupervised morphology induction system described in this thesis, fuses ideas from the unsupervised morphology induction approaches presented in the previous three sections and then builds on them. Like Goldsmith's (2001; 2006) Linguistica, ParaMor intentionally models paradigm structure; and like Creutz's (2006) Morfessor, ParaMor addresses agglutinative sequences of suffixes; but unlike either, ParaMor tackles *agglutinative sequences* of *paradigmatic* morphemes. Similarly, ParaMor's morpheme search space is a synthesis of the paradigmatic/syntagmatic morphology structure modeled by Snover (2002) on the one hand, and the finite state phoneme sequence description of morphology (Harris, 1955; Johnson and Martin, 2003) on the other.

In a related vein, while ParaMor does not model morphology in a minimum description length (MDL) framework as Brent, Murthy, and Lundberg (1995), Baroni (2000),

and Goldsmith (2001; 2006) do, and while ParaMor does not define a probabilistic model of morphology as Snover (2002) and Creutz (2003) do, ParaMor does acknowledge the premise of these systems that a compact morphology model is desirable. Indeed, the basic building blocks of the network search space defined in Chapter 3, schemes, are a compact representation of morphological structure.

The work proposed for this thesis does not directly extend every promising approach to unsupervised morphology described in the first three sections of this chapter. For example, ParaMor only models suffixing morphology, and does not follow Baroni (2000) or Schone and Jurafsky (2001) in addressing prefixes, or Xanthos (2007) in modeling non-concatenative processes. ParaMor also postpones for future work the modeling of morphophonologic change, as considered in Wicentowski (2002). Finally, the progress made by Schone and Jurafsky (2000), Wicentowski (2002), and others on identifying morphologically related word forms by analyzing their semantic and syntactic distance is both interesting and promising. While this thesis does not pursue this direction, integrating semantic and syntactic information into ParaMor's existing algorithms is an exciting path for future work on unsupervised morphology induction.

Clearly, ParaMor owes a great debt to previously proposed ideas for unsupervised morphology induction. Without the example of previously built systems, ParaMor would not have been able to spearhead new work on topics including a comprehensive search space of candidate paradigms, innovative measures to search that space, or how to transform candidate paradigms that individually model a single morpheme boundary into agglutinative analyses consisting of multiple morphemes. As the next chapters will show, biasing the morphology induction problem with the paradigmatic, syntagmatic, and phoneme sequence structure inherent in natural language morphology is the powerful leg-up needed for an unsupervised solution to the morphology induction problem.

Chapter 3: Paradigm Identification with ParaMor

This thesis describes and motivates ParaMor, an unsupervised morphology induction algorithm. To uncover the organization of morphology within a specific language, ParaMor leverages paradigms as the language independent structure of natural language morphology. In particular ParaMor exploits paradigmatic and syntagmatic relationships which hold cross-linguistically among affixes and lexical stems respectively. The paradigmatic and syntagmatic properties of natural language morphology were presented in some detail in Section 1.1. Briefly, an inflectional paradigm in morphology consists of:

1. A set of mutually substitutable, or paradigmatically related, affixes, and
2. A set of syntagmatically related stems which *all* inflect with the affixes in 1.

This chapter describes and motivates ParaMor's unsupervised strategies to initially isolate likely partial models of paradigmatic structures. These initial paradigm models are partial in two ways. First, most of ParaMor's initial models will only describe a subset of the affixes in any particular paradigm. The clustering algorithm described in Chapter 4 is

specifically designed to join initial models that describe subsets of the same underlying paradigm. Second, while many natural languages combine several morphemes to form a single word, each of ParaMor's paradigm models can only describe a single morpheme boundary lying between a pair of morphemes. The word-to-morpheme segmentation algorithm described in Chapter 5 will recombine ParaMor's individual paradigm models to segment a single word into more than two morphemes.

ParaMor's paradigm discovery algorithms begin with the definition of a search space over natural groupings, or schemes, of paradigmatically and syntagmatically related candidate suffixes and candidate stems (Section 3.1). As Chapter 1 motivated, this thesis focuses on identifying suffix morphology. Then, with a clear view of the search space, ParaMor searches for those schemes which most likely model the paradigm structure of true suffixes within the language, (Section 3.2).

3.1 A Search Space of Morphological Schemes

3.1.1 Schemes

The constraints implied by the paradigmatic and syntagmatic structure of natural language can organize candidate suffixes and stems into the building blocks of a search space in which to identify language specific models of paradigms. This thesis names these building blocks *schemes*, as each is "an orderly combination of related parts" (The American Heritage® Dictionary, 2000).

Scheme organization begins by proposing candidate morpheme boundaries at every character boundary in every word form in a corpus vocabulary. The scheme-based approach to unsupervised morphology induction is designed to work on orthographies which at least loosely code each phoneme in each word with a separate character; and, as such, ParaMor's induction approach does not extend to the standard writing systems of many East Asian languages, including Chinese and Japanese.

Languages often mark a specific feature value combination by explicitly *not* changing the form of a stem. One way to describe these empty changes is with an attached null

affix, **Ø**. To account for null suffixes, the set of candidate morpheme boundaries the ParaMor algorithm proposes include the boundary after the final character in each word form. Since this thesis focuses on identifying suffixes, it is assumed that each word form contains a stem of at least one character. Hence, the boundary before the first character of each word form is not considered a candidate morpheme boundary.

Call each string that occurs before a candidate morpheme boundary a *candidate stem* or *c-stem*, and each string after a proposed boundary a *c-suffix*. Let V be a set of strings—a **v**ocabulary of word types. Let T^V be the set of all c-s**t**ems generated from the vocabulary and F^V be the corresponding set of all c-su**f**fixes. With these preliminaries, define a *s**c**heme* C to be a pair of sets of strings (T_C, F_C) satisfying the following four conditions:

1. $T_C \subset T^V$, called the adherents of C
2. $F_C \subset F^V$, called the exponents of C
3. $\forall t_C \in T_C, \forall f_C \in F_C, t_C.f_C \in V$
4. $\forall t^V \in T^V$, if $\forall f_C \in F_C, t^V.f_C \in V$ then $t^V \in T_C$

The first three conditions require each of the syntagmatically related c-stems in a scheme to combine with each of the paradigmatic c-suffixes of that scheme to form valid vocabulary words. The fourth condition forces a scheme to contain *all* of the syntagmatic c-stems that form valid words with each of the paradigmatic c-suffixes in that scheme. The number of c-stems in T_C is the *adherent size* of C, and the number of c-suffixes in F_C is the *paradigmatic level* of C.

At this point, it is worth noting two facts about the definition of a scheme. First, the definition of a scheme allows a single word to contribute to two or more schemes which divide the word differently into c-stem and c-suffix. Although seemingly inoccuous, the fact that different schemes, derived from the same vocabulary, can model different morpheme boundaries in the same word will be the key to ParaMor's ability to segment words into sequences of agglutinative morphemes, see Chapter 5.

The second fact of note is that the fourth condition in the definition of a scheme is intentionally asymmetric. Condition four only requires a scheme to contain all the *c-stems*

that combine with all the *c-suffixes* in the scheme, but *not* necessarily all the *c-suffixes* that combine with all the *c-stems* of a scheme.

This asymmetry in the scheme definition has two far-reaching consequences. First, the asymmetry creates a unique scheme for any and every set of c-suffixes derived from a vocabulary. This one-to-one correspondence between each scheme, *C*, and the set of c-suffixes in *C* allows unambiguous reference to a scheme by its set of c-suffixes. But more importantly, it is the one-to-one correspondence between schemes and sets of c-suffixes that permits, as discussed in Section 3.1.2, the organization of schemes into a search space in which ParaMor identifies initial candidate paradigms.

The second consequence of the asymmetry in the definition of a scheme will impact the search algorithm that ParaMor uses to find likely paradigms (see Section 3.2). Specifically, the assymmetric definition of a scheme implies that any stem that is part of a scheme, *C*, must also be in every scheme that is built from a subset of the suffixes in *C*.

Schemes: An Example

To better understand how, in practice, schemes succinctly capture both the paradigmatic and syntagmatic regularities of natural language morphology, let us look at a few illustrative schemes from a small example. Each box in Figure 3.1 contains a scheme derived from one or more of the word forms listed at the top of the figure. The vocabulary of Figure 3.1 mimics the vocabulary of a text corpus from a highly inflected language where few, if any, lexemes will occur in the complete set of possible surface forms. Specifically, the vocabulary of Figure 3.1 lacks the surface form **blaming** of the lexeme **BLAME**, **solved** of the lexeme **SOLVE**, and the root form **roam** of the lexeme **ROAM**.

Proposing, as ParaMor's procedure does, morpheme boundaries at every character boundary in every word form necessarily produces many ridiculous schemes such as the paradigmatic level three scheme **ame.ames.amed**, from the word forms **blame**, **blames**, and **blamed** and the c-stem **bl**. Dispersed among the incorrect schemes, however, are also schemes that seem very reasonable, such as **Ø.s**, from the c-stems **blame** and **solve**.

Schemes are intended to capture both the paradigmatic and syntagmatic structure of morphology. For example, the fact that the paradigmatically related c-suffixes **Ø**, and **s**

Vocabulary:

blame		solve
blames	roams	solves
blamed	roamed	
	roaming	solving

lame lames lamed *b*	**ame ames amed** *bl*	**me mes med** *bla*	**e es ed** *blam*	**Ø s d** *blame*	**oams oamed oaming** *r*	
lame lames *b*	**ame ames** *bl*	**me mes** *bla*	**e es** *blam solv*	**Ø s** *blame solve*	**olve olves olving** *s*	
lame lamed *b*	**ame amed** *bl*	**me med** *bla*	**e ed** *blam*	**Ø d** *blame*	…	
lames lamed *b*	**ames amed** *bl*	**mes med** *bla*	**es ed** *blam*	**s d** *blame*	**s ed ing** *roam*	**e es ing** *solv*
lame *b*	**ame** *bl*	**me** *bla*	**e** *blam solv*		**s ed** *roam*	**e ing** *solv*
lames *b*	**ames** *bl*	**mes** *bla*	**es** *blam solv*	**s** *blame roam solve*	**s ing** *roam*	**es ing** *solv*
lamed *b*	**ames** *bl ro*	**med** *bla roa*	**ed** *blam roam*	**d** *blame roame*	**ed ing** *roam*	**ng** *roami solvi*
Ø *blame blames blamed roams roamed roaming solve solves solving*					**ing** *roam solv*	**g** *roamin solvin*

Figure 3.1: Some of the schemes, arranged in a systematic but arbitrary order, derived from a small vocabulary (top). Each scheme is specified as a space delimited set of c-suffix exponents in **bold** above a space delimited set of c-stem adherents in *italics*

each concatenate onto both of the syntagmatically related c-stems **solve** and **blame** suggests that the c-stem **blame** should be an adherent of the scheme **Ø.s**—just as the given definition of a scheme requires. But note that if schemes were limited to containing c-stems that concatenate *only* the c-suffixes in that scheme, then the c-stem **blame** could not be part of the **Ø.s** scheme, as the c-stem **blame** also occurs in the form **blamed**.

Before moving on, observe two additional intricacies of scheme generation. First, while the scheme **Ø.s** arises from the pairs of surface forms (**blame**, **blames**) and (**solve**, **solves**), there is no way for the form **roams** to contribute to the **Ø.s** scheme because the surface form **roam** is not in this vocabulary. Second, as a result of English spelling rules,

the scheme **s.d**, generated from the pair of surface forms (**blames**, **blamed**), is separate from the scheme **s.ed**, generated from the pair of surface forms (**roams**, **roamed**).

One Scheme, One Morpheme Boundary

Behind each scheme, *C*, is a set of *licensing word forms*, *W*, which contribute c-stems and c-suffixes to *C*. Each c-suffix in *C* which matches the tail of a licensing word, $w \in W$, segments *w* in exactly one position. Although it is theoretically possible for more than one c-suffix of *C* to match a particular licensing word form[1], in empirical schemes that arise from natural language text, each word *w* typically matches just one c-suffix in *C*. Hence, a naturally occurring scheme, *C*, models only a single morpheme boundary in each word *w* that licenses *C*.

But words in natural language may possess more than one morpheme boundary. In Spanish, as discussed in Section 1.1 of the thesis introduction, *Past Participles* of verbs contain either two or three morpheme boundaries: one boundary after the verb stem and before the *Past Participle* marker; one boundary between the *Past Participle* marker and the ***Gender*** suffix; and, if the *Past Participle* is plural, a final morpheme boundary between the ***Gender*** suffix and the ***Plural*** suffix, **s**; see Figure 1.1 on p. 18.

Although a single scheme models just a single morpheme boundary in a particular word, together separate schemes can model all the morpheme boundaries of a word. In Spanish *Past Participles*, the **Ø.s** scheme can model the paradigm for the optional ***Number*** suffix, while another scheme, **a.as.o.os,** models the *cross-product* of the ***Gender*** and ***Number*** paradigms, and yet another scheme, which includes the c-suffixes **ada**, **adas**, **ado**, and **ados**, models the cross-product of three paradigms: ***Verbal Form***, ***Gender***, and ***Number***. In one particular corpus containing 50,000 unique word types of newswire Spanish, there are 5501 c-stem adherents of the **Ø.s** scheme, 892 adherents of the **a.as.o.os** scheme, and 302 c-stems in the **ada.adas.ado.ados** scheme. Notice that it

---

[1]Consider, for example, a hypothetical corpus vocabulary containing just three words: **ac**, **abc**, and **abbc**. These three words could give rise to the scheme **bc.c** containing the stems **a** and **ab**. Reconcatenating the stems and suffixes in this hypothetical scheme gives the boundary annotated forms: **a+bc**, **a+c**, **ab+bc**, and **ab+c**—but **a+bc** and **ab+c** are different segmentations of the same licensing word, **abc**.

is only when a scheme models the final morpheme boundary of the scheme's licensing words that a scheme can model an individual traditional paradigm. When a scheme captures morpheme boundaries that are not word final, then the scheme's c-suffixes encapsulate the cross-product of two or more traditional morphemes.

Although the only traditional paradigms that schemes can directly model are word-final, schemes still provide this thesis with a strong model of natural language morphology for two reasons. First, as noted in the previous paragraph, while any particular scheme cannot by itself model a single word-internal paradigm, in concert, schemes can identify agglutinative sequences of morphemes. Second, the cross-product structures that are captured by schemes retain the paradigmatic and syntagmatic properties of mutual substitutability and mutual exclusivity that define traditional inflectional paradigms. Just as suffixes in a traditional paradigm can be interchanged on adherent stems to form surface forms, the c-suffixes of a cross-product scheme can be mutually substituted to form valid surface forms with the adherent c-stems in the scheme: A traditional paradigm might replace the final **s** in the Spanish word form **administradas** with **Ø** to yield the valid form **administrada**, while a scheme might suggest replacing the final cross-product c-suffix **as** with **o** to form the grammatical Spanish word form **administrado**.

Ultimately, restricting each scheme to model a single morpheme boundary is computationally simpler than a model which allows more than one morpheme boundary per modeling unit. And, as Chapters 5 and 6 show, algorithms built on the simple scheme allow ParaMor to effectively analyze the morphology of even highly agglutinative languages such as Finnish and Turkish.

3.1.2 Scheme Networks

Looking at Figure 3.1, it is clear there is structure among the various schemes. At least two types of relations hold between schemes. First, hierarchically, the c-suffixes of one scheme may be a superset of the c-suffixes of another scheme. For example the c-suffixes in the scheme **e.es.ed** are a superset of the c-suffixes in the scheme **e.ed**. Second, cutting across this hierarchical structure are schemes which propose different mor-

pheme boundaries within a set of word forms. Compare the schemes **me.mes.med** and **e.es.ed**; each is derived from exactly the triple of word forms **blame**, **blames**, and **blamed**, but differ in the placement of the hypothesized morpheme boundary. Taken together the hierarchical c-suffix set inclusion relations and the morpheme boundary relations impose a lattice structure on the space of schemes.

Figure 3.2 diagrams a scheme lattice over an interesting subset of the columns of Figure 3.1. Each box in Figure 3.2 is a scheme, where, just as in Figure 3.1, the c-suffix exponents are in **bold** and the c-stem adherents are in *italics*. Hierarchical c-suffix set inclusion links, represented by solid lines (———), connect a scheme to often more than one parent and more than one child. The empty scheme (not pictured in Figure 3.2), containing no c-suffixes and no c-stems, can be considered the child of all schemes of paradigmatic level 1 (including the **Ø** scheme). Horizontal morpheme boundary links, dashed lines (------), connect schemes which hypothesize morpheme boundaries which differ by a single character. In most schemes of Figure 3.2, the c-suffixes all begin with the same character. When all c-suffixes begin with the same character, there can be just a single morpheme boundary link leading to the right. Similarly, a morphology scheme network contains a separate leftward link from a particular scheme for each character which ends some c-stem in that scheme. The only scheme with explicit multiple left links in Figure 3.2 is **Ø**, which has depicted left links to the schemes **e**, **s**, and **d**. A number of left links emanating from the schemes in Figure 3.2 are not shown; among other links absent from the figure is the left link from the scheme **e.es** that would lead to the scheme **ve.ves** with the adherent **sol**. Section 4.4.2 defines morpheme boundary links more explicitly.

The scheme network of Figure 3.2 is a portion of the search space in which the ParaMor unsupervised morphology induction algorithm would operate given the small vocabulary of Figure 3.1. But it is important to note that the ParaMor algorithm would not necessarily instantiate every scheme node in Figure 3.2. The number of schemes in a full scheme network is the size of the powerset of the set of all word-final strings in the vocabulary! Much too large to exhaustively search or build for even moderately sized corpora. As Sections 3.2.2 and 3.2.4 discuss, ParaMor dynamically creates just those por-

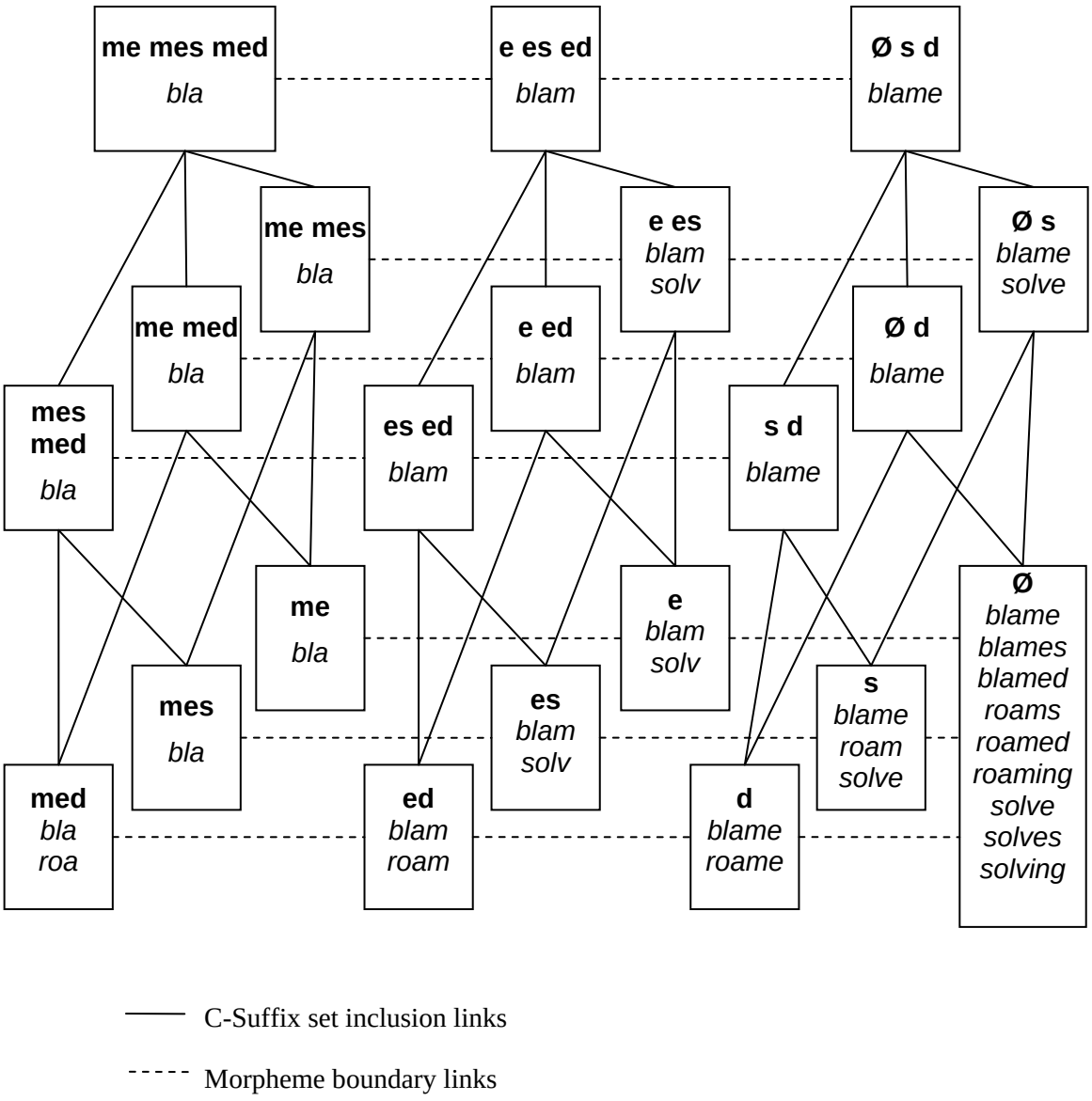

Figure 3.2: A portion of a morphology scheme network generated from the small example vocabulary of Figure 3.1.

tions of the scheme search space that are most likely to correctly model true paradigms of a language.

Two additional graphical examples, these generated from naturally occurring text, will help visualize scheme-based search spaces. Figure 3.3 contains a portion of a search space of schemes automatically generated from 100,000 tokens of the Brown Corpus of English (Francis, 1964); while Figure 3.4 diagrams a portion of a hierarchical lattice over a Spanish newswire corpus of 1.23 million tokens (50,000 types). As before, each box in these networks is a scheme and the c-suffix exponents appear in **bold**. Since most schemes contain more c-stem adherents than can be listed in a single scheme box, abbreviated lists of adherents appear in *italics*. The number immediately below the list of c-suffixes is the total number of c-stem adherents that fall in that scheme.

The scheme network in Figure 3.3 contains the paradigmatic level four scheme covering the English c-suffixes **Ø.ed.ing.s**. These four suffixes, which mark combinations of ***Tense***, ***Person***, ***Number***, and ***Aspect***, are the exponents of a true sub-class of the English verbal paradigm. This true paradigmatic scheme is embedded in a lattice of less satisfactory schemes. The right-most scheme in each row posits, in addition to true inflectional suffixes of English, the derivational suffix **ly**. Immediately below **Ø.ed.ing.s**, a scheme comprising a subset of the suffixes of the true verbal sub-class appears, namely **Ø.ed.ing**. To the left, **Ø.ed.ing.s** is connected to **d.ded.ding.ds**, a scheme which proposes an alternative morpheme boundary for 19 of the 106 c-stems in the scheme **Ø.ed.ing.s**.

Notice that since left links effectively slice a scheme on each character in the orthography, adherent count monotonically decreases as left links are followed. Similarly, adherent count monotonically decreases as c-suffix set inclusion links are followed upward. Consider again the hierarchically related schemes **Ø.ed.ing.s** and **Ø.ed.ing**, which have 106 and 201 adherents respectively. Since the **Ø.ed.ing.s** scheme adds the c-suffix **s** to the three c-suffixes already in the **Ø.ed.ing** scheme, only a subset of the c-stems which concatenate the c-suffixes **Ø**, **ed**, and **ing** also concatenate **s** to produce a word form found in this corpus of English. And so, only a subset of the c-stems in the **Ø.ed.ing**

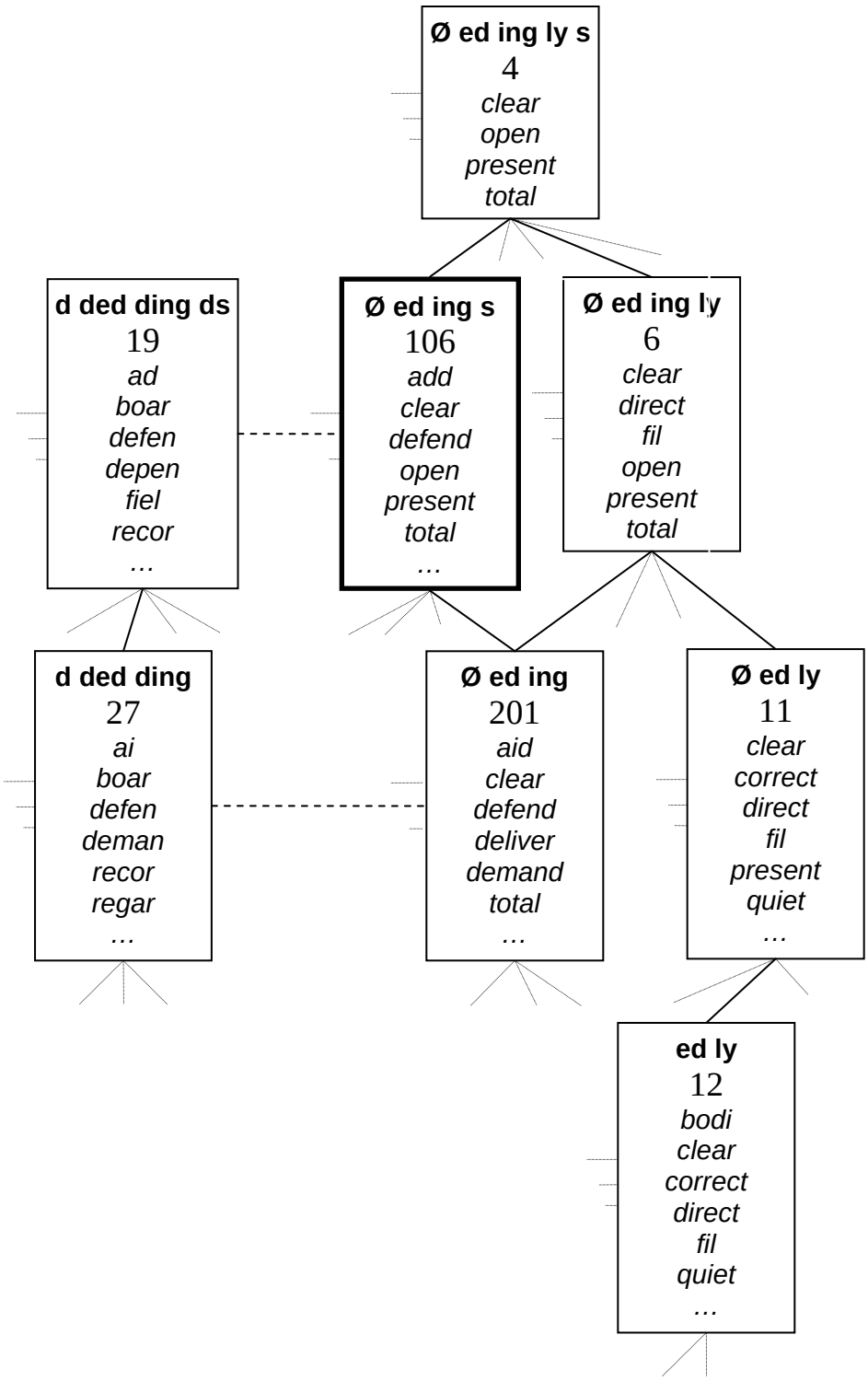

Figure 3.3: A portion of a morphology scheme network generated from 100,000 tokens from the Brown Corpus of English. The scheme that most closely matches a true verbal paradigm of English, **Ø.ed.ing.s**, is outlined in bold.

scheme adhere to the scheme **Ø.ed.ing.s**. The decrease in adherent c-stem count when moving upward through a scheme lattice is a consequence of the asymmetry in the definition of a scheme, see Section 3.1.1.

Now turning to Figure 3.4, this figure covers the ***Gender*** and ***Number*** paradigms on Spanish adjectival forms. As with Spanish *Past Participles*, adjectives in Spanish mark ***Number*** with the pair of paradigmatically opposed suffixes **s** and **Ø**; and ***Gender*** with the pair of strings **a** and **o**. Together the ***Gender*** and ***Number*** paradigms combine to form an emergent cross-product paradigm of four alternating strings: **a**, **as**, **o**, and **os**. Figure 3.4 contains:

1. The scheme containing the true Spanish exponents of the emergent cross-product paradigm for ***Gender*** and ***Number***: **a.as.o.os**. The **a.as.o.os** scheme is outlined in **bold**.
2. All possible schemes whose c-suffix exponents are subsets of **a.as.o.os**, e.g. **a.as.o**, **a.as.os**, **a.os**, etc.
3. The scheme **a.as.o.os.ualidad**, together with its descendents, **o.os.ualidad** and **ualidad**. The Spanish string **ualidad** is arguably a valid Spanish derivational suffix, forming nouns from adjectival stems. But the repertoire of stems to which **ualidad** can attach is severely limited. The suffix **ualidad** does not form an inflectional paradigm with the adjectival endings **a**, **as**, **o**, and **os**.

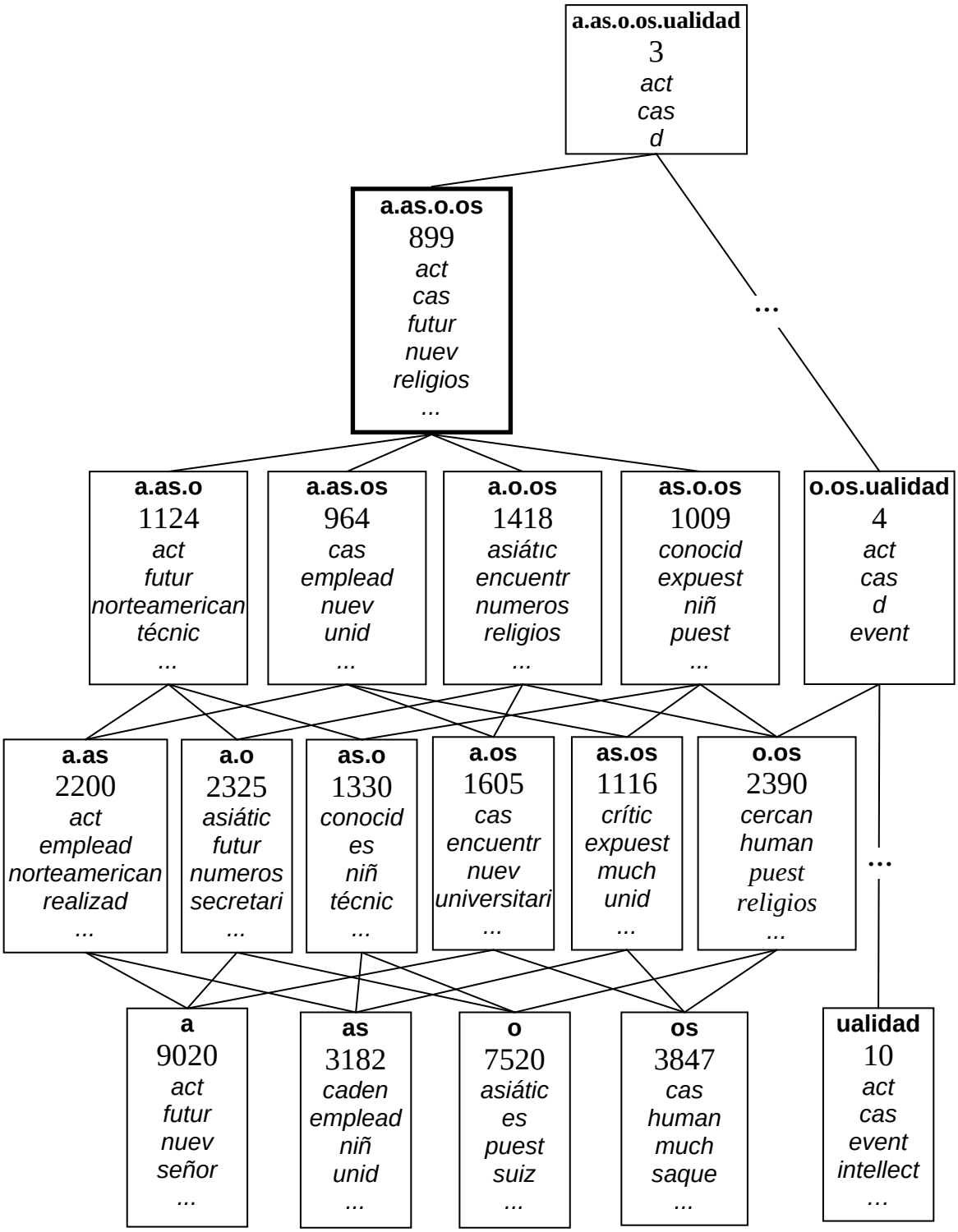

Figure 3.4: A hierarchical scheme-lattice search space automatically derived from a Spanish newswire corpus of 50,000 unique types (1.23 million tokens). For clarity, the scheme matching the productive adjectival paradigm of Spanish, **a.as.o.os**, is outlined in **bold**.

3.2 Searching the Scheme Lattice

Given the framework of morphology scheme networks outlined in Section 3.1, an unsupervised search strategy can automatically identify schemes which plausibly model true paradigms and their cross-products. Many search strategies are likely capable of identifying reasonable paradigmatic suffix sets in scheme networks. Snover (2002), for example, describes a successful search strategy over a morphology network in which each network node is assigned a global probability score (see Chapter 2). In contrast, ParaMor's search strategy, presented in this section, gauges a scheme's value by computing a local score, non-probabilistic, over the scheme's network neighbors.

3.2.1 A Bird's Eye View of ParaMor's Search Algorithm

Figure 3.5 abstractly visualizes ParaMor's initial search for partial paradigms. The two dimensional area of Figure 3.5 represents ParaMor's lattice-structured search space. Embedded in ParaMor's search space are schemes whose c-suffixes match the suffixes of true paradigms—in Figure 3.5, the large circles, atop the shaded triangles, symbolize these correct schemes. ParaMor's paradigm search seeks to find these correct schemes. However, ParaMor is often unable to extract sufficient evidence from the input corpus during this initial search phase to fully reconstruct the paradigms of a language. And so, the primary goal of ParaMor's initial search is to identify schemes which *individually* model subsets of the suffixes in true paradigms, and which *jointly* cover as many correct suffixes as possible. It will then be up to ParaMor's clustering algorithm, described in Section 4.3, to merge the initially selected scheme-models of partial paradigms.

In a scheme lattice, the descendents of a scheme, *C*, are those schemes whose c-suffix sets are subsets of the c-suffix set in *C*. Figure 3.5 conceptualizes the descendents of a scheme as a cone projecting downward. In particular, the shaded triangles in Figure 3.5, headed by the large paradigm circles, represent schemes whose c-suffix sets are subsets of true paradigms. Hence, to model paradigmatic suffixes, ParaMor's search algorithm should seek to identify schemes within the shaded cones. Moreover, ParaMor should pre-

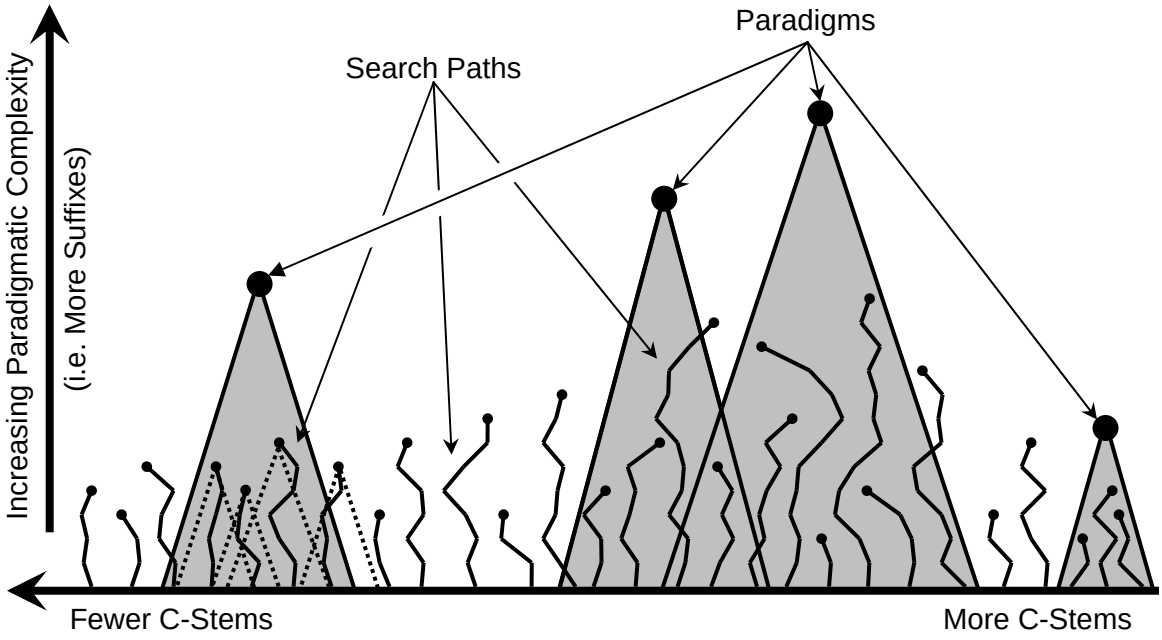

Figure 3.5: A conceptualization of ParaMor's initial search for partial paradigms.

fer widely spaced schemes located high in the lattice—so as to maximize c-suffix coverage.

To identify candidate partial paradigms, ParaMor adopts a bottom-up search strategy. (ParaMor will harness the horizontal morpheme boundary links, which also connect networked schemes in a later stage of the algorithm, see Section 4.4.2). At the bottom of a scheme network, and, conceptually, at the bottom of Figure 3.5, are the paradigmatic level one schemes which contain a single c-suffix each. A priori, ParaMor does not know which sets of c-suffixes model paradigms, and so, the initial search algorithm attempts to grow models of paradigmatic suffix sets by starting with single c-suffix schemes.

To ensure a wide spacing in the selection of schemes, ParaMor pursues a separate upward search path from *each* level-one scheme. By starting separate paths from all individual c-suffixes, ParaMor awards to every c-suffix the chance for consideration as a true suffix. This magnanimous policy is recall-centric: ParaMor's initial search hopes to find scheme models of as many true suffixes as possible. However, in starting a separate

search path from each c-suffix, ParaMor also increases the number of incorrectly selected c-suffixes—many search paths begin from c-suffixes which model no valid inflectional suffix. ParaMor relies on a series of filters, described in Chapter 4, to weed out the poorly chosen initial schemes.

From each level one scheme, ParaMor follows a single upward path. In a greedy fashion, ParaMor selects a single c-suffix to add to the current scheme. ParaMor limits each upward search to a single greedy path for computational reasons—the full search space of candidate paradigm schemes is immense and following all reasonable paths quickly leads to an explosion in the number of paths to follow (see Section 3.2.4).

Returning to Figure 3.5: Each jagged line in Figure 3.5 is a search path ParaMor pursues. Search paths originate at the bottom of the figure and move upward through the lattice. The terminal schemes of each search path, marked by small circles in Figure 3.5, are the final output of ParaMor's initial search. Some of ParaMor's search paths originate within a shaded triangle—these are paths that begin from a c-suffix that models a true suffix in some paradigm. But other search paths inevitably begin from c-suffixes that model no suffix in the language. Furthermore, some paths which begin at schemes which do model suffixes eventually take a wrong turn, introducing an incorrect c-suffix and thereby leaving the shadow of a true paradigm.

The final point to note from Figure 3.5 concerns the left-most paradigm in the figure. For several of the schemes which terminate search paths that originate beneath the left-most paradigm, Figure 3.5 indicates c-suffix subset cones with dashed lines. The union of the bases of these dashed cones closely coincides with the base of the shaded cone of the left-most paradigm. The base of each subset cone contains all schemes which contain just a single c-suffix from the scheme at the cone's apex. Thus, if the c-suffixes from the schemes at the bases of the dashed cones were reunited, the resultant paradigm model would closely match a true paradigm. This is exactly ParaMor's strategy: Section 4.3 will describe a clustering algorithm that unites schemes which each model subsets of suffixes from individual paradigms.

Why a Bottom-Up Search?

The choice to design ParaMor's initial paradigm identification algorithm as a repeated series of bottom-up search paths through a scheme network, a la Figure 3.5, was explicitly taken so as to leverage the paradigmatic and syntagmatic structure that is captured by the vertical c-suffix set inclusion links of the scheme network search space. At the bottom of a network of schemes, syntagmatic c-stem alternations are evident but each scheme contains only a single c-suffix. At successively higher levels, the networked schemes contain successively more paradigmatically opposed c-suffixes, but also successively fewer syntagmatic c-stems. ParaMor's search strategy moves upward through the network, trading off syntagmatic c-stem alternations for paradigmatic alternations of c-suffixes.

Consider the paradigmatic and syntagmatic structure captured by and between the schemes of the Spanish network in Figure 3.4 from p. 61. The schemes at the bottom of this network each contain exactly one of the c-suffixes **a**, **as**, **o**, **os**, or **ualidad**. The syntagmatic c-stem evidence for those level 1 schemes which model productive inflectional suffixes of Spanish, namely **a**, **as**, **o**, and **os**, is significantly greater than the syntagmatic evidence for the unproductive derivational c-suffix **ualidad**: The **a**, **as**, **o**, and **os** schemes contain 9020, 3182, 7520, and 3847 c-stems respectively, while the **ualidad** scheme contains just 10 c-stems.

Moving up the network, paradigmatic-syntagmatic tradeoffs strongly resonate. Among the 3847 c-stems which allow the c-suffix **os** to attach, more than half, 2390, also allow the c-suffix **o** to attach. In contrast, only 4 c-stems belonging to the **os** scheme form a corpus word with the c-suffix **ualidad**: namely, the c-stems **act**, **cas**, **d**, and **event**. Adding the suffix **a** to the scheme **o.os** again reduces the c-stem count, but only by 41%, from 2390 to 1418; and adding **as**, just lowers the c-stem count by a further 37%, to 899. There is little syntagmatic evidence for adding c-suffixes beyond the four in the scheme **a.as.o.os**. Adding the non-paradigmatic c-suffix **ualidad**, for example, drastically reduces the count of syntagmatic c-stems by over 99%, to a meager 3.

It is insightful to consider why morphology scheme networks capture paradigmatic-syntagmatic tradeoffs so succinctly. Take a particular c-suffix, *f*, which models a true in-

flectional suffix (or cross-product of suffixes). Disregarding morphophonologic change, the paradigmatic property of inflectional morphology implies *f* will be mutually substitutable for some distinct c-suffix *f′*. Consequently, both *f* and *f′* will occur in a text corpus attached to many of the same syntagmatically related c-stems. In our example, when *f* is the c-suffix **os** and *f′* the paradigmatically related **o**, many c-stems to which **os** can attach also allow **o** as a word-final string. Conversely, if the *f* and *f′* suffixes lack a paradigmatic relationship in the morphological structure of some language, then there is no a priori reason to expect *f* and *f′* to share c-stems: when *f* is **os** and *f′* **ualidad**, a c-suffix which is not paradigmatically opposed to **os**, few of the c-stems which permit an **os** c-suffix admit **ualidad**.

3.2.2 ParaMor's Initial Paradigm Search: The Algorithm

With the motivation and background presented in the previous sub-section, the specifics of ParaMor's search algorithm follow: The bottom-up search treats each individual non-null c-suffix (that is, all suffixes but **Ø**) as a potential gateway to a model of a true paradigm cross-product. ParaMor considers each one-suffix scheme in turn beginning with that scheme containing the most c-stems, and working toward one-suffix schemes containing fewer c-stems. From each bottom scheme, ParaMor follows a single greedy upward path from child to parent. As long as an upward path takes at least one step, making it to a scheme containing two or more alternating c-suffixes, ParaMor's search strategy accepts the terminal scheme of the path as a model of a portion of a morphological paradigm.

To take each greedy upward search step, ParaMor first identifies the best scoring parent of the current scheme according to a particular scoring function. Section 3.2.5 will propose and evaluate one reasonable class of parent scoring function. ParaMor then applies two criteria to the highest scoring parent. The first criterion is a threshold on the parent's score. ParaMor's upward search will only move to a parent whose score passes the set threshold.

The second criterion governing each search step helps to halt upward search paths before judging parents' worth becomes impossible. As noted in the second half of Section 3.2.1 above, c-stem counts monotonically decrease with upward network moves. But small adherent c-stem counts render statistics that assess parents' strength unreliable. ParaMor's policy is to not move to any scheme that does not contain more c-stems than it has c-suffixes. Moreover, this second halting criterion serves ParaMor well for two additional reasons. First, requiring each path scheme to contain more c-stems than c-suffixes ensures high suffix recall by setting a low bar for upward movement at the bottom of the network. Hence, search paths which begin from schemes whose single c-suffix models a rare but valid suffix, can often take at least one upward search step and manage to be selected. Second, this halting criterion requires the top scheme of search paths that climb high in the network to contain a comparatively large number of c-stems. Reigning in high-reaching search paths, before the c-stem count falls too far, captures path-terminal schemes which cover a large number of word types. In a later stage of ParaMor's paradigm identification algorithm, presented in Sections 4.3.1 and 4.3.2, these larger terminal schemes effectively vacuum up the useful smaller paths that result from the more rare suffixes.

At times, ParaMor's search algorithm can reach the same scheme along more than one path. But, since ParaMor's upward search from any particular scheme is deterministic, there is no need to retrace the same best path more than once. While it might be reasonable to follow a next-best path each time a scheme is re-encountered, ParaMor instead simply abandons the redundant path. In practice, the large number of individual paths that ParaMor follows ensures that ParaMor discovers all the prominent paradigms, without the need for next-best searching.

Pseudo code for ParaMor's bottom-up initial search for candidate paradigmatic schemes is given in Figure 3.6. The core ***search()*** method calls ***searchOneGreedyPath(·)*** once for each scheme which contains a single c-suffix. Most of the function names in the pseudo code are self-explanatory, but note that the ***initializeDynamicNetwork(·)*** method does not build the full scheme lattice. This initialization

```
// Global initializations
dynamicSchemeNetwork = initializeDynamicNetwork(corpus);
visitedSchemes = emptySet;  // Keep track of all schemes in any path

// Follow a separate upward search path for each scheme
// that contains just a single c-suffix.
search() {
  selectedSchemes = emptySet;
  levelOneSchemes = dynamicSchemeNetwork.getNonNullLevelOneSchemes();
  sortedLevelOneSchemes = sortByDecreasingCStemCount(levelOneSchemes);
  foreach (levelOneScheme in sortedLevelOneSchemes) {
    selectedScheme = searchOneGreedyPath(levelOneScheme);
    if (selectedScheme != null)
      selectedSchemes.add(selectedScheme);
  }
  return selectedSchemes;
}

searchOneGreedyPath(levelOneScheme) {
  currentScheme = levelOneScheme;
  while (true) {
    [bestParentScheme, scoreOfBestParentScheme] =
      dynamicSchemeNetwork.
        getBestParentSchemeWithScore(currentScheme);

    // The two criteria a parent scheme must pass for
    // ParaMor to move upward
    if ((scoreOfBestParentScheme < threshold) ||
        ( ! (bestParentScheme.getNumberOfCStems() >
              bestParentScheme.getNumberofCSuffixes())))

      // If the parent doesn't qualify then, as long as this search
      // path took at least one upward step, return the current,
      // now terminal, scheme.
      if (currentScheme.getNumberOfCSuffixes() > 1)
        return currentScheme;
      else
        return null;

    // Abandon redundant paths
    if (visitedSchemes.contains(bestParent))
      return null;

    // Loop updates
    visitedSchemes.add(bestParent);
    currentScheme = bestParent;
  } // End while
}
```

Figure 3.6: Pseudo-code implementing ParaMor's initial search for scheme models of partial paradigms.

method merely prepares the necessary data structures that enable ParaMor to quickly build schemes as needed. Details on ParaMor's dynamic network generation appear in Section 3.2.4.

Experiments with a range of languages, including Spanish, English, German, Finnish, and Turkish, show that ParaMor's bottom-up search algorithm takes less than 10 minutes to complete, when given a paradigm induction corpus containing 50,000 unique types. ParaMor is currently implemented in Java and runs on a standard Linux server.

3.2.3 ParaMor's Bottom-Up Search in Action

Figure 3.7 contains a number of search paths that ParaMor followed when analyzing a Spanish newswire corpus of 50,000 types and when using one particular metric for parent evaluation (See Section 3.2.5 for a discussion of scheme parent-evaluation metrics). Most of the paths in Figure 3.7 are directly relevant to the analysis of the Spanish word **administradas**. As mentioned in Chapter 1, the word **administradas** is the *Feminine Plural Past Participle* form of the verb **administrar**, 'to administer or manage'. The word **administradas** gives rise to many c-suffixes including: **stradas**, **tradas**, **radas**, **adas**, **das**, **as**, **s**, and **Ø**. Of these candidate suffixes, **s** marks Spanish plurals and is a word final string of 10,662 word forms in this Spanish corpus—more than one fifth of the unique word forms in end in **s**! Additionally, in the word **administradas**, the left edges of the word-final strings **as** and **adas** occur at Spanish morpheme boundaries. All other derived c-suffixes incorrectly segment **administradas**: The c-suffixes **radas**, **tradas**, **stradas**, etc. erroneously include part of the stem; The c-suffix **das**, in the analysis of Spanish morphology adopted by this thesis, places a morpheme boundary internal to the *Past Participle* morpheme **ad**; and the null c-suffix **Ø** does not mark any morphosyntactic feature on Spanish adjectives when it occurs *after* a *Plural* suffix. Of course, while we can discuss which c-suffixes are reasonable and which are not, an unsupervised morphology induction system has no a priori knowledge of Spanish morphology. ParaMor does not know what strings are valid Spanish morphemes, nor is ParaMor aware of the feature value meanings associated with morphemes.

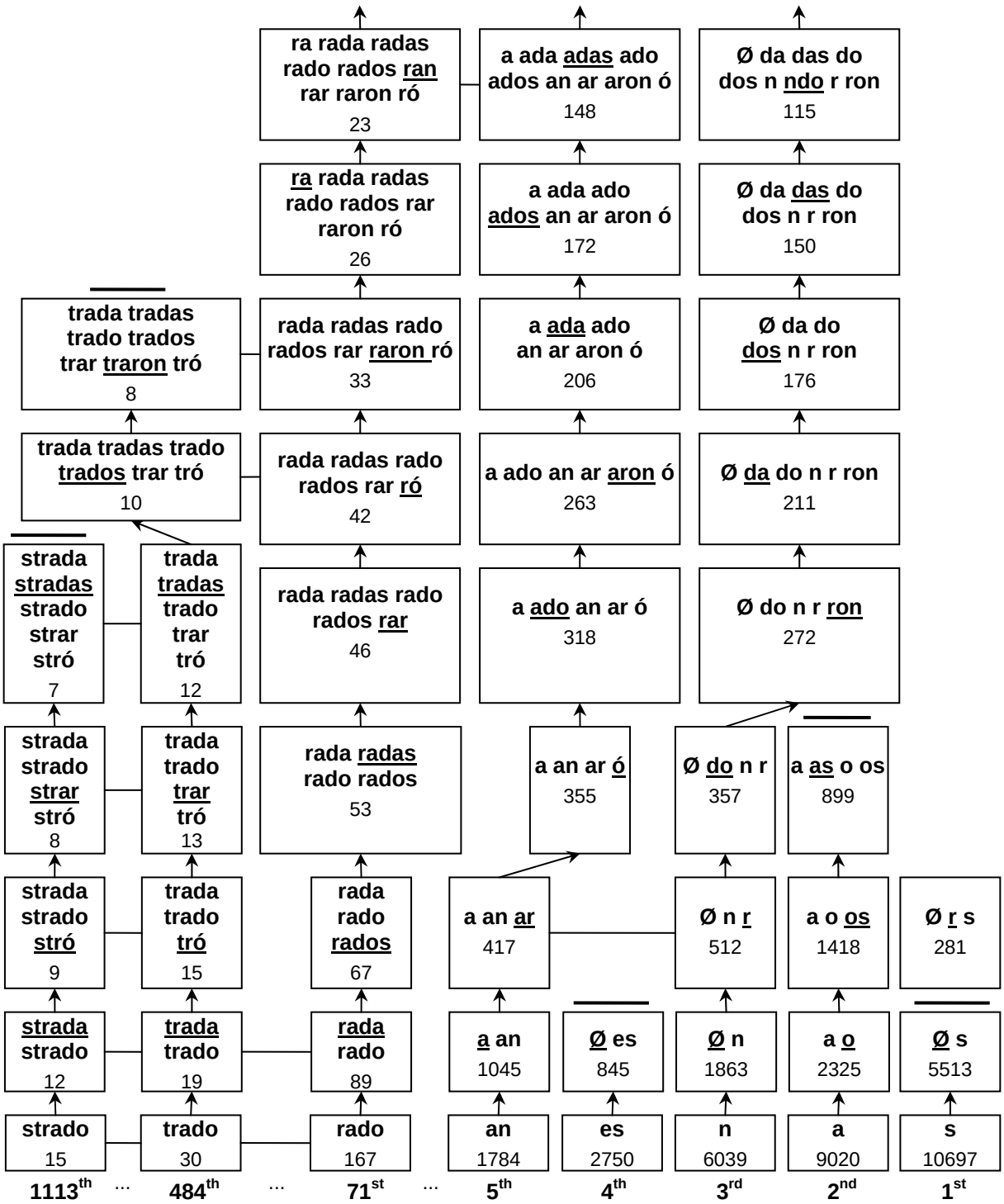

Figure 3.7: Eight search paths that ParaMor follows in search of schemes which likely model inflectional paradigms. Search paths begin at the bottom of the figure and move upward. All c-suffixes appear in **bold**. The underlined c-suffix in each scheme is the c-suffix added by the most recent search step. Each scheme gives the number of adherent c-stems it contains. Horizontal links between schemes connect sets of c-suffixes that differ only in their initial character.

Each search path of Figure 3.7 begins at the bottom of the figure and proceeds upwards from scheme to scheme. In Spanish, the non-null c-suffix that can attach to the most stems is **s**; and so, the first search path ParaMor explores begins from **s**. This first search path is the right-most path shown in Figure 3.7. At 5513 c-stems, the null c-suffix, **Ø**, can attach to the largest number of c-stems to which **s** can attach. The parent-evaluation function gave the **Ø.s** scheme the highest score of any parent of the **s** scheme, and the score of **Ø.s** passed the parent score threshold. Consequently, the first search step moves to the scheme which adds **Ø** to the c-suffix **s**.

ParaMor's parent-evaluation function then identifies the parent scheme containing the c-suffix **r** as the new parent with the highest score. Although no other c-suffix can attach to more c-stems to which **s** and **Ø** can both attach, **r** can only form corpus words in combination with 281 or 5.1% of the 5513 c-stems which take **s** and **Ø**. Accordingly, the score assigned by this particular parent-evaluation function to the **Ø.s.r** scheme falls below the stipulated threshold; and ParaMor does not add **r**, or any other suffix, to the now closed and selected partial paradigm **s.Ø**. That ParaMor did not follow a path to the **Ø.s.r** scheme is indicated in Figure 3.7 by the horizontal bar above the **Ø.s** scheme.

Continuing leftward from the **s**-anchored search path in Figure 3.7, ParaMor follows search paths from the c-suffixes **a**, **n**, **es**, and **an** in turn. The 71st c-suffix from which ParaMor grows a partial paradigm is **rado**. The search path from **rado** is the first path to build a partial paradigm that includes the c-suffix **radas**, a c-suffix potentially relevant for an analysis of the word **administradas**. Similarly, search paths from **trado** and **strado** lead to partial paradigms which include, respectively, the c-suffixes **tradas** and **stradas**—two c-suffixes which also occur in **administradas**. The search path from **strado** illustrates the second criterion that restricts ParaMor's upward search. From **strado**, ParaMor adds four c-suffixes one at a time: **strada**, **stró**, **strar**, and **stradas**. Only seven c-stems form words when combined singly with all five of these c-suffixes. Adding any additional c-suffix to these five brings the c-stem count in this corpus down at least to six. Since six c-stems is not more than the six c-suffixes which would be in the resulting parent scheme, ParaMor does not add a sixth c-suffix—halting this upward search path.

3.2.4 The Construction of Scheme Networks

It is computationally impractical to build full morphology scheme networks, both in terms of space and time. The space and time complexities of full network creation are directly related to the number of schemes in a network. Returning to the definition of a scheme in Section 3.1.1, each scheme contains a set of c-suffixes, $F \subset F^V$, where F^V is the set of all possible c-suffixes generated by a vocabulary. Thus, the set of potential schemes from some particular corpus is the exponentially large powerset of F^V, with $2^{|F^V|}$ members. In practice, the vast majority of potential schemes have no adherent c-stems—that is, for most $F \subset F^V$ there is no c-stem, *t*, such that $\forall f \in F\, t.f$ is a word form in the vocabulary. If a scheme has no adherent c-stems, then there is no evidence for that scheme, and even an exhaustive network generation algorithm would not need to actually create that scheme.

Unfortunately, the number of schemes which *do* possess adherent c-stems also grows exponentially. The dominant term in the number of schemes with a non-zero c-stem count comes from the non-empty scheme with the largest set of c-suffixes. In one corpus of 50,000 unique Spanish types, a scheme exists that has a single adherent c-stem and 5816 c-suffixes. This one 5816^{th} level scheme implies the existence of all 2^{5816} of its descendents. The number 2^{5816} is truly astronomical, larger than 10^{500}, or larger, by far, than the number of hydrogen atoms in the observable universe.

Because of the difficulty in pre-computing full scheme networks, during the initial upward scheme search (described in Sections 3.2.1 and 3.2.2) any needed schemes are calculated dynamically. The rest of this section spells out ParaMor's scheme-building procedure in detail.

Most-Specific Schemes

ParaMor builds individual schemes dynamically from *most-specific schemes*. Like full-blooded schemes, each most-specific scheme, *M*, is a pair of sets of strings, (T_M, F_M): T_M a set of c-stems, and F_M a set of c-suffixes. Furthermore, as in the definition of a scheme found in Section 3.1.1, four conditions stipulate which c-stems and

c-suffixes occur in each most-specific scheme. The first three conditions for most-specific schemes are identical to the first three conditions on full schemes:

1. $T_M \subset T^V$ — where T^V is the set of all c-stems generated by a vocabulary
2. $F_M \subset F^V$ — F^V the set of all c-suffixes generated by a vocabulary
3. $\forall t_M \in T_M, \forall f_M \in F_M,\ t_M . f_M \in V$ — V being the corpus vocabulary

But the fourth condition changes in the definition of most-specific scheme. The new condition is:

4′. $\forall t_M \in T_M,\ \neg\exists\, f^V \in F^V,\ f^V \notin F_M$ such that $t_M . f^V \in V$

This new fourth condition requires each $t_M \in T_M$ to form vocabulary words with exactly and only the c-suffixes in F_M (exactly the situation imagined in connection with Figure 3.1 for the c-stem **blame** and the scheme **Ø.s** in Section 3.1.1 on p. 53).

A consequence of this new fourth restriction that differentiates most-specific schemes from basic schemes is that each corpus c-stem occurs in precisely one most-specific scheme: Suppose a c-stem *t* belonged to two most-specific schemes, $M = (T_M, F_M)$ and $M' = (T'_M, F'_M)$, $F'_M \neq F_M$. Since $F'_M \neq F_M$, without loss of generality, there must exist $f' \in F'_M$ such that $f' \notin F_M$. And because $t \in T'_M$, it follows that $t.f' \in V$. But this situation violates the specification of *M* as a most-specific scheme: There exists a c-stem in T_M, namely *t*, and a c-suffix that is not in F_M, namely f', that concatenate to form a vocabulary item. Hence, *t* cannot be an element of T_M. And *t* cannot belong to both *M* and M'.

The idea of the most-specific scheme has been proposed previously: translating into the terminology of this thesis, Demberg (2007) stores c-stems that are not surface types themselves in most specific schemes. Furthermore, as the c-suffixes of a most-specific scheme, *C*, exactly specify the yield of the c-stems in *C*, the states in the minimal (forward) character-based finite state automaton that exactly accepts a vocabulary are in a one-to-one correspondence with the set of most-specific schemes generated by that vocabulary. And so, finite state approaches to morphology induction, such as Johnson and Martin (2003), also implicitly compute most-specific schemes. But to my knowledge, no

one has suggested dynamically generating full-fledged schemes from their most-specific cousins, as is proposed here.

Computing Most-Specific Schemes

Since the number of most-specific schemes is bounded above by the number of c-stems in a corpus, the number of most-specific schemes grows much more slowly with vocabulary size than does the total number of schemes in a network. In the 50,000 type Spanish corpus from above, a mere 28,800 (exact) most-specific schemes occurred.

A two-part algorithm can quickly compute the set of most-specific schemes from a corpus. The first half of the algorithm associates each c-stem, *t*, with a set of c-suffixes, F_t, where each c-suffix in F_t forms a vocabulary word when combined with *t*. To find F_t for each *t*, ParaMor loops through each (c-stem *t*, c-suffix *f*) division of each vocabulary word. Whenever ParaMor encounters a specific c-stem, *t*, ParaMor adds the corresponding *f* c-suffix of the current pair to F_t. In the second half of the algorithm to compute the set of most-specific schemes, ParaMor associates each unique set of c-suffixes, *F*, that was computed in the algorithm's first half with the set of c-stems that individually pointed to identical copies of *F*. Pseudo-code for the creation of most-specific schemes is given in Figure 3.8.

Overloading the $|\cdot|$ operator to denote both the size of a set and the length of a string, the time complexity of this two-part algorithm to compute most-specific schemes is $O\left(|V| * ave_{v \in V} |v| + \left|T^V\right|\right)$, where, as above, *V* is the vocabulary size and T^V is the set of all c-stems generated from a vocabulary. In practice, computing the most-specific schemes from corpora of 50,000 types takes the current Java implementation of ParaMor less than five minutes on a standard Linux server, with ParaMor spending about equal time in the first and second halves of the algorithm.

To Schemes from Most-Specific Schemes

From the full set of most specific schemes, the c-stems, T_C, of any particular scheme, $C = (T_C, F_C)$, can be directly computed. Since a single c-stem can belong to multiple

```
// Define a c-stem, c-suffix pair
struct SegmentedWord {
  cStem;
  cSuffix;
};

// Two structures in which to save the results of
// the first and second halves of the algorithm
Hash<cStem, Set<cSuffix>> cStemToCSuffixes;
Hash<Set<cSuffix>, Set<cStem>> mostSpecificSchemes;

// First Half: Find all c-suffixes that can attach to each c-stem.
foreach (word in corpus.vocabulary) {
  Set<SegmentedWord> segmentations = corpus.getAllSegmentations(word);
  foreach (segmentation in segmentations) {
    cStemsToCSuffixes{segmentation.cStem}.add(segmentation.cSuffix);
  }
}

// Second Half: Find all c-stems of each c-suffix set.
foreach (cStem in cStemsToCSuffixes) {
  cSuffixes = cStemToCSuffixes{cStem};
  mostSpecificSchemes{cSuffixes}.add(cStem);
}
```

Figure 3.8: Pseudo-code for computing most-specific schemes. Note that this code uses curly braces '{ }' to index into hashes. And as this algorithm is heavily dependent on data structures, this code segment is strongly typed: pointy brackets '< >' indicate the data type that a hash or set may contain.

schemes but to only one most-specific scheme, the c-stems of the *C* scheme are scattered among various most-specific schemes. Recall from the definition of a scheme (Section 3.1.1) that the c-stems in T_C are all those c-stems which form words with all (but *not* only) the c-suffixes in F_C. Let $M = (T_M, F_M)$ be a most-specific scheme where $F_C \subseteq F_M$, and let $t_M \in T_M$. Since all c-stems in T_M form words with all c-suffixes in F_M, and since $F_C \subseteq F_M$, t_M must form a word with all c-suffixes in F_C. Hence, t_M belongs in T_C. The converse, that if $F_C \not\subset F_M$ then t_M does not belong in T_C, follows from similar logic. Consequently, to dynamically compute T_C, ParaMor must place in T_C all

c-stems from all most-specific schemes whose c-suffix sets are (proper or improper) supersets of F_C.

Let $\mathcal{M}^{F_C}$ denote the set of all most-specific schemes whose c-suffix sets are supersets of F_C. And in general, given a set of c-suffixes, *F*, define $\mathcal{M}^{F}$ as the set of most specific schemes whose c-suffixes are a (proper or improper) superset of *F*. To quickly compute the c-stems of any particular scheme, $C = (T_C, F_C)$, ParaMor must efficiently compute $\mathcal{M}^{F_C}$, i.e. without exhaustively comparing F_C to the c-suffix set of every most specific scheme.

Consider $\mathcal{M}^{\{f_C\}}$, the set of all most-specific schemes whose c-suffix sets contain f_C, for each c-suffix $f_C \in F_C$. These $\mathcal{M}^{\{f_C\}}$ are inexpensive to compute: for each most-specific scheme, $M = (T_M, F_M)$, and for each c-suffix $f_M \in F_M$, simply add *M* to the set of all most-specific schemes that f_M occurs in. Furthermore, $\mathcal{M}^{\{f^V\}}$ for any particular c-suffix f^V from a vocabulary need only be computed once. Figure 3.9 gives pseudo-code, that ParaMor calls just once, to compute $\mathcal{M}^{\{f^V\}}$ for all individual f^V c-suffixes that are generated by a corpus.

```
computeMostSpecificAncestorsOfAllSingleCSuffixes(mostSpecificSchemes) {

foreach (mostSpecificScheme in mostSpecificSchemes) {
    foreach (cSuffix in mostSpecificScheme.cSuffixes) {
      mostSpecificAncestors{cSuffix}.add(mostSpecificScheme);
    }
  }
  return mostSpecificAncestors;
}
```

Figure 3.9: Pseudo-code to compute $\mathcal{M}^{\{f^V\}}$, the set of all most-specific schemes whose set of c-suffixes include f^V, where f^V ranges over all c-suffixes generated by a corpus. The pseudo-code refers to $\mathcal{M}^{\{f^V\}}$ as the *ancestors* of f^V: Imagining a network of most-specific schemes mirroring the scheme networks of Section 3.1.2, the set $\mathcal{M}^{\{f^V\}}$ contains all the upward, or ancestor, most-specific schemes of the f^V node. As in Figure 3.8, curly braces '{ }' index hashes.

But to dynamically compute the c-stems of a scheme, $C = (T_C, F_C)$, ParaMor's must find $\mathcal{M}^{F_C}$, the set of all most-specific schemes whose c-suffix sets are supersets of F_C, not individual $\mathcal{M}^{\{f_C\}}$. As the following Lemma shows, $\mathcal{M}^{F_C}$ can be computed from the set of all $\mathcal{M}^{\{f_C\}}$ where f_C ranges over all c-suffixes in F_C:

Lemma: $\bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}} = \mathcal{M}^{F_C}$

To show $\mathcal{M}^{F_C} \subseteq \bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}}$: If the most-specific scheme $M \in \mathcal{M}^{F_C}$, where $M = (T_M, F_M)$, that is, if $F_C \subseteq F_M$, then $\forall f_C \in F_C$, $f_C \in F_M$, and $\{f_C\} \subseteq F_M$. And so $\forall f_C \in F_C$, $M \in \mathcal{M}^{\{f_C\}}$. And thus $M \in \bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}}$.

To show $\bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}} \subseteq \mathcal{M}^{F_C}$: If the most-specific scheme $M \in \bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}}$, where $M = (T_M, F_M)$, then $\forall f_C \in F_C$, $M \in \mathcal{M}^{\{f_C\}}$. And By the def of $\mathcal{M}^{\bullet}$, $\forall f_C \in F_C$, $\{f_C\} \subseteq F_M$; and of course $f_C \in F_M$. But suppose $M \notin \mathcal{M}^{F_C}$, then there must exist $f_C \in F_C$ such that $f_C \notin F_M$—a contradiction. So, $M \in \mathcal{M}^{F_C}$ and $\bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}} \subseteq \mathcal{M}^{F_C}$. The lemma has been proved.

In summary then, for any scheme $C = (T_C, F_C)$, with a specified c-suffix set, F_C, the c-stems T_C are the union of the c-stems from all most-specific schemes in $\mathcal{M}^{F_C}$, that is, $T_C = \bigcup_{M \in \mathcal{M}^{F_C}} T_M$, while the members of each $\mathcal{M}^{F_C}$ are found by intersecting the pre-computed sets of most-specific schemes in the $\mathcal{M}^{\{f_C\}}$, or $\mathcal{M}^{F_C} = \bigcap_{f_C \in F_C} \mathcal{M}^{\{f_C\}}$. The key is that both the intersection step as well as the unification step take just linear time to compute. Pseudo code for the full algorithm to dynamically compute a scheme from a set of most-specific schemes is given in Figure 3.10.

3.2.5 Upward Search Metrics

As described in detail in Sections 3.2.1 and 3.2.2 of this chapter, at each step of ParaMor's bottom-up search, the system selects, or declines to select, a parent of the current scheme as most likely to build on the paradigm modeled by the current scheme. A key

```
computeTheCStemsOfAScheme(cSuffixesOfScheme, mostSpecificAncestors) {

  // Intersect the pre-computed sets of most-specific scheme ancestors
  // of each individual c-suffix in cSuffixesOfScheme.
  mostSpecificAncestorsOfScheme = emptySet;
  foreach (cSuffix in cSuffixesOfScheme) {
    if (mostSpecificAncestorsOfScheme == emptySet)
      mostSpecificAncestorsOfScheme.
        addAll(mostSpecificAncesors{cSuffix});  // Deep Copy
    else {
      foreach (mostSpecificAncestor in mostSpecificAncestorsOfScheme) {
        if ( ! mostSpecificAncestors{cSuffix}.
                  contains(mostSpecificAncesor))
          mostSpecificAncestorsOfScheme.remove(mostSpecificAncestor);
      }
    }
  }

  // Compute the union all c-stems in all the most-specific scheme
  // ancestors of the scheme we are building.
  foreach (mostSpecificAncestor in mostSpecificAncestorsOfScheme) {
    cStemsOfScheme.add(mostSpecificAncestor.cStems);
  }

  return cStemsOfScheme;
}
```

Figure 3.10: Pseudo-code to dynamically compute the c-stems that belong to the scheme containing the specific set of c-suffixes found in `cSuffixesOfScheme`. This algorithm uses the relevant sets of most-specific schemes found in $\mathcal{M}^{\{f^V\}}$, as pre-computed in Figure 3.9, where f^V is a c-suffix generated by the corpus. As in Figure 3.9, the $\mathcal{M}^{\{f^V\}}$ are saved in the variable `mostSpecificAncestors` as a hash on c-suffixes. Curly braces '{ }' index into hashes.

element of ParaMor's bottom-up search is the exact form of the parent-evaluation metric used to select the most promising parent. The parent metric must reliably determine where and when to expand the current candidate paradigm using only the limited information available to an unsupervised induction system. ParaMor has no knowledge of, for example, the part of speech or morphosyntactic features that individual words mark. In-

stead, ParaMor's parent-evaluation function must work from co-occurrence statistics of words and of word substrings, e.g. from c-suffix and c-stem counts in schemes.

Clearly, ParaMor's parent-evaluation procedure directly impacts performance. A variety of parent-evaluation measures are conceivable, from simple local measures to statistical measures which take into consideration the schemes' larger contexts. This section investigates one class of localized metric and concludes that, at least within this metric class, a conceptually and computationally simple metric gives as accurate an indication of the paradigmatic worth of a parent scheme as more complex and expensive metrics.

The Parent Selection Dilemma

To motivate the metrics under investigation, consider the plight of an upward search algorithm that has arrived at the **a.o.os** scheme when searching through the Spanish morphology scheme network of Figure 3.4 from p. 61. All three of the c-suffixes in the **a.o.os** scheme model inflectional suffixes from the cross-product paradigm of ***Gender*** and ***Number*** on Spanish adjectives. In Figure 3.7 on p. 70, the second upward search path that ParaMor pursues brings ParaMor to the **a.o.os** scheme (the second search path in Figure 3.7 is the second path from the right). In both Figure 3.4 and in Figure 3.7, just a single parent of the **a.o.os** scheme is shown, namely the **a.as.o.os** scheme. But these two figures cover only a portion of the full scheme network derived from the 50,000 types in this Spanish corpus. In the full scheme network there are actually 20,949 parents of the **a.o.os** scheme! The vast majority of the parents of the **a.o.os** scheme occur with just a single c-stem. However, 1,283 parents contain two c-stems, 522 contain three c-stems, and 330 contain four, etc.

Seven of the more likely parents of the **a.o.os** scheme are shown in Figure 3.11. Out of nearly 21,000 parents, only one correctly builds on this adjectival inflectional cross-product paradigm of ***Gender*** and ***Number***: The **a.as.o.os** parent adds the c-suffix **as,** which marks *Feminine Plural*. The parent scheme of **a.o.os** that has the second most c-stem adherents, also shown in Figure 3.11, adds the c-suffix **amente.** Like the English suffix **ly**, the Spanish suffix **(a)mente** derives adverbs from adjectives quite productively. The other parents of the **a.o.os** scheme given in Figure 3.11 arise from c-suffixes that

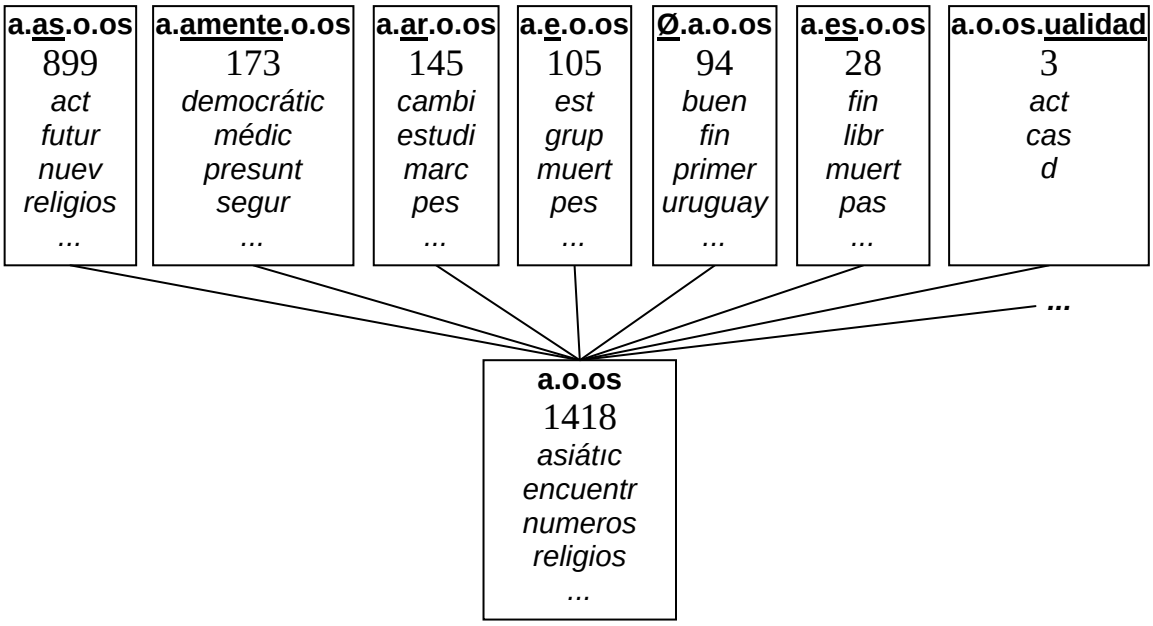

Figure 3.11: Seven of the 20,949 parents of the **a.o.os** scheme derived from the same Spanish newswire corpus of 50,000 types as Figure 3.4. The c-suffix added by each parent is underlined.

model verbal suffixes or that model derivational morphemes. The verbal c-suffixes are **ar**, **e**, and **es**, while the derivational c-suffixes are **Ø** and **ualidad**.

The primary reason for the fairly high c-stem counts among the 'verbal' parents of the **a.o.os** scheme is that Spanish syncretically employs the strings **a** and **o** not only as adjectival suffixes marking *Feminine* and *Masculine*, respectively, but also as verbal suffixes marking *3rd Person* and *1st Person Present Indicative*. However, for a c-stem to occur in a 'verbal' parent of **a.o.os**, such as **a.ar.o.os**, the c-stem must somehow combine with **os** into a non-verbal Spanish word form—as the c-suffix **os** does not model any verbal inflection. In the **a.ar.o.os** scheme in Figure 3.11, the four listed c-stems **cambi**, **estudi**, **marc**, and **pes** model verb stems when they combine with the c-suffixes **a** and **ar**, but they model, often related, noun stems when they combine with **os**, and the Spanish word forms **cambio**, **estudio**, **marco**, and **peso** can ambiguously be both verbs and nouns.

Motivation for Paradigmatic-Syntagmatic Parent-Evaluation Metrics

The task laid before the parent-evaluation function in ParaMor's bottom-up scheme search algorithm is, then, to select, out of potentially thousands of candidates, at most a single parent which best extends the paradigm modeled by the current scheme. As detailed in the second half of Section 3.2.1, ParaMor relies on the paradigmatic-syntagmatic frequency structure of scheme networks to make these parent selection decisions.

To briefly recapitulate the paradigmatic-syntagmatic tradeoffs captured in scheme networks: suppose P is a set of suffixes which form a paradigm or paradigm cross-product. And let $F \subset P$ be the set of c-suffixes in some scheme. Because the suffixes of P are mutually substitutable, it is reasonable to expect that, in any given corpus, many of the stems which occur with F will also occur with other suffixes, $f \in P$, $f \notin F$. Hence, when moving upward through a scheme network among schemes all of whose c-suffixes belong to a single paradigm, adherent c-stem counts should fall only slowly. Conversely, there is no reason to expect that some c-suffix $f' \notin P$ will form valid words with the same c-stems that form words with c-suffixes in *P*.

The C-Stem Ratio: A Simple and Effective Parent-Evaluation Metric

Taking another look at Figure 3.11, the parents of the **a.o.os** scheme clearly display paradigmatic and syntagmatic structure. More than 63% of the c-stems in the **a.o.os** scheme form a word with the c-suffix **as**, a c-suffix which is paradigmatically tied to **a**, **o** and **os** through the Spanish adjectival paradigm. Only 10% of the c-stems in **a.o.os** form corpus words with the c-suffix **ar**, part of a productive inflectional verbal paradigm with **a** and **o**. And only 0.2% of the c-stems in the **a.o.os** scheme form words with the derivational c-suffix **ualidad**, which is not part of a productive inflectional paradigm with any of the three c-suffixes **a**, **o** or **os**.

Among all parents of the **a.o.os** scheme, the correct paradigmatic parent, **a.as.o.os**, retains by far the highest fraction of the child's c-stems. Parent-child c-stem ratios are surprisingly reliable predictors of when a parent scheme builds on the paradigmatic c-suffix interaction of that scheme, and when a parent scheme breaks the paradigm. Indeed, while the remainder of this chapter presents and motivates a variety of parent met-

rics, an empirical evaluation discussed below (p. 91) suggests that no examined metric significantly outperforms the parent-child c-stem ratio at paradigm identification.

Parent-Evaluation Metrics beyond C-Stem Ratios

Parent-child c-stem ratios are a simple measure of a parent scheme's paradigmatic strength, but it seems reasonable that a more sophisticated measure might more accurately predict when a parent extends a child scheme's paradigm. In particular, the c-stem ratio metric does not at all account for the fact that some suffixes occur less frequently than others. In Spanish, for example, although the derivational suffix **(a)mente** so productively converts adjectives to adverbs that its paradigmatic behavior is nearly that of an inflectional suffix, in naturally occurring text, relatively few unique types end in **amente**. In comparison, the verbal infinitive suffix **ar** is much more common. In one corpus of 50,000 unique types 1448 word forms ended in **ar** but only 332 ended in **amente**. But the parent-child c-stem ratio does not strongly differentiate between the parent of the **a.o.os** scheme which introduces the productive (de)adjectival c-suffix **amente** and the parent which introduces the non-adjectival verbal c-suffix **ar**. Referring to Figure 3.11, both the schemes **a.amente.o.os** and **a.ar.o.os** have very nearly the same number of c-stems, 173 and 145 respectively, and so have very similar parent-child c-stem ratios of 0.122 and 0.102. The higher frequency of **ar** forms ensures that a reasonable number of c-stems adhere to the **a.ar.o.os** scheme.

Five additional parent-evaluation metrics are discussed below. All five incorporate the frequency of the expansion c-suffix, that is, they incorporate the frequency of the c-suffix introduced by the parent scheme under consideration. In a scheme network, the frequency of the expansion scheme is found in the level 1 scheme which contains the single c-suffix which expands the current scheme.

Figure 3.12 depicts expansion schemes, with their c-stem frequencies, for four parents of the **a.o.os** scheme. These four parents were chosen to aid exposition. The expansion schemes are: **as**, **amente**, **ar**, and **ualidad**, respectively the fourth member of the

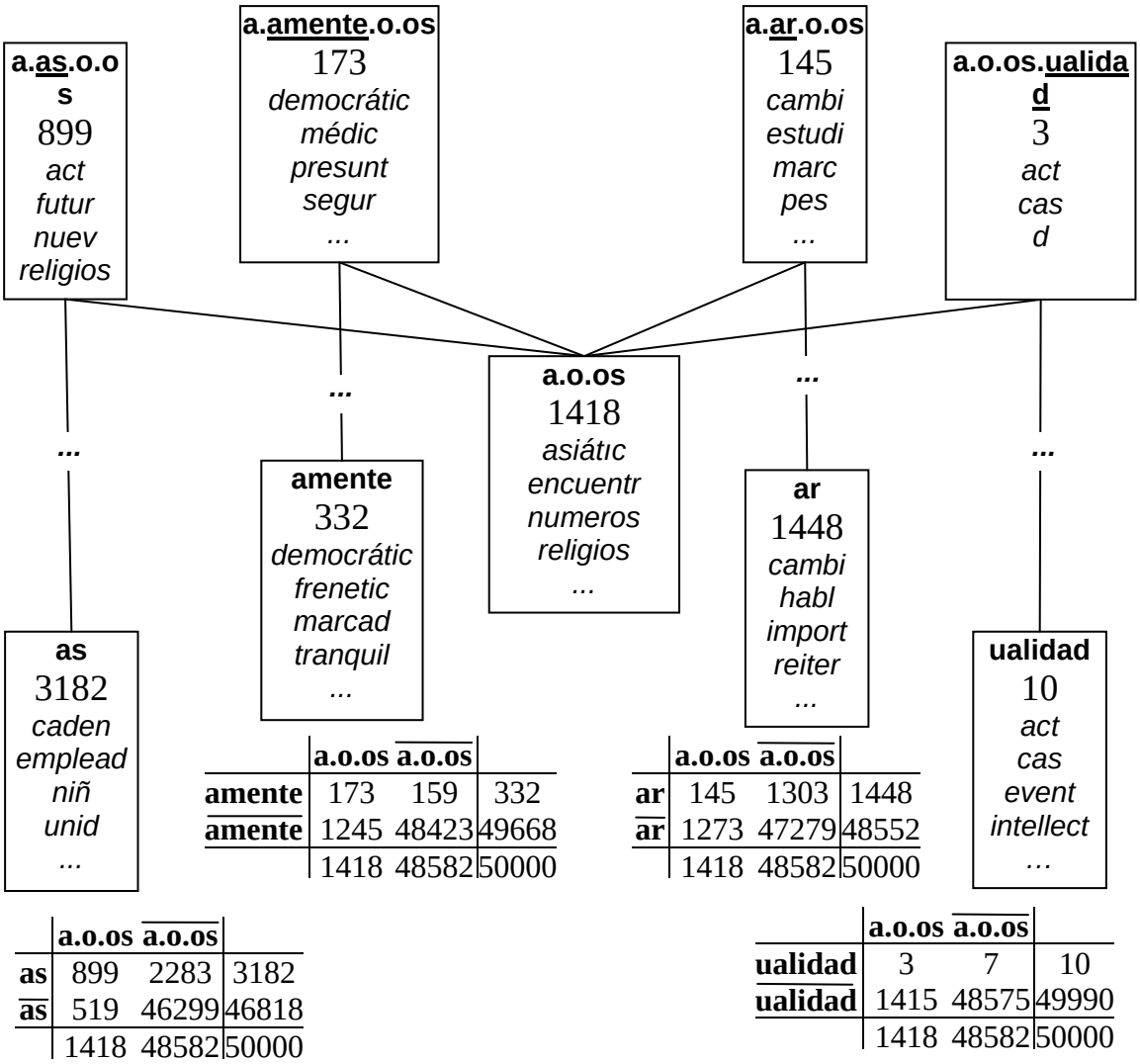

Figure 3.12: Four parents of the **a.o.os** scheme, together with the level 1 expansion schemes which contain the single c-suffix which expands **a.o.os** into the parent scheme. Beneath each expansion scheme is a table of c-stem counts relevant to **a.o.os** and that parent. These schemes and c-stem counts come from a Spanish newswire corpus of 50,000 types.

adjectival cross-product paradigm, a productive deadjectival derivational suffix, a verbal suffix that is not paradigmatically related to **os**, and an unproductive deadjectival derivational suffix.

The Dice Metric

There are many ways the c-stem information from expansion schemes might be exploited. One method is to average parent-expansion c-stem ratios with parent-child c-stem ratios. In the **a.amente.o.os** example, the ratio of c-stem counts from the parent scheme **a.amente.o.os** to the expansion scheme **amente**, at 173/332, can be averaged with the ratio of c-stems from the parent scheme to the child scheme **a.o.os**, namely 173/1418. The bottom-up search of scheme networks particularly seeks to avoid moving to schemes that do not model paradigmatically related c-suffixes. Taking the harmonic mean of the parent-expansion and parent-child c-stem ratios, as opposed to the arithmetic mean, captures this conservative approach to upward movement. Compared with the arithmetic mean, the harmonic mean comes out closer to the lower of a pair of numbers, effectively dragging down a parent's score if either c-stem ratio is low.

Interestingly, after a bit of algebra, it emerges that taking the harmonic mean of the parent-expansion and parent-child c-stem ratios is equivalent to measuring the dice similarity metric on the sets of c-stems present in the child and expansion schemes. The dice metric, a standard measure of set similarity (Manning and Schütze, 1999, p. 299), is defined as $\left(2|X \cap Y|\right)/\left(|X|+|Y|\right)$, for any two arbitrary sets *X* and *Y*. In the context of schemes, the intersection of *X* and *Y* is the intersection of the c-stem sets of the child and expansion schemes—or exactly the c-stem set of the parent scheme.

As hoped, the relative difference between the dice scores for the **amente** and **ar** parents is larger than the relative difference between the parent-child c-stem ratios of these parents. The dice scores are 0.198 and 0.101 for the **amente** and **ar** parents respectively, a difference of nearly a factor of two; as compared with the relative difference factor of 1.2 for the parent-child c-stem ratios of the **amente** and **ar** parents. Note that it is meaningless to directly compare the numeric value of a parent-child c-stem ratio to the nu-

meric value of the dice measure of the same parent—the two scores operate on different scales.

The parent-child c-stem ratio metric and the dice metric are the first two of six metrics that ParaMor investigated as potential guiding metrics for the vertical network search described in Sections 3.2.1 and 3.2.2. All six investigated metrics are summarized in Figure 3.13. Each row of this figure details a single metric. After the metric's name which appears in the first column, the second column gives a brief description of the metric, the third column contains the mathematical formula for calculating that metric, and the final four columns apply each row's metric to the four illustrative parent schemes of the **a.o.os** scheme from Figure 3.12: For example, the parent-child c-stem ratio from the **a.o.os** scheme to the **a.o.os.ualidad** parent is given in the upper-right cell of Figure 3.13 as 0.002.

The variables in the mathematical notation of the third column of Figure 3.13 refer to c-stem counts of schemes involved in the evaluation of parent schemes. For example, the formula, in the first row of Figure 3.13, to calculate a parent-child c-stem ratio is P/C, where P is the count of the c-stems in the **p**arent scheme, and C is the count of c-stems in the **c**urrent scheme. Comparing the metric formulas in the third column of Figure 3.13, the parent-child c-stem ratio is by far the simplest of any metric ParaMor examined. In addition to P and C, the formulas in Figure 3.13 refer to two other variables: E represents the total number of c-stems that can attach to the c-suffix that **e**xpands the current scheme into the parent (i.e. the c-stem adherent size of the paradigmatic level one **e**xpansion scheme), and V is the size of the **v**ocabulary over which the network was built. The formula for the dice metric uses E in addition to P and C: $2P/(C+E)$.

Pointwise Mutual Information as a Parent-Evaluation Metric

Of the six parent-evaluation metrics that ParaMor examined, the four which remain to be described all look at the occurrence of c-stems in schemes from a probabilistic perspective. To convert c-stem counts into probabilities, ParaMor must estimate the total number of c-stems which could conceivably occur in a single scheme. ParaMor takes this

	Metric	Explanation	Formula	Parent Score			
				as	amente	ar	ualidad
Heuristics	**Ratio**	Ratio of parent c-stems to current c-stems	$\frac{P}{C}$	0.634	0.122	0.102	0.002
	Dice	Harmonic mean of parent-child and parent-extension c-stem ratios	$\frac{2P}{C+E}=\frac{2(P/C)(P/E)}{(P/C)+(P/E)}$	0.391	0.198	0.101	0.004
	Pointwise Mutual Information	Pointwise MI between current and expansion schemes	$\log_2\left(\frac{P/V}{(C/V)(E/V)}\right)$	3.32	4.20	1.82	3.40
Statistical Tests	**Pearson's χ^2**	Nonparametric test for independence of random variables in categorical data	$\frac{P^2}{EC/V}+\frac{(E-P)^2}{E(V-C)/V}+\frac{(C-P)^2}{C(V-E)/V}+\frac{(V-E-C+P)^2}{(V-E)(V-C)/V}-V\xrightarrow{d}\chi_1^2$	777	2940	279	26.8
	Wald Test for Mean of Bernoulli	If the current and expansion schemes are independent, the Wald statistic converges to a standard normal: $N(0,1)$	$\frac{(P/V)-(C/V)(E/V)}{\sqrt{(P/V(1-P/V))/V}}$	27.2	12.5	8.64	1.57
	Likelihood Ratio of Bernoulli	Ratio of likelihood of expansion scheme given independence from current scheme to the likelihood of expansion scheme given dependence	$-2\log_e\left(\frac{\left(\frac{E}{V}\right)^P\left(1-\frac{E}{V}\right)^{C-P}\left(\frac{E}{V}\right)^{E-P}\left(1-\frac{E}{V}\right)^{V-E-C+P}}{\left(\frac{P}{C}\right)^P\left(1-\frac{P}{C}\right)^{C-P}\left(\frac{E-P}{V-C}\right)^{E-P}\left(1-\frac{E-P}{V-C}\right)^{V-E-C+P}}\right)$	3410	803	174	9.57

Figure 3.13: Six metrics which might gauge the paradigmatic unity of parent schemes during ParaMor's search of a vertical morphology scheme network. Each row describes a single metric. The top three metrics are heuristic measures of paradigmatic coherence. The bottom three metrics treat the current and expansion schemes as random variables and test their statistical correlation. In the Formula column: C, E, and P are the c-stem counts of the **C**urrent, **E**xpansion, and **P**arent schemes respectively, while V is the size of the corpus **V**ocabulary. An expansion scheme heads each of the final four columns, which each contain the value of that row's metric applied from the **a.o.os** scheme of Figure 3.12 to that column's expansion scheme. The mini-table at left shows where C, E, P, and V fall in a 2x2 table of c-stem counts, a la Figure 3.12.

P	E
C	V

upper limit to be the corpus vocabulary size. With this upper limit in hand, the maximum likelihood estimate of the probability that a c-stem will occur in a given scheme, *S*, is the count of the adherent c-stems in *S* over the size of the corpus vocabulary, or, in the notation of Figure 3.13, *C*/*V*. Note that the joint probability of finding a c-stem in the current scheme and in the expansion scheme is just the probability of a c-stem appearing in the parent scheme.

The first parent-evaluation metric that makes use of the probabilistic view of c-stem occurrence in schemes is pointwise mutual information. The pointwise mutual information between values of two random variables measures the amount by which uncertainty in the first variable changes when a value for the second has been observed. In the context of morphological schemes, the pointwise mutual information registers the change in the uncertainty of observing the expansion c-suffix when the c-suffixes in the current scheme have been observed.

The formula for pointwise mutual information between the current and expansion schemes is given on the third row of Figure 3.13. Like the dice measure, the pointwise mutual information identifies a large difference between the **amente** parent and the **ar** parent (reference the final columns of Figure 3.13). As Manning and Schütze (1999, p. 181) observe, however, pointwise mutual information increases as the number of observations of a random variable decrease. And since the expansion schemes **amente** and **ualidad** have comparatively low c-stem counts, the pointwise mutual information score is higher for the **amente** and **ualidad** parents than for the truly paradigmatic **as**—undesirable behavior for a metric guiding a search that must identify productive, and therefore likely frequent, paradigmatic suffixes.

Three Statistical Tests for Parent-Evaluation Metrics

While the three heuristic parent-evaluation metrics, namely parent-child c-stem ratios, dice similarity, and pointwise mutual information scores, seem intuitively reasonable, it would be theoretically appealing if ParaMor could base an upward search decision on a statistical test of a parent's worth. Just such statistical tests can be defined by viewing each c-stem in a scheme as a successful trial of a boolean random variable.

Taking the view of schemes as boolean random variables, the joint distribution of pairs of schemes can be tabulated in 2x2 grids. The grids beneath the four expansion schemes of Figure 3.12 hold the joint distribution of the **a.o.os** scheme and the respective expansion scheme. The first column of each table contains counts of adherent c-stems that occur with all the c-suffixes in the current scheme; While the second column contains an estimate of the number of c-stems which do not form corpus words with each c-suffix of the current child scheme. Similarly, the table's first row contains adherent counts of c-stems that occur with the expansion c-suffix; And the second row gives estimates for the number of c-stems which do not co-occur with the expansion suffix. Consequently, the cell at the intersection of the first row and first column contains the adherent stem count of the parent scheme.

The bottom row and the right-most column of each 2x2 table contain marginal adherent counts. In particular, the bottom cell of the first column contains the count of all the c-stems that occur with all the c-suffixes in the current scheme. In mirror image, the right-most cell of the first row contains the adherent count of all c-stems which occur with the expansion c-suffix. The corpus vocabulary size, as the estimate of the total number of c-stems, is the marginal of the marginal c-stem counts, and appears in the bottom right-hand corner of each 2x2 table. A mini-table at the bottom left of Figure 3.13 summarizes where to find the c-stem counts for the current, parent, and expansion schemes together with the estimate of vocabulary size in a 2x2 joint-distribution grid.

Treating sets of c-suffixes as boolean random variables, we must ask what measurable property of random variables might indicate that the c-suffixes of the current scheme and the c-suffix of the expansion scheme belong to the same paradigm. One answer is correlation. As described both earlier in this section as well as in Section 3.2.1, suffixes which belong to the same paradigm are likely to have occurred attached to the same stems—this co-occurrence is statistical correlation. Think of a big bag containing all possible c-stems. We reach our hand in, draw out a c-stem, and ask: Did the c-suffixes of the current scheme all occur attached to this c-stem? Did the expansion c-suffix occur with this c-stem? If the expansion c-suffix belongs to the same paradigm as the c-suffixes in

the current scheme, then the answer to both of these questions will often be the same, and the random variables will be correlated.

A number of standard statistical tests are designed to detect when two random variables are correlated. In designing ParaMor's parent-evaluation metric, three statistical tests were examined:

1. Pearson's χ^2 test,
2. The Wald test for the mean of a Bernoulli population, and
3. A likelihood ratio test for independence of Binomial random variables

Pearson's χ^2 test is a nonparametric test designed for categorical data, that is, data in which each observed trial point can be categorized as belonging to one of a finite number of types. Pearson's test compares the expected number of occurrences of each category with the observed number of occurrences of that category. In a 2x2 table, such as the tables of c-stem counts in Figure 3.12, the four central cells in the table are the categories. Pearson's χ^2 test statistic relies on the observation that if two random variables are independent, then the expected number of observations in each cell is the product of the marginal probabilities along that cell's row and column. Pearson's test statistic for a 2x2 table, given in the fourth row of Figure 3.12, converges to the χ^2 distribution as the size of the data increases. (DeGroot, 1986 p. 536).

The second statistical test investigated for ParaMor's vertical scheme search is a Wald test of the mean of a Bernoulli population (Casella and Berger, 2002 p. 493). This Wald test compares the observed number of c-stems in the parent scheme to the number which would be expected if the child c-suffixes and the expansion c-suffixes were independent. When the current and expansion schemes are independent, the central limit theorem implies that the statistic given in the fifth row of Figure 3.13 converges to a standard normal distribution.

Since the sum of Bernoulli random variables is a Binomial distribution, we can view the random variable which corresponds to any particular scheme as a Binomial. This is the view taken by the final statistical test ParaMor considers as a potential parent-evaluation metric for the initial bottom-up scheme search. In this final test, the random

variables corresponding to the current and expansion schemes are measured for independence using a likelihood ratio statistic from Manning and Schütze (1999, p. 172). When the current and expansion schemes are not independent, then the occurrence of a c-stem, *t*, in the current scheme will affect the probability that *t* appears in the expansion scheme. On the other hand, if the current and expansion schemes *are* independent, then the occurrence of a c-stem, *t*, in the current scheme will *not* affect the likelihood that *t* occurs in the expansion scheme. The denominator of the formula for the likelihood ratio test statistic, given in the final row of Figure 3.13, describes the likelihood of current and expansion schemes which are not independent; while the numerator gives the independent case. Taking two times the negative log of the ratio of these likelihoods produces a statistic that is χ^2 distributed.

One caveat, both the likelihood ratio test and Pearson's χ^2 test only assess the independence of the current and expansion schemes, they cannot disambiguate between random variables which are positively correlated and variables which are negatively correlated. When c-suffixes are negatively correlated it is extremely likely that they do *not* belong to the same paradigm. Clearly, ParaMor's search strategy should not move to parent schemes whose expansion c-suffix is negatively correlated with the c-suffixes of the current scheme. Negative correlation occurs when the observed frequency of c-stems in a parent scheme is less than the predicted frequency assuming that the current and expansion c-suffixes are independent. Thus, when evaluating parent schemes with either the likelihood ratio test or Pearson's χ^2 test, ParaMor explicitly checks for negative correlation.

Returning to Figure 3.13, the values of the three statistical tests for the four parents of the **a.o.os** scheme suggest that the tests are generally well behaved. For each of the tests, a larger score indicates that an expansion scheme is more likely to be correlated with the current scheme—although, again, comparing the absolute scores of one test to the numeric values from another test is meaningless. All three statistical tests correctly score the unproductive derivational **ualidad** scheme as the least likely of the four expansion schemes to be correlated with the current scheme. And each test gives a large margin of difference between the **amente** and the **ar** parents. The only obvious misbehavior of any

of these statistical tests is that Pearson's χ^2 test ranks the productive but derivational **amente** parent as more likely than the inflectional **as** parent to be correlated with the current scheme.

An Empirical Comparison of Parent-Evaluation Metrics

To quantitatively compare parent-evaluation metrics at their ability to identify paradigmatically coherent schemes, ParaMor performed an oracle experiment. This oracle experiment places on an even playing field all six parent-evaluation metrics summarized in Figure 3.13. Looking at Spanish data, the oracle evaluation assessed each metric at its ability to identify schemes in which every c-suffix is string identical to a suffix of some single inflectional paradigm. The inflectional paradigms of Spanish used as the oracle answer key in this experiment were hand complied from a standard Spanish textbook (Gordon and Stillman, 1999), and are detailed in Appendix A.

The methodology of the oracle experiment that ParaMor used to quantitatively compare parent-evaluation metrics is as follows: ParaMor visited every scheme that contained only c-suffixes which model suffixes from a single inflectional paradigm—call such schemes *sub-paradigm schemes*. Each parent of a sub-paradigm scheme is either a sub-paradigm scheme itself, or else the parent's c-suffixes no longer form a subset of the suffixes of a true paradigm. The oracle experiment evaluated each metric at its ability to classify each parent of each sub-paradigm scheme as either being a sub-paradigm scheme itself or as introducing a non-paradigmatic c-suffix.

Notice that the oracle experiment does not directly evaluate parent metrics in the context of the greedy upward search procedure described in Sections 3.2.1 and 3.2.2. Following the methodology of this oracle experiment allows a direct comparison between parent-evaluation metrics: where ParaMor's greedy search is not guaranteed to visit identical sets of schemes when searching with different upward metrics or with different halting thresholds, the oracle experiment described here evaluates all metrics over the same set of upward decisions.

Also, the non-greedy methodology of this oracle experiment necessitated using a considerably smaller corpus than do other experiments that are reported in this thesis.

Figure 3.14 gives results of the oracle evaluation over a corpus of free-running Spanish newswire text containing 6,975 unique types. A small corpus is necessary because the oracle experiment visits *all* parents of *all* sub-paradigm schemes, and as is discussed in Section 3.2.4, a larger corpus creates a search space that is too large to fully instantiate.

To compare parent-evaluation metrics ParaMor must factor out threshold effects. Each metric's performance at identifying sub-paradigm schemes varies with the cutoff threshold below which a parent is believed to not be a sub-paradigm scheme. For example, when considering the c-stem ratio metric at a threshold of 0.5, ParaMor would take as a sub-paradigm scheme any parent that contains at least half as many c-stems as the current scheme. But if this threshold were lowered to 0.25, then a parent need have only one-quarter the number of c-stems as the child to pass for a sub-paradigm scheme.

Moreover, while each of the six metrics described in the previous section score each parent scheme with a real value, the scores are not normalized. The ratio and dice metrics produce scores between zero and one, Pearson's χ^2 test and the Likelihood Ratio test produce non-negative scores, while the scores of the other metrics can fall anywhere on the real line. But even the scores of metrics which lie in the same range are not comparable. Referencing Figure 3.13, the ratio and dice metrics, for example, can produce very different scores for the same parent scheme. Furthermore, while statistical theory can give a confidence level to the absolute scores of the metrics that are based on statistical tests, the theory does not suggest what confidence level is appropriate for the task of paradigm detection in scheme networks. The ratio, dice, and pointwise mutual information metrics lack even an interpretation of confidence.

To remove threshold effects and fairly compare parent-evaluation metrics, the oracle experiment performs a peak-to-peak comparison. The oracle evaluation measures the precision, recall, and F_1 of each metric over a range of threshold values relevant to that metric. And the maximum F_1 value each metric achieves is its final score.

Figure 3.14 reports the peak F_1 score for each of the six metrics presented in this section. Two results are immediately clear. First, all six metrics consistently outperform the baseline algorithm of considering every parent of a sub-paradigm scheme to be a sub-

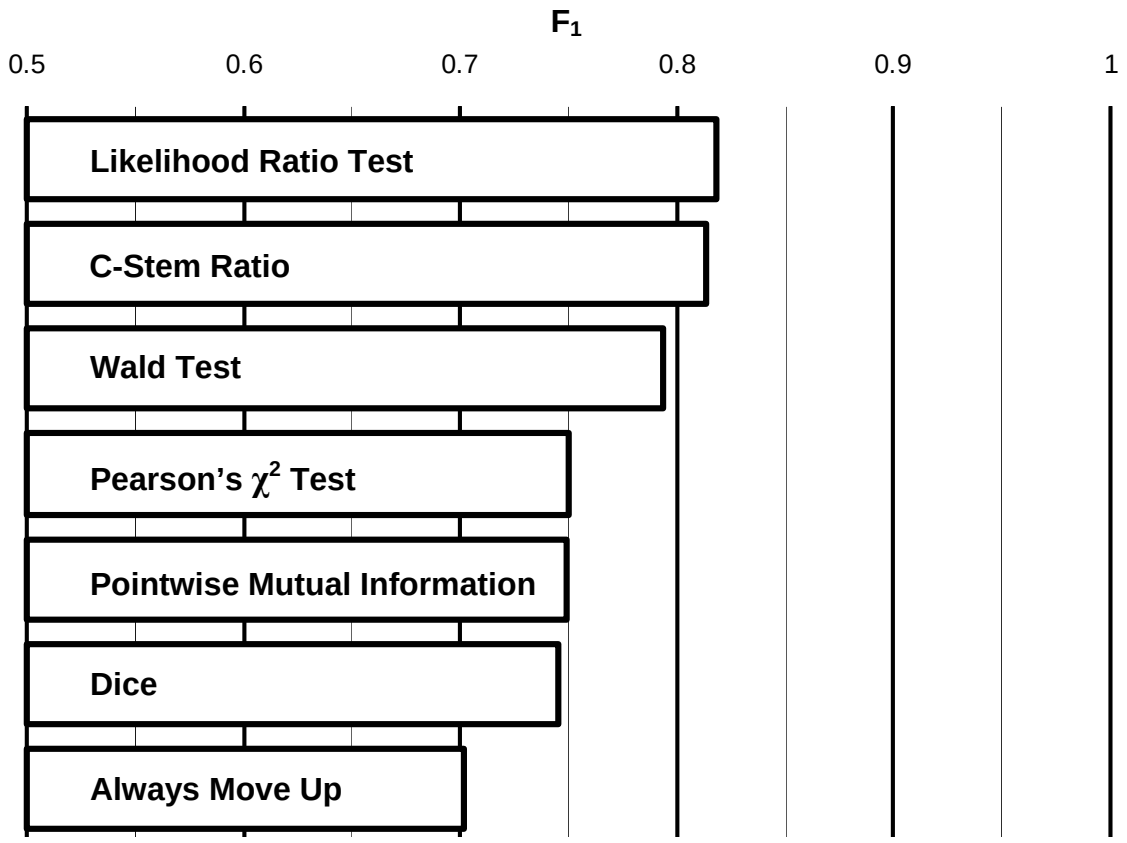

Figure 3.14: A small-scale oracle evaluation of six metrics at the task of identifying schemes where each c-suffix models a suffix in the same true inflectional paradigm. Each bar reports the peak F_1 of its metric over a range of cutoffs appropriate for that metric.

paradigm scheme. Although a seemingly weak baseline, the always-move-up rule correctly classifies the paradigmatic integrity of scheme parents with an F_1 score above 70%.

The second result evident from Figure 3.14 is that the most simple metric, the parent-child c-stem ratio, does surprisingly well, identifying parent schemes which contain no extra-paradigmatic suffixes just as consistently as more sophisticated tests, and outperforming all but one of the considered metrics. The primary reason the parent-child c-stem ratio performs so well appears to be that the ratio metric is comparatively robust when

data is sparse. In 79% of the oracle decisions that each metric faced, the parent scheme had fewer than 5 c-stems!

Interestingly, Hammarström (2006b) has also found that simple c-stem ratios are unexpectedly effectively at identifying suffixes that paradigmatically contrast. In the context of a stemming algorithm designed for information retrieval, Hammarström reports that, in assessing the c-stem coherence of sets of c-suffixes, he "has not had much success with standard vector similarity measures." Hammarström then turns to parent-child c-stem ratios to define a novel measure of the extent to which c-suffixes share c-stems. Hammarström's metric, which, unfortunately, this oracle evaluation does not investigate, looks only at parent-child c-stem ratios that occur at the bottom of scheme networks, namely, between paradigmatic level 2 and level 1 schemes.

In addition to strong performance in this oracle evaluation, the parent-child c-stem ratio has a second, unrelated, advantage over other metrics: simplicity of computation. In taking each upward step, ParaMor's bottom-up search procedure, described in Sections 3.2.1 and 3.2.2, greedily moves to that parent with the highest score. Hence, to find the best greedy parent, ParaMor need not necessarily compute the scores of parent schemes, all ParaMor must do is identify which parent *would* have the highest score. As the c-stem ratio metric does not incorporate the frequency of the expansion c-suffix, and as the number of c-stems in the current scheme does not change, the parent scheme with the largest c-stem ratio is always that parent with the most c-stems—an easily computable statistic.

On the basis of both the sturdy performance in the oracle evaluation, as well as the simplicity and speed of identifying the parent with the largest c-stem ratio, all further experiments in this thesis use the parent-child c-stem metric to guide ParaMor's vertical search.

Setting the Halting Threshold in ParaMor's Bottom-Up Search

Having settled on the parent-child c-stem ratio as the parent-evaluation metric of choice, ParaMor must next select a threshold value at which to halt upward search paths. The oracle experiment described in this section was designed to ascertain which parent-

evaluation metric most accurately identifies paradigmatic parents. Somewhat in contrast, ParaMor's greedy search algorithm, described in Sections 3.2.1 and 3.2.2, prizes caution ahead of accuracy. As discussed in Section 3.2.1, ParaMor's greedy search takes a separate upward path from all individual single-c-suffix schemes. Pursuing many paths in parallel, it is likely that most c-suffixes that model a true suffix will be selected as part of some path, thereby yielding high suffix recall. ParaMor then relies on a clustering algorithm, described in Chapter 4, to bring together paradigmatically related c-suffixes that only appear in separate search paths. A cautious initial search helps ParaMor to keep initially selected models of partial paradigms largely free of non-paradigmatic c-suffixes.

The threshold value at which the parent-child c-stem ratio achieved its peak F_1 in the oracle experiment of Figure 3.14 is 0.05. At this threshold, for ParaMor's greedy search to accept a parent scheme, the parent need only contain one-twentieth the number of c-stems as the current scheme. A qualitative examination of ParaMor's selected schemes suggests a threshold of 0.05 is not cautious enough; Over a corpus of 50,000 Spanish types, at the 0.05 threshold, ParaMor's greedy search selects many schemes that include both inflectional paradigmatic c-suffixes and c-suffixes that model only marginally productive derivational suffixes. Hence, the remainder of this thesis sets the parent-child c-stem ratio threshold at the more cautious value of 0.25.

It is possible that a threshold value of 0.25 is sub-optimal for paradigm identification. And future work should more carefully examine the impact that varying this threshold has on morphological segmentation of words (Chapter 5). However, the quantitative evaluations of Chapter 6 will show that the current setting of 0.25 leads to morphological segmentations of a diverse set of natural languages that out-perform the segmentations of state-of-the-art unsupervised morphology induction systems.

3.3 Summarizing the Search for Candidate Paradigms

This chapter has presented the strategy that ParaMor employs to identify initial candidate paradigms. This strategy has three main steps. First, ParaMor organizes candidate stems and candidate suffixes into natural groupings, or schemes, that model potential in-

flectional paradigms. Second, ParaMor relates the scheme groupings to one another, in the process forming a network of candidates. And third, ParaMor searches the candidate paradigm network in a recall-centric fashion for schemes which individually cover diverse portions of the paradigms of a language.

Although, as the next chapter will show quantitatively, ParaMor's initial paradigm search successfully uncovers most suffixes of languages like Spanish, the output of the search is still far from a tight model of paradigm structure. With ParaMor's search focused on recall, many of the initially selected candidate paradigms are erroneously selected. And of those which are correct, many redundantly cover overlapping portions of the same paradigms. Overcoming these two shortcomings of the initial search procedure is the topic of the next chapter.

Chapter 4: Clustering and Filtering of Initial Paradigms

This chapter builds on the initial models of paradigms that are selected by ParaMor's greedy bottom-up search strategy of Chapter 3. Taking in the initial set of scheme models, this chapter consolidates and purifies a focused set of paradigm models. And, in turn, the algorithms described in Chapter 5 wield these focused models to segment words into morphemes.

4.1 From Schemes to Comprehensive Models of Paradigms

The bottom-up search strategy presented in Chapter 3 is a solid first step toward identifying useful models of productive inflectional paradigms. However, as Section 3.2.1 explains, ParaMor's greedy bottom-up search strategy narrowly focuses on finding partial models of paradigms that, in aggregate, concentrate on recall. ParaMor's recall-centric strategy of starting upward paths from *all* individual c-suffixes inevitably seeds some paths with c-suffixes which do not model suffixes. And while ParaMor's initial scheme-

models jointly recover many inflectional suffixes, individual schemes will likely cover only portions of full linguistic paradigms. Section 4.1.1 illustrates, by way of an extended example, the successes and shortcomings of ParaMor's initially selected schemes, while Section 4.1.2 outlines the strategies that ParaMor takes to mitigate the shortcomings. Sections 4.2, 4.3, and 4.4 then present each mitigation strategy in detail. And finally, Sections 4.5 and 4.6 analyzes the paradigm models that ParaMor ultimately produces.

4.1.1 A Sample of Initially-Selected Schemes

To better see the strengths of the schemes that ParaMor initially selects as paradigm models, and to motivate the steps ParaMor takes to overcome gaps in the initial models, Figure 4.1 presents a range of schemes selected during a typical run of ParaMor's bottom-up search procedure, from Chapter 3. Each row of Figure 4.1 lists a scheme selected while searching over a Spanish newswire corpus of 50,000 types using the parent-child c-stem ratio metric at a halting threshold of 0.25 (see the final sub-section of Section 3.2.5). On the far left of Figure 4.1, the **RANK** column states the ordinal rank at which that row's scheme was selected during the search procedure: the **Ø.s** scheme was the terminal scheme of ParaMor's 1st upward search path, **a.as.o.os** the terminal scheme of the 2nd path, **ido.idos.ir.iré** the 1592nd, etc. The right four columns of Figure 4.1 present raw data on the selected schemes, giving the number of c-suffixes in that scheme, the c-suffixes themselves, the number of adherent c-stems of the scheme, and a sample of those c-stems.

Between the rank on the left, and the scheme details on the right, are columns which categorize the scheme on its success, or failure, to model a true paradigm of Spanish. Appendix A lists the inflectional paradigms of Spanish morphology. A dot appears in the columns marked **N**, **ADJ**, or **VERB** if multiple c-suffixes in a row's scheme clearly represent suffixes in a paradigm of that part of speech. The verbal paradigm is further broken down by inflection class: **ar**, **er**, and **ir**. A dot appears in the **DERIVATION** column if at least one c-suffix of a scheme models a derivational suffix.

The remaining six columns of Figure 4.1 classify the correctness of each row's scheme. The **GOOD** column of Figure 4.1, for example, is marked if the c-suffixes in a scheme take the surface form of true suffixes. Initially selected schemes in Figure 4.1 that correctly capture real paradigm suffixes are the 1st, 2nd, 4th, 5th, 12th, 30th, 40th, 127th, 135th, 400th, 1592nd, and 2000th selected schemes.

Most true inflectional suffixes of Spanish are modeled by some scheme that is selected during the run of ParaMor's initial search that is presented in Figure 4.1: The bottom-up search identifies partial paradigms which, between them, contain 91% of all string-unique suffixes of the Spanish verbal inflectional paradigms, as summarized in Appendix A. If we ignore as undiscoverable all suffix strings which occurred at most once in the Spanish newswire corpus, ParaMor's coverage jumps to 97% of unique verbal suffixes. In addition to verbal suffixes, ParaMor identifies schemes which model:

1. Each of the two phonologically determined inflection classes that express ***Number*** on nouns: the 1st selected scheme models **Ø.s**, while the 4th selected scheme models **Ø.es**; and
2. The full adjectival cross-product paradigm of ***Gender*** and ***Number***, **a.as.o.os**, in the 2nd selected scheme.

But, while most true inflectional suffixes are modeled by some scheme selected in the initial search, ParaMor's initially selected schemes also display two broad shortcomings.

A First Shortcoming of Initial Schemes: Fragmentation

No single initially selected scheme comprehensively models all the suffixes of the larger Spanish paradigms. Instead, c-suffix models of paradigm suffixes are fragmented across large numbers of schemes. The largest schemes that ParaMor identifies from the newswire corpus are the 5th and 12th selected schemes: Shown in Figure 4.1, both of these schemes contain 15 c-suffixes which model suffixes from the **ar** inflection class of the Spanish verbal paradigm. But the full **ar** inflection class has 36 unique surface suffixes. In an agglutinative language like Turkish, the cross-product of several word-final para-

Rank	Model of: N	Adj	Verb: ar	er	ir	Derivation	Error: Good	Complete	Partial	Stem-Internal	Suffix-Internal	Chance	C-Suffixes		C-Stems	
1	●	●					●	●					2	Ø s	5501	apoyada barata hombro oficina reo …
2		●					●	●					4	a as o os	892	apoyad captad dirigid junt próxim …
3			●						●		●		15	Ø ba ban da das do dos n ndo r ra ron rse rá rán	17	apoya disputa lanza lleva toma …
4	●	●					●	●					2	Ø es	847	emir inseguridad orador pu ramon …
5			●				●		●				15	a aba aban ada adas ado ados an ando ar aron arse ará arán ó	25	apoy desarroll disput lanz llev …
10		●						●		●			5	ta tamente tas to tos	22	cier direc insóli modes sangrien …
11			●						●		●		14	Ø ba ción da das do dos n ndo r ron rá rán ría	16	acepta celebra declara fija marca …
12			●				●		●				15	a aba ada adas ado ados an ando ar aron ará arán e en ó	21	apoy declar enfrent llev tom …
20		●						●		●		●	6	Ø l ra ras ro ros	8	a ca coste e ente gi o pu
30				●	●		●		●				11	e en ida idas ido idos iendo ieron ió ía ían	16	cumpl escond recib transmit vend …
40	●	●				●	●	●					7	Ø es idad ización izado izar mente	8	actual final natural penal regular …
100												●	8	Ø a an e ea en i ino	9	al c ch d g p s t v
127				●			●		●				9	e en er erá ería ido ieron ió ía	11	ced deb ofrec pertenec suspend …
135				●	●		●		●				10	a e en ida ido iendo iera ieron ió ía	12	assist cumpl ocurr permit reun un …
200						●				●			4	tal tales te tes	5	acciden ambien continen le pos
300						●						●	4	o os ual uales	7	act cas concept event grad man us
400			●			●	●		●				8	a aciones ación ados an ar ativas ó	10	administr conmemor estim investig …
1000				●					●	●			8	ce cen cer cerán cido cieron ció cía	9	apare estable ofre pertene ven …
1592					●		●		●				4	ido idos ir iré	6	conclu cumpl distribu exclu reun …
2000				●	●		●		●				4	e en ieron iesen	5	aparec crec impid invad pud
3000												●	2	Ø zano	3	li lo man
4000												●	2	icho io	4	b capr d pred
5000			●						●	●			2	egará ega	3	desp entr ll
…													…	…	…	…

Figure 4.1: Schemes selected by ParaMor's initial bottom-up search algorithm over a Spanish corpus of 50,000 types. While some selected schemes contain c-suffixes that correctly model paradigmatic suffixes, others are incorrect collections of word final strings.

digms may have an effective size of hundreds or thousands of suffixes, and ParaMor will only identify a minute fraction of these in any one scheme.

In Figure 4.1, the **COMPLETE** column is marked when a scheme contains separate c-suffixes that correspond to each suffix of a paradigm or paradigm cross-product. On the other hand, if a scheme's c-suffixes model suffixes of a paradigm of Spanish, but manage to model only a portion of the full paradigm, then Figure 4.1 has a dot in the **PARTIAL** column. Among the many schemes in Figure 4.1 which faithfully describe significant partial fractions of legitimate paradigms are the 5th, 12th, and 400th selected schemes. These three schemes each contain c-suffixes which clearly model suffixes from the verbal **ar** paradigm—but each contains c-suffixes that correspond to only a subset of the suffixes in the full **ar** inflection class. Notice that while some legitimate inflectional suffixes occur in only one **ar** scheme, e.g. **aban** and **arse** in the 5th selected scheme, other c-suffixes appear in two or more schemes that model the **ar** paradigm, e.g. **a**, **ados**, **ó**. Indeed, the **ados** c-suffix occurs in 31 schemes that were selected during this run of ParaMor's initial search. The search paths that identified these 31 schemes each originate from a distinct initial c-suffix, including such c-suffixes as: **an**, **en**, **ación**, **amos**, etc.

Separate scheme patchworks cover the other inflection classes of Spanish verbs as well. For example, schemes modeling portions of the **ir** inflection class include the 30th, 135th, 1592nd, and 2000th selected schemes. Consider the 1592nd scheme, which contains four c-suffixes. Three of these c-suffixes, **ido**, **idos**, and **ir**, occur in other schemes selected during the initial search, while the uncommon *1st Person Singular Future Tense* suffix **iré** is unique to the 1592nd selected scheme—in all the schemes selected during this run of ParaMor's initial bottom-up search, the c-suffix **iré** occurred in only one. ParaMor's recall-centric search has correctly identified the **iré** suffix, but as yet **iré** is isolated from most other c-suffixes of the **ir** paradigm.

A Second Shortcoming of Initial Schemes: Poor Precision

The second broad shortcoming of ParaMor's initial search strategy, apparent from Figure 4.1, is simply that many schemes do not satisfactorily model Spanish morphological suffixes. The vast majority of schemes with this second shortcoming belong to one of

two sub-types. The first sub-type of unsatisfactory paradigm model comprises schemes which systematically misanalyze word forms. These systematically incorrect schemes come in two flavors: a scheme will either hypothesize morpheme boundaries in its licensing words that fall internal to true stems *or* a scheme will propose boundaries that are consistently internal to suffixes.

Schemes which misanalyze morpheme boundaries in Figure 4.1 are marked in the **ERROR: STEM-INTERNAL** or **ERROR: SUFFIX-INTERNAL** columns, and comprise the 3rd, 10th, 11th, 20th, 200th, 1000th, and 5000th selected schemes. Of these, the 3rd and 11th selected schemes place a morpheme boundary at suffix-internal positions, truncating the full suffix forms: Compare the 3rd and 11th selected schemes with the 5th and 12th. In symmetric fashion, a significant fraction of the c-suffixes in the 10th, 20th, 200th, 1000th, and 5000th selected schemes hypothesize morpheme boundaries for their licensing word forms internal to real Spanish stems. In placing morpheme boundaries internal to stems, these schemes inadvertently include the final characters of verb stems as leading characters on their c-suffixes. In a random sample of 100 Spanish schemes from the 8339 schemes which the initial search strategy selected, 48 schemes incorrectly placed morpheme boundaries stem-internally, while one scheme hypothesized morpheme boundaries at locations inside the suffixes of the scheme's licensing forms.

The second sub-type of unsatisfactory paradigm model exemplified in Figure 4.1 occurs when a scheme's c-suffixes are related not by belonging to the same paradigm, but rather by a chance string similarity of surface type. Schemes which arise from chance string collisions are marked in the **ERROR: CHANCE** column of Figure 4.1, and include the 20th, 100th, 3000th, and 4000th selected schemes. In the random sample of 100 schemes selected by ParaMor's initial bottom-up search, 40 were schemes produced from a chance similarity between word types.

These chance schemes are typically small along two distinct dimensions. First, the string lengths of the c-stems and c-suffixes of these chance schemes are often quite short. The longest c-stem of the 100th selected scheme is two characters long; while both the 100th and the 3000th selected schemes contain the null c-suffix, **Ø**, which has length zero. That ParaMor might erroneously select schemes because of their short c-stem and

c-suffix lengths is easily explained combinatorially: While the inventory of possible strings grows exponentially with the length of the string, there just aren't very many length-one or length-two strings. Within the restricted space of short strings, it should come as no surprise when a variety of (often short) c-suffixes happen to occur attached to the same set of very short c-stems.

Schemes arising through a chance string similarity of word types are small on a second dimension as well. Chance schemes typically contain few c-stems, and, by virtue of the details of ParaMor's search procedure (see Section 3.2.2), even fewer c-suffixes. For example, the 3000th selected scheme contains just three c-stems and two c-suffixes. The evidence for this 3000th scheme arises, then, from a scant six (short) types, namely: **li**, a Chinese name; **lo**, a Spanish determiner and pronoun; **man**, part of an abbreviation for 'Manchester United' in a listing of soccer statistics; **lizano**, a Spanish name; **lozano**, a Spanish word meaning 'leafy'; and **manzano**, Spanish for 'apple tree'.

Schemes formed from a chance string similarity of a few types, such as the 3000th selected scheme, are particularly prevalent among schemes chosen later in the search procedure, where search paths originate from level 1 schemes whose single c-suffix is less frequent. Although there are less frequent c-suffixes that do correctly model portions of true paradigms (including the c-suffix **iré**, which led to the paradigmatically coherent 1592nd selected scheme, see above) the vast majority of less frequent c-suffixes do not model true suffixes. And because the inventory of word final strings in a moderately sized corpus is enormous, some few of the many available c-suffixes happen to be interchangeable with some other c-suffix on some few (likely short) c-stems of the corpus.

4.1.2 ParaMor's Paradigm-Processing Pipeline

This chapter describes the algorithms ParaMor adopts to remedy the two shortcomings of ParaMor's initially selected schemes that were identified in Section 4.1.1. To consolidate the patchwork modeling of paradigms and to corral free c-suffixes into structures which more fully model complete paradigms, ParaMor adapts an unsupervised clustering algorithm to automatically group related schemes. Meanwhile, to remove schemes which

fail to model true suffixes, ParaMor takes a two pronged approach: First, a clean-up of the training data reduces the incidence of chance similarity between strings, and second, targeted filtering algorithms identify and discard schemes which likely fail to model paradigms.

To organize these additional steps of paradigm identification, ParaMor adopts a pipeline architecture. ParaMor's initial morphology network search algorithm, described in Chapter 3, becomes one step in this pipeline. Now ParaMor must decide where to add the pipeline step that will cluster schemes which model portions of the same paradigm, and where to add steps that will reduce the incidence of incorrectly selected schemes.

At first blush, it might seem most sound to place steps that remove incorrectly selected schemes ahead of any scheme-clustering step—after all, why cluster schemes which do not model correct suffixes? At best, clustering incorrect schemes seems a waste of effort; at worst, bogus schemes might confound the clustering of legitimate schemes. But removing schemes before they are clustered has its own dangers. Most notably, a discarded correct scheme can never be recovered: If the distraction of incorrect schemes could be overcome, corralling schemes into monolithic paradigm models might safeguard individual useful schemes from imperfect scheme-filtering algorithms. And by the same token, scheme filters can also mistake incorrect schemes for legitimate models of paradigms. Hence, if a clustering algorithm could place together such misanalyses as the 3rd and 11th selected schemes from Figure 4.1, which both model the same incorrect morpheme boundaries in their licensing types, then clustering incorrect schemes might actually facilitate identification and removal of misanalyzed schemes.

As Section 4.3 explains, ParaMor's clustering algorithm can, in fact, accommodate schemes which hypothesize incorrect morpheme boundaries, but has more difficulty with non-paradigmatic schemes which are the result of chance string similarity. To retain a high recall of true suffixes within the framework of a pipeline architecture, ParaMor takes steps which reduce the inventory of selected schemes only when necessary. Section 4.2 describes a technique that vastly reduces the number of selected schemes which result from chance string similarity, while insignificantly impacting correctly selected schemes. Section 4.3 then describes ParaMor's scheme-clustering algorithm. And Section 4.4 pre-

sents two classes of filtering algorithm which remove remaining incorrectly selected, but now clustered, schemes.

4.2 Training Corpus Clean-Up

ParaMor's clustering algorithm (Section 4.3) is specifically tailored to leverage the paradigmatic structure of schemes, and, as such, is ill-suited to schemes which do not exhibit a regular paradigmatic alternation. Schemes which result from chance similarities in word forms pointedly lack such paradigmatic structure. Thus ParaMor seeks to remove chance schemes before the scheme-clustering step.

As mentioned in Section 4.1.1, the string lengths of the c-suffixes and c-stems of chance schemes are typically quite short. And if the c-suffixes and c-stems of a scheme are short, then the underlying types which license the scheme are also short. These facts suggest the simple data clean up step of excluding short types from the vocabulary that ParaMor uses to induce paradigms. As described momentarily, placing this simple word-length requirement on the paradigm-induction vocabulary virtually eliminates the entire category of chance scheme.

For all of the languages considered in this thesis ParaMor has raw text corpora available that are much larger than the 50,000 types used for paradigm induction. Consequently, for the experiments reported in this thesis, ParaMor does not merely remove short types from the induction vocabulary, but replaces each short word with a new longer word. ParaMor's word-to-morpheme segmentation algorithm, presented in Chapter 5, is independent of the set of types from which schemes and scheme-clusters are built. Consequently, removing short types from training does not preclude these same short types from being analyzed as containing multiple morphemes during segmentation.

The string length below which words are removed from the paradigm-induction vocabulary is a free parameter. ParaMor is designed to identify the productive inflectional paradigms of a language. Unless a productive paradigm is restricted to occur only with short stems, a possible but unusual scenario (as with the English adjectival comparative, c.f. *faster* but **exquisiter*), we can expect a productive paradigm to occur with a reason-

able number of longer stems in a corpus. Hence, ParaMor needn't be overly concerned about discarding short types. A qualitative examination of Spanish data suggested excluding types 5 characters or less in length. All experiments reported in this thesis which exclude short types only permit types longer than this 5 character cutoff.

An Evaluation of Schemes from a Vocabulary that Excludes Short Types

When restricted to a corpus containing types longer than 5 characters in length, the schemes that ParaMor selects during the initial search phase are remarkably similar to the schemes that ParaMor's search algorithm selects over a corpus containing types of all lengths—except for the notable absence of incorrect chance schemes. In a random sample of 100 schemes selected by ParaMor over a type-length restricted corpus of Spanish newswire containing 50,000 unique word forms, only 2 schemes resulted from a chance similarity of word forms—this is down from 40 chance schemes in a random sample of 100 schemes selected from a corpus unrestricted for type length.

The 3000th selected scheme, **Ø.zano**, shown in Figure 4.1 and discussed in the final sub-section of Section 4.1.1, is an example of the kind of scheme ParaMor's search algorithm no longer selects once a type-length restriction is in place. The **Ø.zano** scheme contains just three c-stems: **li**, **lo**, and **man**. Because the **Ø.zano** scheme contains the null c-suffix, **Ø**, the three surface word types, **li**, **lo**, and **man**, are among those that license this scheme. And because all three of these word types are not longer than 5 characters in length, all three are excluded from the Spanish paradigm induction corpus. Removing these three types strips the **Ø.zano** scheme of all evidence for the **Ø** c-suffix, and the 3000th scheme cannot be selected by the search algorithm.

In all, ParaMor's search algorithm selects 1430 fewer schemes, 6909 vs. 8339 when training over a type-length restricted corpus. However, including additional long types in the paradigm-induction vocabulary can actually increase the fragmentation of true paradigms across schemes. For example, the number of schemes that contain the c-suffix **ados**, which models a suffix from the **ar** verbal paradigm of Spanish, increases from 31 to 40 when restricting the paradigm-induction corpus to contain no short word types.

Because the training corpus has changed, schemes selected from a corpus restricted for type length are often not identical to schemes selected from an unrestricted corpus. But, in Spanish, ParaMor continues to identify schemes which model the major inflection classes of nouns, adjectives, and verbs. Moreover, with one notable exception, the schemes which model these major paradigms of Spanish contain numbers of c-suffixes that are similar to the numbers of c-suffixes of corresponding schemes induced over an unrestricted corpus.

From the type-length *un*restricted corpus, ParaMor directly models the **Ø.es** inflection class of ***Number*** on Spanish nouns with the scheme **Ø.es** (see Figure 4.1 on p. 100); But from the corpus that is restricted for type-length, ParaMor only models the two suffixes **Ø** and **es** of this less common nominal paradigm in combination with derivational suffixes: in schemes like **Ø.es.idad.idades.mente**. In Spanish, although nouns, such as **fin** 'end', pluralize by adding an **es** suffix, **fines**, many words which end in **es** cannot strip off that **es** to form a new surface form. For example, to form the *Singular* of the *Plural* adjective **grandes** 'big', only the **s** is removed, yielding the word **grande**. It so happens that in the type-length restricted Spanish corpus only 751 of the 3045 word strings which end in **es** can remove that **es** to form a new word that occurred in the Spanish corpus. And 751 out of 3045 is 24.7%—just short of the 25.0% c-stem ratio threshold used in the upward search algorithm. Additionally, since the ParaMor search strategy does not begin any search path from the **Ø** scheme, the only search paths which include the c-suffixes **Ø** and **es** necessarily begin from some third c-suffix—like the fairly productive derivation suffix **idad**. The **idad** suffix is the Spanish analogue of the English derivational suffix **ity**. As Chapter 5 discusses, ParaMor can still analyze the morphology of inflected forms despite the distracting presence of derivational suffixes in a scheme. Thus, although ParaMor is no longer able to identify the **Ø.es** scheme in isolation, the type-length restriction does not cause ParaMor to lose the ability to morphologically analyze Spanish nouns that mark *Plural* with **es**.

4.3 Clustering of Partial Paradigms

With the careful weeding of the paradigm-induction corpus described in the previous section, ParaMor largely avoids selecting schemes which arise through chance string similarity. And with the non-paradigmatic chance schemes out of the way, ParaMor is free to apply a clustering algorithm so as to group paradigmatically related schemes that hypothesize compatible morpheme boundaries.

As an unsupervised algorithm, ParaMor must use an unsupervised clustering algorithm to merge schemes. A variety of unsupervised clustering algorithms exist, from k-means to self-organizing kohonen maps. ParaMor's current clustering algorithm is an adaptation of bottom-up agglomerative clustering. Two reasons underlie the choice of bottom-up agglomerative clustering. First, agglomerative clustering is simple, there are just two steps:

1. Clustering begins with each item, i.e. scheme in this application, as its own separate cluster; and
2. At each time step, unless a halting criterion is met, that pair of clusters which is most similar is merged to form a new cluster.

The second reason ParaMor adopts bottom-up agglomerative clustering is that the algorithm produces a tree structure that can be examined by hand. And as the remainder of this section describes, careful examination of scheme-cluster trees directly led to adaptations of vanilla agglomerative clustering which accommodate the unique structure of paradigmatic schemes.

4.3.1 Three Challenges Face any Scheme-Clustering Algorithm

ParaMor's scheme-clustering algorithm must address three challenges that arise when schemes are the items being clustered. Two of the three challenges concern intrinsic properties of linguistic paradigms; and two of the three involve schemes as computational models of paradigms. First, the purely linguistic challenge: morphological syncre-

tism. Common cross-linguistically, syncretism occurs when distinct paradigms in a single language contain surface-identical suffixes. Syncretism implies there will be schemes which should not be merged even though they contain some identical c-suffixes.

Second, ParaMor faces the wholly computational challenge of deciding when to halt clustering. Whenever possible, an unsupervised algorithm, such as ParaMor, would like to avoid introducing free parameters that must somehow be tuned. To decide when to stop merging clusters, bottom-up agglomerative clustering typically uses a free parameter: Clustering ends when the similarity score between the pair of most similar clusters falls below a threshold. To avoid introducing this tunable halting parameter, it is desirable to devise a threshold-free method tailored to the peculiarities of paradigmatic schemes.

The third challenge ParaMor must overcome has both linguistic and computational roots: competing schemes may hypothesize rival morpheme boundaries in a common set of surface types, but such competing schemes should not be merged into the same cluster as they are distinct models of morphological structure. Different individual schemes hypothesize different morpheme boundaries in a single word type both for the computational reason that ParaMor does not know, a priori, where the correct morpheme boundaries lie, but also because, in natural language, morphological agglutination allows a single word type to legitimately contain more than one morpheme boundary. The following subsections motivate and present adaptations to the basic bottom-up agglomerative clustering algorithm that address these three challenges.

The First Challenge: Syncretism

Some examples will help elucidate the first challenge facing ParaMor's clustering algorithm: surface identical c-suffixes in distinct paradigms, or syncretism. Figure 4.2 lists six syncretic schemes taken from Figure 4.1: As a refresher, the schemes in Figure 4.1 were selected by ParaMor's initial search over a Spanish corpus of 50,000 types unrestricted for type length.

In Spanish, verbs that belong to the **er** paradigm systematically share many inflectional suffixes with verbs of the **ir** paradigm. In Figure 4.2 the 30^{th}, 135^{th}, and 2000^{th} se-

RANK	MODEL OF						GOOD	COMPLETE	PARTIAL	ERROR			C-SUFFIXES		C-STEMS	
	N	ADJ	VERB ar	VERB er	VERB ir	DERIV.				STEM	SUFFIX	CHANCE				
12			●				●		●				**15**	a aba ada ... arán e en ó	**21**	apoy declar enfrent llev …
30				●	●		●		●				**11**	e en ida idas ido idos ...	**16**	cumpl escond recib vend …
127				●			●		●				**9**	e en er erá ería ido ieron ...	**11**	ced deb ofrec pertenec …
135				●	●		●		●				**10**	a e en ida ido iendo iera ...	**12**	assist cumpl ocurr permit …
1592					●		●		●				**4**	ido idos ir iré	**6**	conclu cumpl distribu reun …
2000				●	●		●		●				**4**	e en ieron iesen	**5**	aparec crec impid invad pud

Figure 4.2: An excerpt from Figure 4.1. Six schemes selected by ParaMor's initial bottom-up search algorithm that exhibit syncretism in the **ar**, **er**, and **ir** inflection classes of the Spanish verbal paradigm.

lected schemes all contain only c-suffixes which model suffixes found in both the **er** and the **ir** inflection classes. But grammars of Spanish still distinguish between an **er** and an **ir** inflection class because **er** and **ir** verbs do not share *all* inflectional suffixes. In Figure 4.2, the 127th selected scheme contains the c-suffixes **er**, **erá**, and **ería** which all only occur on **er** verbs, while the 1592nd selected scheme contains the c-suffixes **ir** and **iré** which only occur attached to **ir** verbs. A scheme-clustering algorithm must not place the 127th and 1592nd selected schemes into the same cluster, but should instead produce distinct clusters to model the **er** and **ir** inflection classes.

More insidious than the suffix overlap between the **er** and **ir** inflection classes of Spanish verbs, is the overlap between the verbal **ar** inflection class on the one hand and the **er** and **ir** inflection classes on the other. While shared surface suffixes in the **er** and **ir** inflection classes consistently mark identical sets of morphosyntactic features, most suffixes whose forms are identical across the **ar** and the **er**/**ir** inflection classes mark *different* morphosyntactic features. The *Present Tense* suffixes **as**, **a**, **amos**, and **an** mark various ***Person-Number*** features in the *Indicative Mood* on **ar** verbs but *Subjunctive Mood* on **er** and **ir** verbs. Conversely, **es**, **e**, **emos** (**imos** on **ir** verbs), and **en** mark *Indicative Mood* on **er** verbs, but these **e**-forms mark *Subjunctive* on **ar** verbs. Of course, ParaMor is unaware of morphosyntactic features, and so an appeal to syntactic features is

irrelevant to ParaMor's current algorithms. But clearly, ParaMor must carefully consider the 12th and 30th selected schemes (Figure 4.2), which ultimately model the **ar** and **er**/**ir** classes respectively, but which contain the c-suffixes **e** and **en** in common.

Syncretic paradigm overlap is widespread in languages beyond Spanish. In English, many nouns form their plural with the suffix **s**, but verbs can also take an **s** suffix, marking *3rd Person Present Tense*. Important for the evaluation of the work in this thesis, paradigm overlap also occurs in German, Finnish, Turkish, and Arabic which, together with English, comprise the evaluation languages of the Morpho Challenge competitions (see Chapter 6). When modeling the inflectional paradigms of English, Spanish, German, Finnish, Turkish, Arabic or any other language, ParaMor must retain distinct models of each paradigm, though some of their suffixes might be string identical.

Suffix string identity overlap between paradigms has direct implications for the scheme similarity measure ParaMor employs during clustering. Bottom-up agglomerative clustering, like most unsupervised clustering algorithms, decides which items belong in the same cluster by measuring similarity between and among the items being clustered. To cluster schemes, ParaMor must define a similarity between pairs of scheme-clusters. Perhaps the most intuitive measure of scheme similarity would compare schemes' c-suffix sets. But because a suffix surface form may appear in more than one paradigm, comparing c-suffix sets alone could spuriously suggest that schemes which model distinct but syncretic paradigms be merged.

Two facts about paradigm structure can rescue c-suffix-based scheme similarity measures from the complication of paradigm overlap. First, as a robust rule of thumb, while two paradigms may share the surface forms of one or more suffix, each paradigm will also contain suffixes the other does not. For example, although the **ar**, **er**, and **ir** inflection classes of the Spanish verbal paradigm have suffixes in common, each contains suffixes which the others do not—the infinitive suffixes **ar**, **er**, and **ir** themselves uniquely distinguish each inflection class. Similarly, in English, although verbal and nominal paradigms share the string **s** as a suffix, the verbal paradigm contains the suffixes **ing** and **ed** which the nominal paradigm does not; and the nominal paradigm contains suffix surface forms which the verbal paradigm does not: the English possessive

ending, which appears after the optional ***Number*** suffix, yields the orthographic paradigm cross-product forms: **'s**, **s's**, and **s'**, which are unique to nouns.

The second useful fact about paradigm structure that can aid a clustering algorithm is that a single lexical item can only belong to a single paradigm. In Spanish, any particular verb will belong to exactly one of the three inflection classes: **ar**, **er**, or **ir**. In the Spanish newswire corpus of 50,000 types that Figure 4.2 is built from, for example, no c-stem forms a word type by attaching the c-suffix **adas** while also forming a separate word by attaching **idas**: The c-suffixes **adas** and **idas** mark the *Past Participle Feminine Plural* in the **ar** and in the **ir**/**er** inflection classes respectively. Since **adas** and **idas** share no c-stem, it is reasonable to propose that **adas** and **idas** belong to distinct paradigms and that the 12th selected scheme, which contains **adas**, and the 30th selected scheme, which contains **idas** (see Figure 4.2), should not be merged

These two facts about paradigm structure lead ParaMor to permit no scheme-cluster to be formed which would contain a pair of c-suffixes which share no c-stems. The general technique of preventing induced clusters from containing items that are significantly dissimilar is known as discriminative clustering. ParaMor easily calculates the set of c-stems that two c-suffixes, f_1 and f_2, have in common by revisiting the morphology scheme network used in the initial search for candidate schemes, see Chapter 3. The level 2 network scheme subsuming f_1 and f_2 exactly holds the c-stems which form words with both f_1 and f_2. Typically, the $f_1 \cdot f_2$ scheme will be empty of c-stems exactly when, without loss of generality, f_1 is a c-suffix that is unique to a paradigm to which f_2 does not belong. Note that since the initial search will only select schemes containing a positive number of c-stems, if f_1 and f_2 are not mutually substitutable on at least one c-stem, then no single selected scheme can contain both f_1 and f_2.

ParaMor's discriminative clustering requirement that every pair of c-suffixes in a cluster share some c-stem is not foolproof. First of all, the general rule that distinct paradigms contain some distinct surface suffix is not a language universal. Even in Spanish, the two suffixes in the paradigm of ***Number*** on *adjectives* are identical to the suffixes of the largest inflection class of the ***Number*** paradigm on *nouns*—both paradigms contain

exactly the two suffixes **Ø** and **s**. And, indeed, both adjectives and nouns appear in ParaMor's final cluster which contains the **Ø.s** scheme.

At the same time, it is also possible for two lexical items that participate in two different paradigms to have surface-identical stems. In English, many verbal lexical items have a nominal counterpart: **to run** vs. **a run**, **a table** vs. **to table**, etc. And in Spanish, where stems do not readily convert part-of-speech without a change in the form of the stem, there are still lexical collisions. Among the most egregious collisions are pairs of unrelated Spanish verbs whose stems are surface identical but which belong to distinct inflectional paradigms: **parar** 'to stop' vs. **parir** 'to give birth,' **crear** 'to create' vs. **creer** 'to believe.' Thus, pairs of c-suffixes that uniquely distinguish paradigms, such as the infinitive endings **ar** and **ir**, or **ar** and **er**, can still share c-stems.

In practice, pairs of stem-identical lexical items are rare enough, and paradigm unique suffixes common enough, that with the discriminative restriction on clustering in place ParaMor is able to identify paradigms. Nevertheless, when a paradigm has no distinguishing suffix, or when two distinct paradigms contain many lexemes with surface-identical stems, ParaMor struggles to prevent distinct paradigms from merging—this is a clear area for future research.

The Second Challenge: Halting ParaMor's Clustering

In addition to helping keep distinct paradigms separate, ParaMor's discriminative restriction on the c-suffixes which can belong to a cluster also provides a principled halting criterion that avoids the introduction of an arbitrary similarity cutoff parameter. ParaMor allows agglomerative clustering to proceed until merging any pair of clusters would place into the same cluster two c-suffixes that share no c-stem. Thus, discriminatively restricting clustering by c-suffixes solves two of the three challenges facing a scheme-clustering algorithm.

The Third Challenge: Isolating Distinct Morpheme Boundaries

Now ParaMor must solve the third challenge: To not coalesce schemes that model competing morpheme boundary hypotheses. Consider the 1st, 2nd, 3rd, 5th, and 12th se-

Rank	Model of					Deriv.	Good	Complete	Partial	Error			C-Suffixes		C-Stems	
	N	Adj	Verb							Stem	Suffix	Chance				
			ar	er	ir											
1	●	●					●	●					2	Ø s	5501	apoyada barata hombro …
2		●					●	●					4	a as o os	892	apoyad captad dirigid junt …
3			●						●		●		15	Ø ba ban da das do dos ...	17	apoya disputa lanza lleva …
5			●				●		●				15	a aba aban ada adas ado ...	25	apoy desarroll disput lanz …
12			●				●		●				15	a aba ada adas ado ados ...	21	apoy declar enfrent llev …

Figure 4.3: An excerpt from Figure 4.1. Five schemes licensed by the word **apoyadas**.

lected schemes, as first presented in Figure 4.1 and repeated in Figure 4.3 for convenience. Each of these five schemes is licensed by the Spanish word **apoyadas** '*supported (Adjectival Feminine Plural)'*: The 1st selected scheme contains the c-stem **apoyada** and the c-suffix **s**, the 2nd scheme has the c-stem **apoyad** and the c-suffix **as**, the 3rd contains **apoya** and **das**, while the 5th and 12th selected schemes each contain the c-stem **apoy** and the c-suffix **adas**. Between them, these five schemes model, correctly or incorrectly, four distinct morpheme boundaries in the same word: **apoyada+s**, **apoyad+as**, **apoya+das**, and **apoy+adas**. As the 5th and 12th selected schemes hypothesize the same boundary, **apoy+adas**, it is reasonable to consider placing this pair of schemes in a single cluster. The other schemes licensed by **apoyadas** each hypothesize distinct morpheme boundaries and should remain in separate clusters.

Now consider the implications for a scheme similarity metric that arise from competition between schemes over their morpheme boundary hypotheses. Since schemes possess syntagmatic information in their sets of adherent c-stems, it seems reasonable to add information from c-stems to information from c-suffixes when evaluating scheme similarity. It also seems reasonable to give c-stem similarity and c-suffix similarity approximately equal weight. Fair representation of c-stems and c-suffixes is of concern as the number of c-stems can far outstrip the number of c-suffixes in a scheme: the 1st scheme selected in the ParaMor run of Figure 4.3 contains just two c-suffixes but more than five thousand c-stems.

To fairly weight syntagmatic and paradigmatic information in a scheme similarity metric, ParaMor compares schemes based on the sets of word types which license each scheme. Comparing schemes by their sets of licensing types weights the aggregate evidence from c-stems equally with the aggregate evidence from c-suffixes because each licensing type consists of one c-stem and one c-suffix. But, because a single word, such as **apoyadas**, may license two or more schemes which hypothesize distinct morpheme boundaries, directly measuring scheme similarity by comparing sets of licensing word types could erroneously merge schemes such as **Ø.s** and **a.as.o.os** which model distinct morpheme boundaries in many of the same words!

To address the problem of separating schemes which model distinct morpheme boundaries, ParaMor annotates each licensing type of each scheme with the morpheme boundary proposed by that scheme. For example, ParaMor annotates the licensing type **apoyadas** as **apoyada+s** for the **Ø.s** scheme, but as **apoyad+as** for the **a.as.o.os** scheme. Then, to compute the similarity between any pair of schemes, ParaMor measures the similarity between the pair's sets of morpheme boundary annotated types. Morpheme boundary annotated types retain the naturalness of measuring scheme similarity through sets of licensing word types, while distinguishing between schemes that model distinct morpheme boundaries,

To measure the similarity between pairs of scheme-clusters, as opposed to pairs of lone schemes, ParaMor must specify for each cluster the set of morpheme boundary annotated word types the cluster covers. There are at least two reasonable ways to define the set of boundary annotated types covered by a cluster. First, ParaMor could define this set as the union of the sets of morpheme boundary annotated types which belong to any individual scheme in the cluster. Alternatively, ParaMor could define a cluster's set of morpheme boundary annotated types as the cross-product of all c-stems and all c-suffixes contained in any scheme in the cluster. ParaMor opts for the more conservative first option. By allotting to each cluster just those morpheme boundary annotated types which explicitly license some selected scheme, ParaMor guarantees that reconcatenating the c-stem and c-suffix portions of any boundary annotated type in any cluster will produce a word that directly occurred in the paradigm induction corpus.

As long as scheme-cluster similarity is measured over sets of morpheme boundary annotated licensing types, ParaMor's performance at clustering schemes is not significantly affected by the particular set similarity metric used. For the experiments reported in this thesis, ParaMor relies on the cosine metric of set similarity. The formula for the cosine similarity of two arbitrary sets X and Y is: $|X \cap Y| / \left(|X||Y|\right)^{1/2}$.

4.3.2 Clustering Large and Clustering Small

To finish out the description of ParaMor's clustering algorithm, this section presents an adaptation that ParaMor makes to the basic agglomerative clustering algorithm that prevents schemes which are all individually licensed by very few types from joining forces to form a single larger cluster. The adaptation described in this section has a relatively mild impact on the final set of scheme-clusters that ParaMor induces. However, this section's adaptation is described for completeness, as all experiments reported in this thesis take this step.

Because individual schemes that receive support from few licensing types may still introduce valid c-suffixes, ParaMor does not want to simply discard all small selected schemes. Instead, ParaMor leverages larger selected schemes to rope in the valid small schemes. Specifically, ParaMor requires at least one large scheme for each small scheme a cluster contains, where the size of a scheme is measured as the number of unique morpheme boundary annotated word forms that license it. The threshold size above which schemes are considered large is a free parameter. As described in Section 4.4, the scheme size threshold is reused during ParaMor's filtering stage. Section 4.4.1 details the setting of this scheme size threshold.

4.3.3 An Examination of ParaMor's Scheme-Clusters

By tailoring bottom-up agglomerative clustering of schemes with first, a discriminative restriction over the c-suffixes a cluster may contain, and second, a similarity measure

over sets of morpheme boundary annotated types, ParaMor solves the three challenges that face a scheme-clustering algorithm. ParaMor:

1. Recognizes syncretic c-suffix overlap in paradigms,
2. Halts the agglomerative clustering algorithm without introducing a free parameter, and
3. Preserves the differing morpheme boundary hypotheses proposed by competing schemes.

As a technical summary of ParaMor's scheme-clustering algorithm, Appendix B contains a pseudo-code implementation of bottom-up agglomerative clustering adapted for schemes as described in Sections 4.3.1 and 4.3.2. This section focuses instead on illustrating the output of ParaMor's agglomerative clustering algorithm by looking at clusters built over Spanish selected schemes.

But first a note on the runtime complexity of ParaMor's clustering algorithm: In the worst case, standard bottom-up agglomerative clustering runs in $O(n^3)$ time, where n is the number of items to be clustered. In practice, the discriminative and scheme-size restrictions ParaMor places on clustering significantly speed up computation. The current Java implementation of ParaMor's scheme-clustering algorithm completes operation within ten minutes when clustering schemes induced from a corpus of 50,000 unique word types.

Figure 4.4 contains typical scheme-clusters that ParaMor builds after the three pipelined steps of:

1. Data clean-up (Section 4.2),
2. Initial scheme selection from a morphology network (Chapter 3), and
3. Scheme clustering (Sections 4.3.1 and 4.3.2).

Like Figure 4.1 found on p. 100, Figure 4.4 was built from a Spanish newswire corpus of 50,000 types, but all word types in the corpus from which the clusters in Figure 4.4 were built are longer than five characters. Because the corpus underlying Figure 4.1 is not

Rank	Corresponds to Figure 4.1	Licensing Types	# of Schemes	Model of N	Model of Adj	Model of Verb ar	Model of Verb er ir	Deriv. Allo.	Good	Err. Stem	Err. Suffix	C-Suffixes	C-Stems
1	1	11055	3	●	●			●	●			**4** Ø menente mente s	**5401** apoyada barata hombro inevitable segura …
2	2	4740	23			●		●			●	**37** Ø ba ban cion ciones ción da das do dor dora doras dores dos miento mos n ndo r ra ran remos rla rlas rle rles rlo rlos rme rnos ron rse rá rán ré ría rían	**1805** apoya compra disputa genera lanza lleva reclama senti utiliza visite …
3	3, 11	4142	4		●			●	●			**11** a amente as illa illas o or ora oras ores os	**976** apoyad captad frank guerr negociad …
4	5 12 (400)	1909	23			●		●	●			**41** a aba aban acion aciones ación ada adas ado ador adora adoras adores ados amos an ando ante antes ar ara aran aremos arla arlas arlo arlos arme aron arse ará arán aré aría arían ase e en ándose é ó	**113** apoy celebr desarroll genera hall lanz llev public realiz termin utiliz visit …
5	-	1095	16					●	●			**22** ion iona ionaba ionada ionadas ionado ionados ionamiento ionamientos ionan ionando ionantes ionar ionario ionaron ionará ionarán ione ionen iones ionó ión	**418** administrac condic cuest emoc gest les ocas pres reacc sanc ses vinculac …
10	10	722	4		●			●			●	**7** ta tamente tas titud to tos tísimo	**324** cier direc gra insóli len modes sangrien ...
11	30 127 (135)	720	17			●		●	●			**29** e edor edora edoras edores en er erlo erlos erse erá erán ería erían ida idas ido idos iendo iera ieran ieron imiento imientos iéndose ió í ía ían	**62** abastec aparec crec cumpl desconoc es-cond ofrec ocurr permit pertenec recib reun sal suspend transmit vend …
17	1592	384	11				●	●	●			**20** ida idas ido idor idores idos imos ir iremos irle irlo irlos irse irá irán iré iría irían ía ían	**39** abat conclu cumpl distribu eleg exclu permit persegu recib reun segu transmit …
21	1000	344	18			●		●				**29** ce cedores cemos cen cer cerlo cerlos cerse cerá cerán cería cida cidas cido cidos ciendo ciera cieran cieron cimiento cimientos cimos ció cí cía cían zca zcan zco	**26** abaste apare agrade compare conven est-able fortale ofre pertene recono ven …
100	-	82	2	●						●		**4** eta etas eto etos	**33** atl bol concr libr metrall papel secr tarj …
122	(40)	58	2	●								**6** Ø es idad idades mente ísima	**15** casual fort important irregular primer …
200	-	35	1							●		**5** staba stado star staron stará	**7** co conte entrevi gu in manife pre
300	-	30	1									**5** pondrán pone ponen poner puesta	**6** com dis ex im pro su
1000	-	15	1							●		**3** dismo dista distas	**5** bu ovie parti perio perre
2000	-	12	1							●		**3** ral rales ralismo	**4** electo libe neolibe plu
3000	-	8	1							●		**2** unas unos	**4** alg fa oport vac
5000	-	6	1							●		**2** ntra ntrarse	**3** ade ce conce
…		…										**…** …	**…** …

Figure 4.4: Typical clusters produced by ParaMor over a Spanish corpus of 50,000 types each longer than 5 characters.

identical to the corpus backing Figure 4.4, the schemes from which the clusters of Figure 4.4 were built are not identical to the schemes in Figure 4.1. But most schemes from Figure 4.1 have a close counterpart among the schemes which contribute to the clusters of Figure 4.4. For example, Figure 4.1 contains a **Ø.s** scheme, modeling the most frequent inflection class of ***Number*** on Spanish nouns and adjectives. A **Ø.s** scheme also contributes to the first cluster given in Figure 4.4, but where the **Ø.s** scheme of Figure 4.1 contains 5501 c-stems, the **Ø.s** scheme contributing to the 1st cluster of Figure 4.4 has a c-stem count of 5399. Note that only full clusters are shown in Figure 4.4, not the **Ø.s** scheme, or any other scheme, in isolation. As another example of similarity between the schemes of Figure 4.4 and those of Figure 4.1, turn to the third cluster of Figure 4.4. This third cluster contains a scheme model of ***Gender*** and ***Number*** on Spanish adjectives that consists of the same c-suffixes as the 2nd selected scheme in Figure 4.1, namely **a.as.o.os**.

Further correspondences between the clusters of Figure 4.4 and the schemes of Figure 4.1 are given in the second column of Figure 4.4, labeled **CORRESPONDS TO FIGURE 4.1**. If the cluster of a row of Figure 4.4 contains a scheme whose set of c-suffixes is identical, or nearly identical, to that of a scheme in Figure 4.1, then the rank of the corresponding scheme of Figure 4.1 is given outright in the **CORRESPONDS** column of Figure 4.4; if the majority of the c-suffixes of a scheme of Figure 4.1 appear in a cluster of Figure 4.4, but no particular scheme in that cluster exactly corresponds to the scheme of Figure 4.1, then the **CORRESPONDS** column of Figure 4.4 gives the rank of the Figure 4.1 scheme in parentheses.

The clusters in Figure 4.4 are sorted by the number of unique morpheme boundary annotated surface types which license schemes of the cluster—this number of unique licensing types appears in the third column of Figure 4.4. Because most c-stems do not occur in all of a cluster's schemes, the number of unique licensing types of a cluster is not simply the number of c-suffixes multiplied by the number of c-stems in the cluster. The fourth column of Figure 4.4 gives the number of schemes which merged to form that row's cluster. The only other column of Figure 4.4 which does not also appear in Figure 4.1 is the column labeled **ALLO**. The **ALLO**. column is marked with a dot when a row's

cluster models an allomorphic alternation. Clusters marked in the **ALLO**. column are discussed further in Section 4.4.2. For additional explanation of the other columns of Figure 4.4, reference their description in Section 4.1.1.

Zooming in close on one scheme-cluster, Figure 4.5 contains a portion of the clustering tree for the scheme-cluster from Figure 4.4 with the 4th most licensing types—a cluster covering suffixes which attach to **ar** verbs. The cluster tree in Figure 4.5 is of course binary, as it was formed through bottom-up agglomerative clustering. Schemes in Figure 4.5 appear in solid boxes, while intermediate clusters consisting of more than one scheme are in dotted boxes. Each scheme or cluster lists the full set of c-suffixes it contains, where a cluster contains all c-suffixes that appear in any scheme it subsumes. Leaf schemes also report their full sets of c-stems; and clusters state the cosine similarity between the sets of boundary annotated licensing types of the cluster's two children. It is interesting to note that similarity scores do not monotonically decrease moving up the tree structure of a particular cluster. Non-decreasing similarities are a consequence of computing similarities over sets of objects, in this case sets of morpheme boundary annotated types, which are *unioned* up the tree.

The bottom-most cluster of Figure 4.5, which covers 343 types, is built directly from two schemes. Two additional schemes then merge in turn with the bottom-most cluster. Finally, the top-most cluster of Figure 4.5 is built from the merger of two clusters which already have internal structure. The full cluster tree continues upward beyond the small branch shown in Figure 4.5 until the cluster contains 23 schemes. Although ParaMor can form clusters from children which do not both introduce novel c-suffixes, each child of each cluster in Figure 4.5 brings to its parent some c-suffix not found in the parent's other child. In each intermediate cluster of Figure 4.5, any c-suffix which does not occur in both children is underlined.

Keeping in mind Figures 4.4 and 4.5, examine ParaMor's scheme-clusters in light of the two broad shortcomings of the initially selected schemes, as discussed in Section 4.1.1, namely:

1. The fragmentation of language paradigms across many scheme models, and
2. The poor precision of selected models against underlying paradigms.

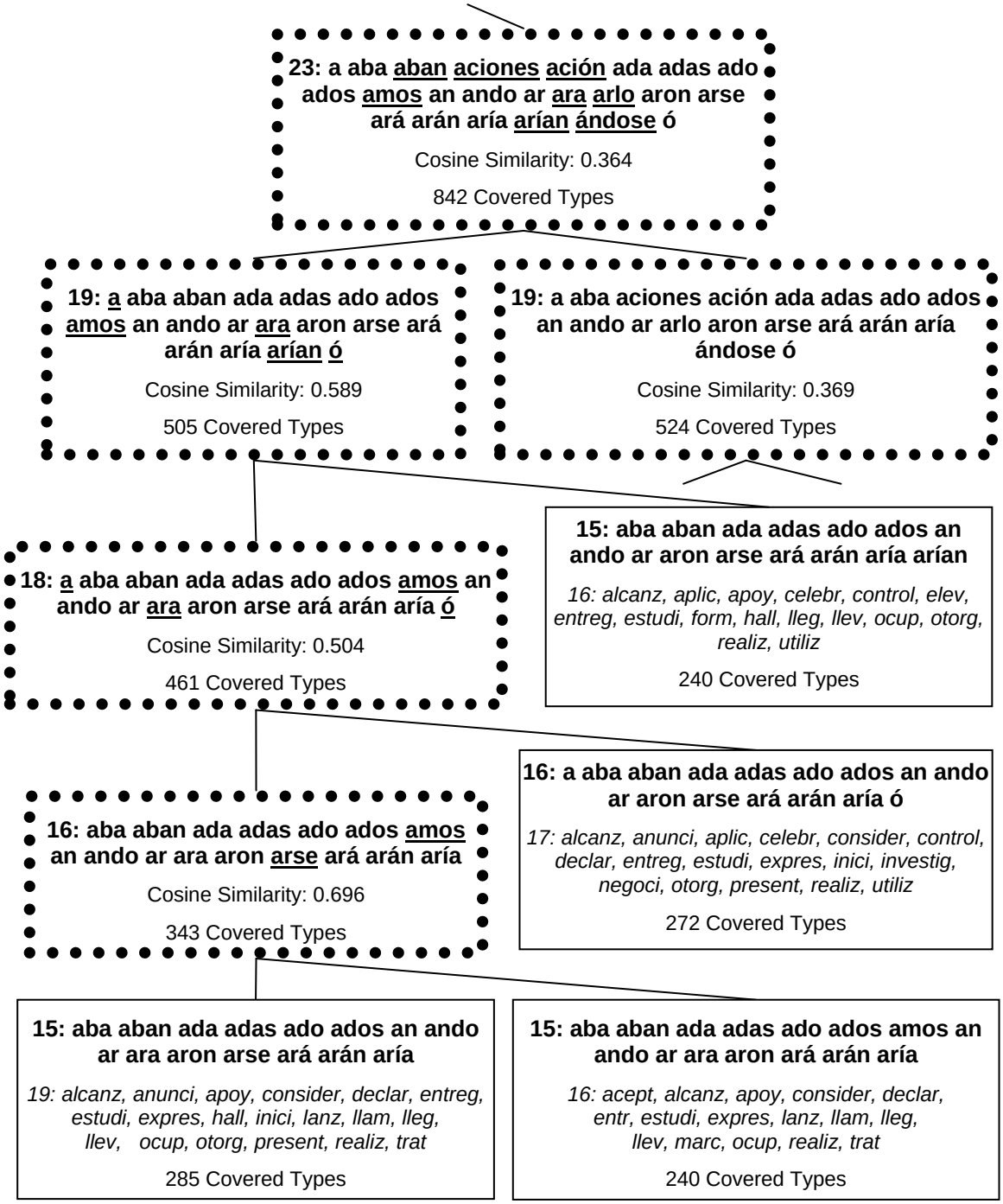

Figure 4.5: A portion of a cluster tree ParaMor built from schemes selected during a search run over 50,000 Spanish types longer than 5 characters. This cluster tree was constructed using ParaMor's bottom-up agglomerative clustering algorithm adapted for schemes as described in Sections 4.3.1 and 4.3.2. Each scheme appears in a solid box; each intermediate cluster in a dotted box. The c-suffixes of each scheme or cluster of schemes are in **bold**, c-stems are in *italics*. Each c-suffix in a cluster which uniquely originates in one child is underlined.

Clustering Improves Fragmentation

ParaMor's scheme clustering was specifically designed to address the patchwork fragmentation of paradigms across schemes. One of the most striking features of Figure 4.4 are the clusters which merge schemes that jointly and correctly model significant fractions of a single large Spanish paradigm.

One such significant joint model is the cluster with the 4th largest number of licensing types. A portion of this 4th largest cluster appears in the cluster-tree of Figure 4.5. All told, the 4th cluster contains 41 c-suffixes, more than any other scheme-cluster. These 41 c-suffixes model Spanish suffixes which attach to **ar** verb stems: 7 c-suffixes model agglutinative sequences of a non-finite inflectional suffix followed by a pronominal clitic, namely: **arla**, **arlas**, **arlo**, **arlos**, **arme**, **arse**, and **ándose**; 9 of the c-suffixes are various surface forms of the relatively productive derivational suffixes **ación**, **ador**, and **ante**; And more than half of the c-suffixes in this cluster are inflectional suffixes in the **ar** inflection class proper: This 4th cluster contains 24 c-suffixes which model inflectional **ar** suffixes, as presented in Appendix A; while one additional c-suffix, **ase**, is a less common alternate form of the *3rd Person Singular Past Subjunctive*. Counting just the 24 c-suffixes, the 4th scheme-cluster contains 64.9% of the 37 unique surface forms found among the suffixes of the **ar** inflection class of Spanish verbs that is listed in Appendix A. Among the 24 inflectional suffixes are all of the *3rd Person* endings for both *Singular* and *Plural* ***Number*** for all seven ***Tense-Mood*** combinations that are marked in Spanish. These *3rd Person* endings are: **a**, **an**, **ó**, **aron**, **aba**, **aban**, **ará**, **arán**, **aría**, **arían**, **e**, **en**, **ara**, and **aran**. Since newswire is largely written in *3rd Person*, it is to be expected that the *3rd Person* morphology is most readily identified from a newswire corpus.

Focusing in on one suffix of the **ar** verbal paradigm, **ados**, an example suffix followed throughout this chapter, clustering reduces the fragmentation of this one suffix across partial paradigms. Before clustering, the c-suffix **ados** occurred in 40 schemes, after clustering it is present in just 13 distinct clusters. Clearly, ParaMor's clustering algorithm has considerably consolidated the fragmented models of the Spanish **ar** verbal paradigm that were output by the initial bottom-up scheme search.

The **ar** inflection class is the most frequent of the three regular Spanish verbal inflection classes, and so is most completely identified by ParaMor. But the clusters with the 11th and 17th most licensing types cover, respectively, the **er** and **ir** inflection classes nearly as completely as the 4th cluster covers the **ar** paradigm: The 11th cluster models 19 of the 37 unique inflectional suffixes in the **er** inflection class, 4 inflection+clitic sequences, and 6 derivational suffixes; And the 17th cluster contains 14 of the 36 unique surface forms of inflectional suffixes in the **ir** inflection class, 4 inflection+clitic sequences, and 2 derivational suffixes.

Cluster Precision

Now consider how the clusters of Figure 4.4 stack up against the second broad shortcoming of ParaMor's initially selected schemes: the many original schemes that did not model paradigms. First, the data clean-up step, described in Section 4.2, which excludes short types from ParaMor's paradigm-induction corpus, virtually eliminated the first subclass of unsatisfactory schemes, namely those schemes which resulted from chance string similarities between word types. None of the scheme-clusters in Figure 4.4, for example, are built from schemes that arise from chance lexical collisions.

Although ParaMor now avoids constructing non-paradigmatic schemes that result from accidental string similarities, because ParaMor's bottom-up scheme search begins an upward path from each individual c-suffix, frequent or infrequent (Section 3.2.2), the initial search algorithm constructs many schemes that are licensed by very few word types. Some of these small schemes are absorbed into larger clusters, but ParaMor's c-suffix discriminative restriction on scheme clustering (Section 4.3.1), in combination with ParaMor's heuristic restriction on the number of small schemes which a cluster may contain (Section 4.3.2), prevents the majority of these small schemes from joining any cluster. From the 6909 original schemes that ParaMor selects when training on a corpus of types longer than 5 characters, clustering only reduces the total number of separate paradigm models to 6087 scheme-clusters.

The last six rows of Figure 4.4 all contain 'clusters' consisting of just a single scheme that was prevented from merging with any other scheme. All six singleton clusters are

licensed by no more than 35 word types. And none of the clusters correctly models inflectional affixes of Spanish. Five of the six singleton clusters misanalyze the morpheme boundaries in their few types, while the cluster with the 300th most licensing types treats a set of unproductive verbal prefixes as c-stems, placing valid verb *stems* into the c-suffix set. Section 4.4.1, directly addresses ParaMor's strategy for removing the many small incorrect singleton clusters that the clustering procedure produces.

In addition to the large number of incorrect singleton clusters, many initially created clusters misanalyze morpheme boundaries. But these incorrect morpheme boundary models are expected: As described in Section 4.1.2, ParaMor intentionally postpones discarding schemes which hypothesize unlikely morpheme boundaries until after the schemes have been clustered, in the hope that clustering might aggregate schemes which all model the same incorrect boundaries. Half of the clusters in Figure 4.4 hypothesize inferior morpheme boundaries in their licensing types. The largest such cluster is the cluster with the 2nd most licensing types. Like the 2nd selected scheme of Figure 4.1 which it subsumes, the 2nd cluster places morpheme boundaries after the **a** vowel which begins most suffixes in the **ar** inflection class. And exactly as hoped, the 2nd cluster has nicely unified schemes which all hypothesize the same morpheme boundaries in a large set of types—only this time, the hypothesized boundaries happen to be incorrect. Section 4.4.2 describes steps of ParaMor's pipeline which specifically remove clusters which hypothesize incorrect morpheme boundaries.

A New Shortcoming: Overgeneralization

Before moving on to a discussion of the algorithms ParaMor employs to improve the c-suffix precision of scheme-clusters, note that clustering introduces a new shortcoming into ParaMor's models of paradigms: overgeneralization. Each scheme, *C*, is a computational model that the specific set of c-stems and c-suffixes of *C* are paradigmatically related. When ParaMor merges *C* to a second scheme, *C′*, the paradigmatic relationship of the c-stems and c-suffixes of *C* is generalized to include the c-stems and c-suffixes of *C′* as well. Sometimes a merger's generalization is well founded, and sometimes it is misplaced. When both *C* and *C′* model inflectional affixes of the same paradigm on syntacti-

cally similar stems, then the c-stems of *C* usually do combine to form valid words with the c-suffixes of *C′*. For example, the suffixes **iré** and **imos** are regular inflectional suffixes of the **ir** inflection class of Spanish verbs. Although the c-suffix **iré** never occurs in any selected scheme with the c-suffix **imos**, and although the Spanish word **cumplimos** '*we carry out*' never occurs in the Spanish corpus from which ParaMor learns paradigms, the cluster of Figure 4.4 with the 21st most licensing types places the c-suffixes **iré** and **imos** in the same cluster together with the c-stem **cumpl**—correctly predicting that **cumplimos** is a valid Spanish word form.

On the other hand, when a c-suffix, *f*, of some scheme, *C*, models a syntactically or idiosyncratically restricted suffix, it is unlikely that *f* forms valid words with all the c-stems of a merged cluster *C′*. Consider the 1st scheme-cluster of Figure 4.4 which joins the scheme **Ø.s** with the schemes **Ø.mente.s** and **menente.mente**. The c-suffixes **Ø** and **s** mark *Singular* and *Plural* ***Number***, respectively, on nouns and adjectives; The suffix (**a**)**mente** productively converts an adjective into an adverb, something like the suffix **ly** in English; and the string **menente** is simply a misspelling. Where the **Ø.s** scheme contains 5399 c-stems, the scheme **Ø.mente.s** contains 253, and the scheme **menente.mente** contains just 3 candidate stems: **inevitable**, **unáni**, and **única**. The 1st scheme-cluster of Figure 4.4 contains many Spanish c-stems that represent only nouns, including **hombro** '*shoulder*' listed in the **C-STEM** column of Figure 4.4. These nominal c-stems originate in the **Ø.s** scheme and do not form legitimate Spanish adverbs by attaching **mente**: ***hombromente** '**shoulderly*'. Furthermore, productively assuming that the c-suffix **menente** can attach to any candidate stem is wrong. Thus this 1st cluster has *overgeneralized* in merging these three schemes.

Overgeneralization is endemic to all clustering algorithms, not just unsupervised bottom-up agglomerative clustering of schemes. And in the particular case of scheme clustering, it would be difficult for any unsupervised method to reliably distinguish between infrequent inflectional affixes on the one hand and reasonably frequent derivational affixes, such as **mente**, on the other. Chapters 5 and 6 of this thesis describe applying ParaMor's induced scheme-clusters to a morphological analysis task: Specifically ParaMor segments word forms into constituent morphemes. But before ParaMor could be applied

to a generation task that would propose novel full form words, the problem of overgeneralization in scheme-clusters would need to be seriously addressed.

4.4 Filtering of Merged Clusters

With most valid schemes having found a safe haven in a cluster with other schemes that model the same paradigm, ParaMor focuses on improving precision by removing erroneous scheme-clusters. ParaMor applies two classes of filters to cull out unwanted clusters. These two filter classes address the two remaining types of cluster-precision error described in Section 4.3.3, p. 116. The first filter class, detailed in Section 4.4.1, targets the many scheme-clusters with support from only few licensing types. The second class of filter, presented in Section 4.4.2 identifies and removes remaining scheme-clusters which hypothesize incorrect morpheme boundaries.

4.4.1 Filtering of Small Scheme-clusters

ParaMor's first class of filtering algorithm consists of just a single procedure which straightforwardly removes large numbers of erroneous small clusters: the filter discards all clusters that are licensed by less than a threshold number of morpheme boundary annotated word types. To minimize the number of free parameters in ParaMor, the threshold below which this filter discards small clusters is tied to the clustering threshold described in Section 4.3.2, which restricts the number of small schemes that may join a cluster to be no more than the number of large schemes in the cluster. These two thresholds can be reasonably tied together for two reasons. First, both thresholds limit the influence of small erroneous schemes. Second, both thresholds measure the size of a cluster as its number of licensing morpheme-boundary annotated types.

Figure 4.6 graphs the number of clusters that ParaMor identifies over a Spanish corpus after first clustering schemes with a particular setting, k, of the cluster-size threshold

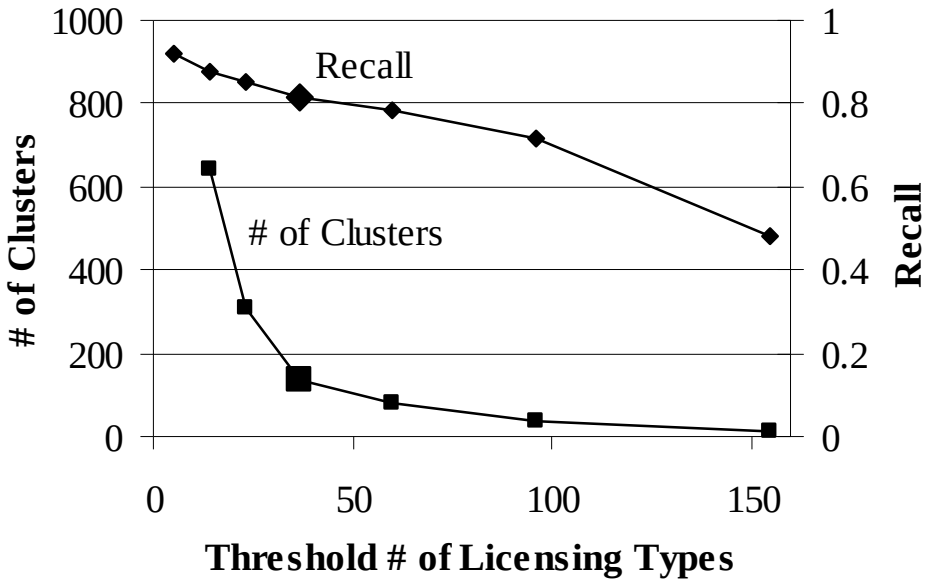

Figure 4.6: The number of clusters and their recall of unique Spanish suffixes as the scheme-cluster size threshold varies. The graph marker of each function at the threshold value of 37 unique licensing types is larger in size because it is this value of 37 which ParaMor uses in experiments reported in this thesis.

and then filtering out those clusters which are not licensed by at least *k* word types. Figure 4.6 also contains a plot of suffix recall as a function of these tied thresholds. ParaMor calculates suffix recall by counting the number of string-unique surface forms of Spanish inflectional suffixes, as given in Appendix A, that appear in any identified cluster. The technique described in this section for removing small clusters was developed before ParaMor adopted the practice of only training on longer word types; and Figure 4.6 presents the cluster count and suffix recall curves over a corpus that includes types of all lengths. Figure 4.6 presents results over corpora that are not restricted by type length because it is from the unrestricted corpus data that the threshold for small cluster filtering was set to the value which ParaMor uses in experiments throughout this thesis. As noted

below, over a Spanish corpus that only includes types longer than 5 characters, the effect of filtering by licensing type count is similar.

Looking at Figure 4.6, as the size threshold increases, the number of clusters that ParaMor retains quickly drops. But suffix recall only slowly falls during the steep decline in cluster count: ParaMor is therefore discarding mostly bogus schemes containing illicit suffixes. Because recall degrades gracefully, the exact size threshold below which clusters are discarded should have a relatively minor effect on the paradigms that ParaMor induces. At a threshold size of 37 morpheme-boundary annotated licensing word types, ParaMor retains more than 80% of the string-unique inflectional suffixes of Spanish, while significantly reducing the number of clusters that ParaMor selects.

At this threshold value of 37, all but 137 of the 7511 clusters that formed from the 8339 originally selected schemes are removed, a 98.2% reduction in the number of clusters. Note that the vertical scale on Figure 4.6 goes only to 1000 clusters. Counting in another way, before filtering, the scheme-clusters contained 9896 unique c-suffixes, and after filtering just 1399, an 85.9% reduction. The recall of unique inflectional suffixes at a threshold value of 37 licensing types is 81.6%, or 71 out of 87. Except where otherwise noted, all experiments reported in this thesis use a cluster-size threshold of 37.

Before filtering schemes for the number of licensing types they contain, 92.0% of the unique suffixes of Spanish morphology appeared as a c-suffix in some scheme-cluster. But this automatically derived value of 92.0%, or 80 of 87, is somewhat misleading. At a threshold value of 37, nine unique c-suffixes which are string identical to true Spanish suffixes are lost. But six of the nine lost unique c-suffixes appear in schemes that do not model Spanish inflectional suffixes. For example, one c-suffix that is removed during cluster size filtering is **erías**, which is string identical to a *2nd Person Singular Present Conditional* Spanish verbal suffix. But the c-suffix **erías** appears in only one cluster—a cluster which clearly does not model Spanish verbs. The cluster in which **erías** occurs consists of the single scheme **ería.erías.o.os** with its c-stems **escud.ganad.grad.libr.-mercad**. Although the c-suffixes **ería**, **erías**, and **o** are all string identical to suffixes in the Spanish **er** verbal paradigm, in *this* scheme these c-suffixes do not arise from verbal licensing forms: Reconcatenating c-stems and c-suffixes, most word forms which both

license this cluster and end in **ería** or **erías**, like **librería**/**librerías** '*library*/*libraries*' and **ganadería**/**ganaderías** '*ranch*/*ranches*', are not verbs but nouns; And forms with the **o**/**os** endings, like **libro/libros** '*book/books*' and **ganado/ganados** '*cattle* (sg/pl)', are derivationally related nouns. After ignoring c-suffixes which appear in schemes where the c-suffix does not model a Spanish inflectional suffix given in Appendix A, only three true c-suffixes are lost during this first filtering step that removes small clusters.

When training from a Spanish corpus that consists entirely of types longer than 5 characters in length, the numbers follow a similar pattern to those given for the unrestricted corpus: Before clustering, ParaMor identifies 6909 schemes; this is reduced to 6087 after clustering; and after filtering at a threshold of 37, only 150 clusters remain. The recall of unique suffixes, when training over a Spanish corpus restricted for type length, are identical to the values over the unrestricted corpus: 92.0% before filtering and 81.6% after, although, oddly, ParaMor does not either identify or filter exactly the same set of inflectional suffixes. Of the clusters presented in Figure 4.4, the six clusters in the last six rows, repeated here as Figure 4.7, each contains fewer than 37 licensing types, and each is therefore discarded.

Rank	4.1Error! Reference	Licensing Types	# of Schemes	Model of N	Model of Adj	Model of Verb ar er ir	Model of Deriv. Allo.	Good	Err. Stem	Err. Suffix	C-Suffixes	C-Stems
200	-	35	1						●		**5** staba stado star staron stará	**7** co conte entrevi gu in ...
300	-	30	1								**5** pondrán pone ponen poner puesta	**6** com dis ex im pro su
1000	-	15	1						●		**3** dismo dista distas	**5** bu ovie parti perio perre
2000	-	12	1						●		**3** ral rales ralismo	**4** electo libe neolibe plu
3000	-	8	1						●		**2** unas unos	**4** alg fa oport vac
5000	-	6	1						●		**2** ntra ntrarse	**3** ade ce conce

Figure 4.7: An excerpt from Figure 4.4: Six scheme-clusters that are removed by ParaMor's small-cluster filter. Each removed cluster is licensed by fewer than the threshold number of licensing types, 37. The third column gives the number of types that license each scheme-cluster.

And finally, when inducing paradigms over a corpus restricted by word-length, paradigm fragmentation improves for the particular suffix of the **ar** inflection class that this chapter follows: After filtering, 7 clusters remain out of the original 13 clusters which contained the c-suffix **ados**. Two of the six discarded clusters that contain **ados** are clearly correctly discarded as they are among the few surviving clusters which arise from chance string similarities between word types. Of the additional four clusters containing **ados** that ParaMor removes, each models at least one relatively unproductive derivational suffix. The discarded cluster **ado.ados.amento.amentos.ando**, for example, contains c-suffixes which model the inflectional suffixes **ado**, **ados**, and **ando**, but also c-suffixes modeling forms of the derivational suffix **amento/amentos**, which forms nouns from some **ar** verbs. As ParaMor is designed to identify productive inflectional morphology, it is not unreasonable or unexpected that scheme-clusters modeling unproductive derivational suffixes are lost.

4.4.2 Morpheme Boundary Filtering

In Spanish, as described in Section 4.4.1, filtering scheme-clusters by thresholding the number of types that must license each cluster drastically reduces the number of clusters that ParaMor proposes as models for inflectional paradigms. From the thousands of initially created scheme-clusters, type-license filtering leaves fewer than two hundred. This is progress in the right direction. But as Spanish has less than ten productive inflectional paradigms, see Appendix A, ParaMor still vastly over estimates the number of Spanish paradigms.

A hand analysis of scheme-clusters reveals that the major type of erroneous cluster which persists after size filtering are those clusters that incorrectly model morpheme boundaries. After training and filtering over a Spanish corpus of 50,000 types that are each longer than 5 characters in length, exactly 150 scheme-clusters remain. And of these 150 clusters, more than two-thirds, 108, hypothesize an incorrect morpheme boundary in their licensing types. That misplaced morpheme boundaries are the major source of error in the remaining scheme-clusters is not surprising. Morpheme boundary errors comprised

one of the major error classes of the initially selected schemes identified in Section 4.1.1. And thus far in ParaMor's processing pipeline no algorithm has specifically dealt with boundary errors. Indeed, as described in Section 4.1.1, ParaMor has intentionally postponed the removal of schemes and scheme-clusters that hypothesize incorrect morpheme boundaries until after ParaMor's scheme-clustering step.

The Problem: Mutual Substitutability

ParaMor's initial bottom-up morphology network search strategy, described in Chapter 3, is designed to detect the hallmark of inflectional paradigms in natural language: mutual substitutability between sets of affixes. However, when the c-suffixes of a scheme break not at a morpheme boundary, but rather at some character boundary internal to a true morpheme, the incorrect c-suffixes are sometimes still mutually substitutable. Figure 4.8 contains three scheme-clusters, built over a Spanish corpus, that illustrate non-paradigmatic mutual substitutability. The schemes of Figure 4.8 were first shown in Figure 4.4. The scheme-cluster on the 1st row of Figure 4.8 incorrectly hypothesizes a morpheme boundary that is after the **a** vowel which begins many inflectional and derivational **ar** verb suffixes. In placing the morpheme boundary after the **a**, this scheme-cluster cannot capture the full paradigm of **ar** verbs. Compare, for example, the cluster on the 2nd row of Figure 4.8, which includes inflectional suffixes such as **o** *1st Person Singular Present Indicative* and **é** *1st Person Singular Past Indicative* which do not begin with **a**. Nevertheless, the (incorrect) c-suffixes which appear in the 1st row's cluster are mutually substitutable: in the Spanish word **administrados** *'administered* (*Adjectival Masculine Plural*)*'*, the erroneously segmented c-suffix **dos** can be replaced by **Ø** to form the Spanish word **administra** *'administer* (*3rd Person Singular Present Indicative*)', or by the c-suffix **da** to form **administrada** '(*Adjectival Feminine Singular*)', etc.

Similarly, the scheme-cluster on the 3rd row of Figure 4.8 models some of the many Spanish adjective stems which end in **t**, **abier<u>t</u>o** *'open'*, **cier<u>t</u>o** *'certain'*, **pron<u>t</u>o** *'quick'* etc.—but this cluster incorrectly prepends the final **t** of these adjective stems to the adjectival suffixes, forming c-suffixes such as: **ta**, **tas**, and **to**. And these prepended c-suffixes are mutually substitutable on adjectives whose stems end in **t**: **abierto** becomes **abiertas**

RANK	CORRESPONDS TO FIGURE 4.1	LICENSING TYPES	# OF SCHEMES	MODEL OF N ADJ	MODEL OF VERB ar er ir	MODEL OF DERIV. ALLO.	GOOD	ERR. STEM	ERR. SUFFIX	C-SUFFIXES		C-STEMS
2	2	4740	23		•	•			•	**37**	Ø ba ban cion ciones ción da das do dor dora doras dores dos miento mos n ndo r ra ran remos rla rlas rle rles rlo rlos rme rnos ron rse rá rán ré ría rían	**1805** apoya compra disputa genera lanza lleva reclama senti utiliza visite …
4	5 12 (400)	1909	23		•	•	•			**41**	a aba aban acion aciones ación ada adas ado ador adora adoras adores ados amos an ando ante antes ar ara aran aremos arla arlas arlo arlos arme aron arse ará arán aré aría arían ase e en ándose é ó	**113** apoy celebr desarroll genera hall lanz llev public realiz termin utiliz visit …
10	10	722	4	•		•		•		**7**	ta tamente tas titud to tos tísimo	**324** cier direc gra insóli len modes sangrien ...

Figure 4.8: An excerpt from Figure 4.4. The rank 2 cluster hypothesizes a morpheme boundary internal to true suffixes; the rank 4 cluster correctly models morpheme boundaries that immediately follow verbal stems; while the rank 10 cluster models boundaries internal to stems.

when **to** is replaced by **tas**. Encountering schemes like those that contribute to the clusters on the 1st and 3rd rows of Figure 4.8, ParaMor's initial search strategy discovers mutual substitutability and erroneously selects these incorrectly segmented c-suffix sets. Since ParaMor cannot rely on mutual substitutability of suffixes to identify correct morpheme boundaries, ParaMor turns to a secondary characteristic of paradigms.

The Solution: Letter Successor Variety

The secondary characteristic that ParaMor adapts in order to identify and discard those scheme-clusters which hypothesize incorrect morpheme boundaries is *letter successor variety*. Letter successor variety is an idea that was originally proposed by Harris (1955). Take any string *t*. Let *F* be the set of strings such that for each $f \in F$, *t.f* is a word form of a particular natural language. Harris noted that when the right edge of *t* falls

at a morpheme boundary, the strings in *F* typically begin in a wide variety of characters; but when *t* divides a word form at a character boundary internal to a morpheme, any legitimate word final string must first complete the erroneously split morpheme, and so will begin with the same character. This argument similarly holds when the roles of *t* and *f* are reversed.

Harris harnesses this idea of letter successor variety by first placing a corpus vocabulary into a character tree, or trie, and then proposing morpheme boundaries after trie nodes that allow many different characters to immediately follow. Consider Harris' algorithm over a small English vocabulary consisting of the twelve word forms: **rest**, **rests**, **resting**, **retreat**, **retreats**, **retreating**, **retry**, **retries**, **retrying**, **roam**, **roams**, and **roaming**. The upper portion of Figure 4.9 places these twelve English words in a trie. The bottom branch of the trie begins **r**-**o**-**a**-**m**. Three branches follow the **m** in **roam**, one branch to each of the trie nodes **Ø**, **i**, and **s**. Harris suggests that such a high branching factor indicates there may be a morpheme boundary after **r**-**o**-**a**-**m**. The trie in Figure 4.9 is a forward trie in which all vocabulary items share a root node on the left. A vocabulary also defines a backward trie that begins with the final character of each vocabulary item.

Adapting Letter Successor Variety to Schemes

Interestingly there is a close correspondence between trie nodes and ParaMor schemes. Each circled sub-trie of the trie in the top portion of Figure 4.9 corresponds to one of the four schemes in the bottom-right portion of the figure. For example, the right-branching children of the **y** node in **retry** form a sub-trie consisting of **Ø** and **i**-**n**-**g**, but this same sub-trie is also found following the **t** node in **rest**, the **t** node in **retreat**, and the **m** node in **roam**. ParaMor conflates all these sub-tries into the single scheme **Ø.ing** with the four adherent c-stems **rest**, **retreat**, **retry**, and **roam**. Notice that the number of leaves in a sub-trie corresponds to the paradigmatic c-suffix level of a scheme, e.g. the level 3 scheme **Ø.ing.s** corresponds to a sub-trie with three leaves ending the trie paths **Ø**, **i**-**n**-**g**, and **s**. Similarly, the number of sub-tries which conflate to form a single scheme corresponds to the number of adherent c-stems belonging to the scheme.

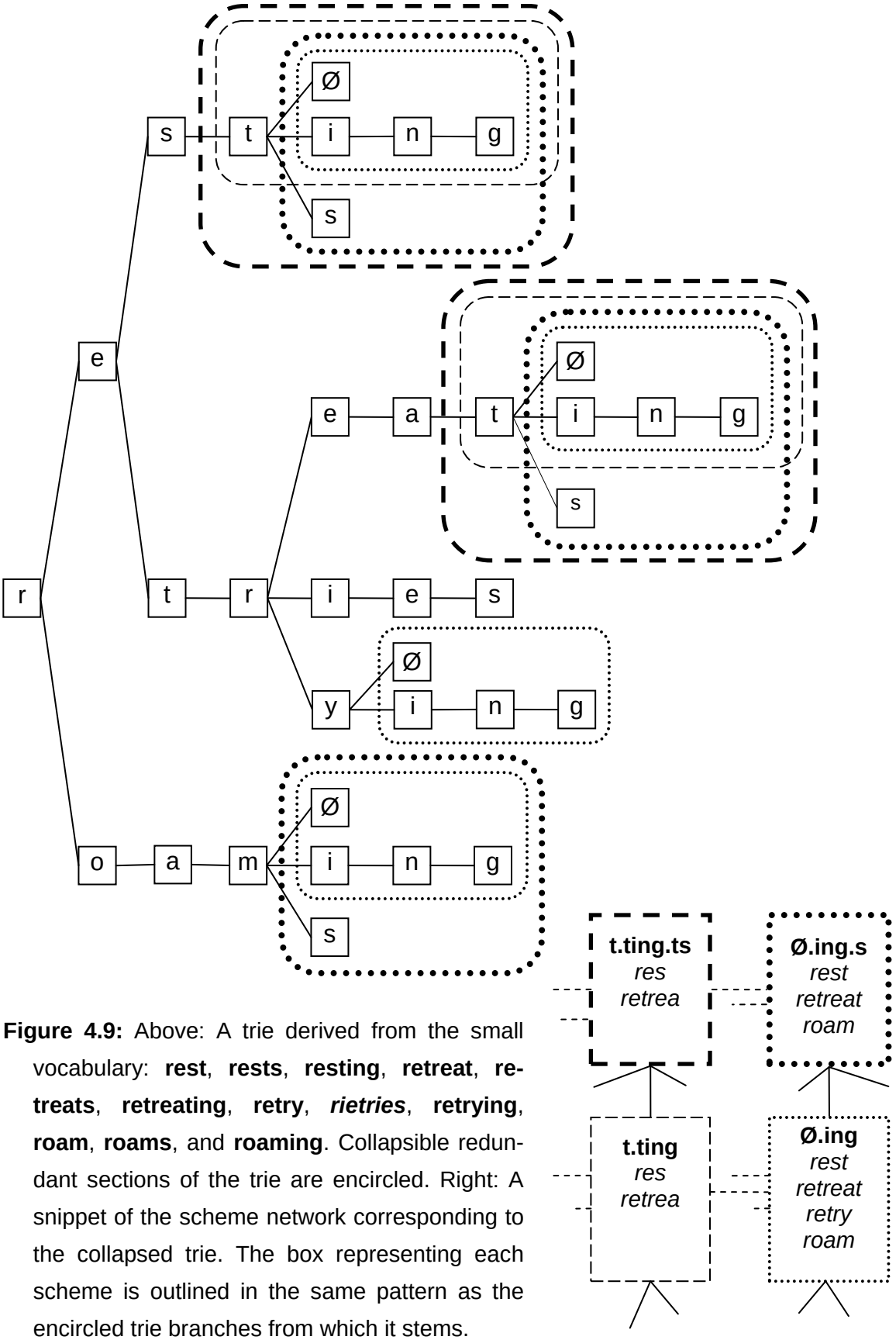

Figure 4.9: Above: A trie derived from the small vocabulary: **rest**, **rests**, **resting**, **retreat**, **retreats**, **retreating**, **retry**, ***rietries***, **retrying**, **roam**, **roams**, and **roaming**. Collapsible redundant sections of the trie are encircled. Right: A snippet of the scheme network corresponding to the collapsed trie. The box representing each scheme is outlined in the same pattern as the encircled trie branches from which it stems.

Just as schemes are analogues of trie nodes, ParaMor can link schemes in a fashion analogous to transition links between nodes in forward and backward tries. Transition links emanating to the right from a particular scheme, *C*, will be analogues of the transition links in a forward trie, and links to the left, analogues of transition links in a backward trie. To define forward scheme links from a scheme, *C*, let the set F_h consist of all c-suffixes of *C* which begin with the same character, *h*. Strip *h* from each c-suffix in F_h, forming a new set of c-suffixes, $F_{\bar{h}}$. Link *C* to the scheme containing exactly the set of c-suffixes $F_{\bar{h}}$. Schemes whose c-suffixes all begin with the same character, such as **t.ting.ts** and **t.ting**, have exactly one rightward path that links to the scheme where that leading character has been stripped from all c-suffixes. For example, in Figure 4.9 the **t.ting.ts** scheme is linked to the **Ø.ing.s** scheme.

Leftward links among schemes are defined by reversing the roles of c-stems and c-suffixes as follows: for each character, *h*, which ends a c-stem in a particular scheme, *C*, a separate link takes *C* to a new scheme where *h* starts all c-suffixes. For example, the **Ø.ing.s** scheme contains the c-stems **rest** and **retreat**, which both end in the character **t**, hence there is a link from the **Ø.ing.s** scheme to the **t.ting.ts** scheme. Note that when all the c-suffixes of a scheme, *C*, begin with the same character, the rightward link from *C* to some scheme, *C′*, exactly corresponds to a leftward link from *C′* to *C*.

Drawing on the correlation between character tries and scheme networks, ParaMor ports Harris' trie based morpheme boundary identification algorithm quite directly into the space of schemes and scheme-clusters. Just as Harris identifies morpheme boundaries by examining the variety of the branches emanating from a trie node, ParaMor identifies morpheme boundaries by examining the variety in the trie-style scheme links. ParaMor employs two filters which examine trie-style scheme links: the first filter seeks to identify scheme-clusters, like the cluster on the 1st row of Figure 4.8, whose morpheme boundary hypotheses are placed inside true suffixes; while the second filter removes scheme-clusters that hypothesize stem-internal morpheme boundaries, as the cluster on the 3rd row of Figure 4.8 does.

Filtering Suffix-Internal Morpheme Boundary Errors

To identify scheme-clusters whose morpheme boundary hypotheses are incorrectly suffix-internal, ParaMor's first boundary filter examines the Harris-style letter successor variety of the leftward trie links of the schemes in a cluster. When a scheme places its morpheme boundaries at locations that are internal to true suffixes, *left*ward trie links lead back toward the stem and back toward the correct morpheme boundary of each licensing type.

Following Hafer and Weiss (1974) in a trie algorithm and Goldsmith (2001; 2006) in a paradigm-based system, ParaMor measures letter successor variety using entropy. Each leftward scheme link, *l*, can be weighted by the number of c-stems in that scheme whose final character advocates *l*. In Figure 4.9 two c-stems in the **Ø.ing.s** scheme end in the character **t**, and thus the leftward link from **Ø.ing.s** to **t.ting.ts** receives a weight of two. ParaMor then measures the entropy of the distribution of the c-stem weighted links. A leftward link entropy close to zero indicates that the c-stems of a scheme have little variety in their final character; And minimal character variety is the sign of a boundary hypothesis placed at a morpheme-internal position.

To judge whether a cluster of schemes hypothesizes an incorrect morpheme boundary, ParaMor's suffix-internal boundary filter examines the leftward link entropy of each scheme in each cluster. Each scheme with a leftward link entropy below a threshold is flagged as an error. And if at least half of the schemes in a cluster are flagged, then ParaMor's suffix-internal error filter discards that cluster. ParaMor's suffix-internal filter is conservative in discarding schemes. ParaMor uses a threshold value, 0.5, over the leftward link entropy that only flags a scheme as containing a boundary error when virtually all of a scheme's c-stems end in the same character. **Figure 4.10** is a pseudo-code implementation of ParaMor's algorithm to filter out suffix-internal morpheme boundary errors.

Filtering Stem-Internal Morpheme Boundary Errors

ParaMor employs a second morpheme boundary filter to identify scheme-clusters which incorrectly hypothesize boundaries that are internal to the stems of the words

```
FilterSuffixInternalBoundaryErrors(schemeClusters) {
  foreach (schemeCluster in schemeClusters) {
    countOfErrorSchemes = 0;
    countOfNonErrorSchemes = 0;
    foreach (scheme in schemeCluster) {
      if (likelyModelOfAMorphemeLeftEdge(scheme))
        countOfNonErrorSchemes++;
      else
        countOfErrorSchemes++;
    }
    if (countOfErrorSchemes >= countOfNonErrorSchemes) {
      schemeClusters.remove(schemeCluster);
    }
  }
  return schemeClusters;
}

// Measure the entropy of a scheme's leftward trie-style links
likelyModelOfAMorphemeLeftEdge(scheme) {
  foreach (cStem in scheme.cStems) {
    stemFinalChars{cStem.finalChar()}++;
  }
  if (entropy(stemFinalChars) > threshold)
    return true;
  else
    return false;
}
```

Figure 4.10: A pseudo-code implementation of ParaMor's algorithm to filter out scheme-clusters that likely hypothesize morpheme boundaries for their licensing types that fall internal to true suffixes.

which license the cluster. This filter over stem-internal boundary errors is designed to remove schemes like that found on the 2nd row of Figure 4.11, where the final **t** of true Spanish stems has been stripped away and erroneously added to the front of all c-suffixes in the scheme.

But consider ParaMor's quandary when examining the morpheme boundaries proposed by the schemes on the 1st and 3rd rows of Figure 4.11. The schemes on the 1st and 2nd rows of Figure 4.11 are taken from Figure 4.1 on p. 100, which lists schemes selected in one particular Spanish run of ParaMor's initial scheme-search procedure. The scheme on the 3rd row of Figure 4.11 is a valid scheme which ParaMor could have selected, but

did not. The schemes on the 1st and 3rd rows of Figure 4.11 both attempt to model inflectional suffixes of the verbal **ar** paradigm. The potentially selected 3rd row scheme arises from the same word types which license the 1st row's scheme, but proposes different morpheme boundaries in this set of words. And indeed, it is somewhat ambiguous where to place the morpheme boundary in Spanish verbs. All of the c-stems of the 1st row's scheme end with the character **a**, but all of the c-suffixes of the 3rd row's scheme begin with the character **a**. While traditional grammarians would likely prefer the boundaries hypothesized by the scheme on the 3rd row, many linguists would argue that the inflectional suffix of a Spanish verb begins after the characteristic vowel, as the 1st row's scheme suggests. ParaMor adopts a pragmatic view, when there is ambiguity about a morpheme boundary, ParaMor retains the left-most reasonable boundary. In Figure 4.11, it is the scheme on the 3rd row that proposes the left-most boundary that is consistent with the word forms which license the 1st and the 3rd rows' schemes. So ParaMor should discard the 1st row scheme.

Biasing ParaMor's morpheme boundary filters toward preferring the left-most reasonable boundary requires that a filter designed to identify misplaced stem-internal boundaries cannot merely be the mirror image of the suffix-internal error filter. As described in the previous sub-section, the suffix-internal boundary error filter will note the

Rank	Model of					Error		C-Suffixes		C-Stems	
	N	Adj	Verb	Deriv.	Good Complete	Partial	Stem Suffix Chance				
3			ar er ir					15	Ø ba ban da das do dos n ndo r ra ron rse rá rán	17	apoya disputa lanza lleva toma …
10	•				•		•	5	ta tamente tas to tos	22	cier direc insóli modes …
n/a			•		•			15	a aba aban ada adas ado ados an ando ar ara aron arse ará arán	17	apoy disput lanz llev tom ...

Figure 4.11: The first two rows contain schemes first encountered in Figure 4.1. The final row is a valid scheme not selected by ParaMor.

low entropy of the c-stem-final character distribution among the c-stems of the scheme on the 1st row of Figure 4.11; consequently the suffix-internal error filter will flag the 1st row's scheme as hypothesizing an incorrect morpheme boundary. If the stem-internal error filter were simply the mirror image of the suffix-internal filter, then, because all of the c-suffixes of the 3rd row's scheme begin with the same character, the scheme on the 3rd row of Figure 4.11 would *also* be flagged as not modeling a correct morpheme boundary! ParaMor's stem-internal error filter achieves asymmetry from the suffix-internal filter by examining both rightward and leftward scheme links. Indeed, the core subroutine of the suffix-internal error filter, that measures the entropy of leftward scheme links, is called from within ParaMor's stem-internal filter.

The first requirement that a scheme, *C*, must meet to be flagged as a stem-internal boundary error is that all the c-suffixes in *C* must begin with the same character, or equivalently, that the entropy in the distribution of the initial characters of *C*'s c-suffixes must be zero. When all the c-suffixes of *C* begin with the same character, ParaMor strips that character from each c-suffix in *C* and examines the scheme, *C′*, that contains exactly the stripped set of c-suffixes. For example, since all the c-suffixes in the scheme on the 3rd row of Figure 4.11 begin with the character **a**: **a.aba.aban.ada.adas.ado.ados.an.-ando.ar.ara.aron.arse.ará.arán**, ParaMor's stem-internal boundary error filter follows the single rightward path along the character **a** to the scheme found on the 1st row of Figure 4.11, namely **Ø.ba.ban.da.das.do.dos.n.ndo.r.ra.ron.rse.rá.rán**. Similarly, because all of the c-suffixes that belong to the scheme on the 2nd row of Figure 4.11 begin with the character **t**: **ta.tamente.tas.to.tos**, ParaMor examines the scheme **a.amente.as.-o.os**.

Once ParaMor has identified a *C′* scheme that lies along an unambiguous rightward scheme link, ParaMor considers the likelihood that *C′* is the left edge of a morpheme boundary. If *C′* does fall at a likely morpheme boundary then ParaMor discards the original *C* scheme—ParaMor assumes that the original *C* misplaced the morpheme boundary a little bit to the left of the correct location, accidentally moving inside a true stem. To assess the likelihood that *C′* correctly models the left edge of a morpheme boundary, ParaMor simply asks the *suffix*-internal morpheme boundary error filter to measure the en-

tropy of the leftward distribution of c-stem-final characters of *C′*. For example, when *C* is the scheme on the 2nd row of Figure 4.11 and *C′* is the scheme **a.amente.as.o.os** (a scheme which correctly models morpheme boundaries that divide adjectival stems and suffixes) the suffix-internal boundary error filter finds that *C′* does indeed mark a morpheme boundary—and so **ta.tamente.tas.to.tos** is rightly flagged as erroneously hypothesizing a stem-internal morpheme boundary. Conversely, when *C* is the scheme on the 3rd row of Figure 4.11 and *C′* is the 1st row's scheme, the 3rd row's scheme will *not* be discarded, because the suffix-internal error filter will *not* flag the 1st row's scheme as lying at a morpheme boundary—the c-stems of the 1st row scheme end in a wide variety of characters.

The scheme on the 2nd row of Figure 4.11 has mis-hypothesized morpheme boundaries exactly one character inside the stems of its licensing types. But it is not uncommon for ParaMor to select schemes that clip two or more characters off the tail of each stem. To catch stem-internal boundary errors that are placed deep inside stems, ParaMor follows chains of rightward scheme links. Specifically, if ParaMor can reach a scheme along a non-branching rightward path that scores as a likely left edge of a morpheme boundary, then the original scheme is discarded.

Two examples will help illustrate rightward scheme chains in the stem-internal filter. First, as mentioned, when ParaMor's stem-internal error filter evaluates the scheme from the 3rd row of Figure 4.11, the filter first visits the scheme on the 1st row of Figure 4.11 and decides that the 1st row's scheme does not model a morpheme boundary. At this point, the stem-internal error filter attempts to strip off yet another character from the front of the c-suffixes that belong to the 1st row's scheme. But the c-suffixes of the scheme on the 1st row of Figure 4.11 do not all begin with the same character: the null c-suffix, **Ø**, begins with the null character, the c-suffix **ba** begins with **b**, the c-suffix **da** with **d**, etc. Since multiple rightward paths emanate from the 1st row's scheme, the stem-internal error filter cannot examine any more schemes. And, as the stem-internal filter encountered no rightward scheme with high leftward entropy, ParaMor accepts the scheme **a.aba.aban.ada.adas.ado.ados.an.ando.ar.ara.aron.arse.ará.arán** as modeling a valid morpheme boundary.

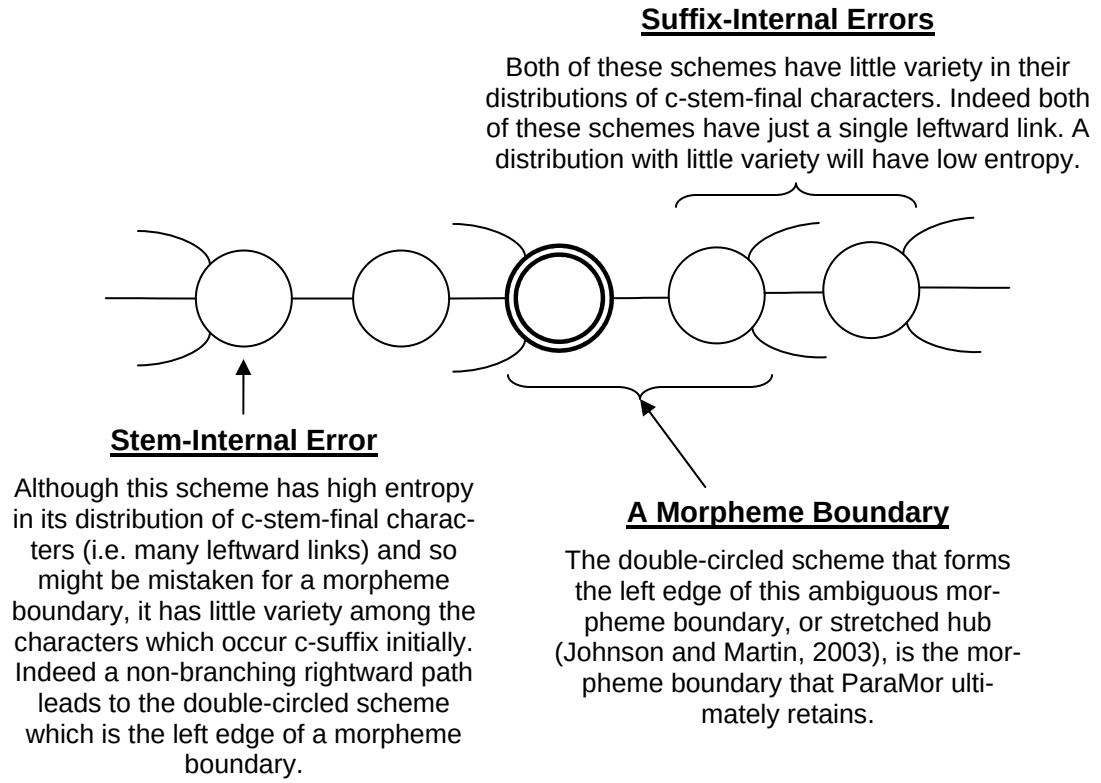

Figure 4.12: An illustration of ParaMor's morpheme boundary error filters. In this diagram, each circle represents a scheme. Schemes are connected along transition links that follow c-stem-final characters to the left and c-suffix-initial characters to the right. ParaMor examines schemes' transition-link distributions to decide which schemes most likely model true morpheme boundaries.

As a second example of rightward scheme chains, consider the diagram in Figure 4.12. Each circle in this diagram abstractly represents a scheme. With its multiple leftward scheme links and consequent high leftward entropy, the double-circled scheme in the center of the figure is correctly identified as a morpheme boundary. Although the far-left scheme-circle in Figure 4.12 also has high leftward entropy, a non-branching path leads rightward. To determine that the far-left scheme is in fact a stem-internal boundary

error, ParaMor's stem-internal filter must follow two rightward links to reach the double-circled scheme.

In summary, ParaMor will flag a scheme, *C*, as likely hypothesizing an erroneous stem-internal morpheme boundary if there exists a non-branching rightward path from *C* leading to a scheme, *C′*, such that ParaMor believes *C′* falls at the left edge of a correct morpheme. Like the suffix-internal morpheme boundary error filter, to discard a cluster of schemes, the stem-internal error filter must flag half of the schemes in a cluster as hypothesizing an incorrect morpheme boundary. Figure 4.13 contains a pseudo-code implementation of ParaMor's stem-internal morpheme boundary error filter. The computational expense of all three of ParaMor's filtering algorithms is negligible—the run time of each is linear in the number of scheme-clusters to consider. The small-cluster filter and both morpheme boundary error filters conclude their analyses in a matter of seconds.

4.5 ParaMor's Paradigm Models

Section 4.4 completes the description of all steps in ParaMor's paradigm identification pipeline. The description of ParaMor's pipeline began in Chapter 3 with the initial scheme search and continued in this chapter with scheme clustering and filtering. Figure 4.14 is a graphical representation of ParaMor's pipelined paradigm discovery algorithms. Beginning at the top of the figure: A monolingual natural language corpus is screened of all words 5 characters or less in length (Section 4.2). From the new corpus of longer types, ParaMor searches a network of candidate paradigms, or schemes, for those which most likely model true inflectional paradigms (Chapter 3). The many overlapping and fragmented scheme models of partial paradigms are then clustered into unified models of individual inflectional paradigms (Section 4.3). And finally, three filtering algorithms remove those clusters which, upon closer inspection, no longer appear to model inflectional paradigms: one filter removes small singleton clusters; while two others examine for errors the morpheme boundaries that the proposed scheme-clusters hypothesize in their licensing types. After these six steps, ParaMor outputs a relatively small and co-

```
FilterStemInternalBoundaryErrors(schemeClusters) {
  foreach (schemeCluster in schemeClusters) {
    [countOfErrorSchemes, countOfNonErrorSchemes] = [0, 0];
    foreach (scheme in schemeCluster) {
      if (boundaryIsLikelyStemInternal(scheme))
        countOfErrorSchemes++;
      else
        countOfNonErrorSchemes++;
    }
    if (countOfErrorSchemes >= countOfNonErrorSchemes)
      schemeClusters.remove(schemeCluster);
  }
  return schemeClusters;
}

// If an unbranching rightward path exists from 'scheme', follow the
// path until either 1) the path forks or 2) we reach a scheme that is
// likely the left edge of a morpheme—i.e. we reach a scheme with high
// leftward entropy. If we do reach a scheme that is a likely left edge
// of a morpheme then return 'true', that is, return that 'scheme'
// likely models an incorrect stem-internal morpheme boundary.
boundaryIsLikelyStemInternal(scheme) {
  currentScheme = scheme;
  while (currentScheme.allCSuffixesBeginWithSameNonNullCharacter()) {
    foreach (cSuffix in currentScheme.cSuffixes) {
      rightwardCSuffixes.add(cSuffix.removeFirstCharacter());
    }
    currentScheme =
      dynamicSchemeNetwork.generateScheme(rightwardCSuffixes);
    if (likelyModelOfAMorphemeLeftEdge(currentScheme))
      return true;
  }
  return false;
}
```

Figure 4.13: A pseudo-code implementation of ParaMor's algorithm to filter out scheme-clusters that likely hypothesize morpheme boundaries for their licensing types that fall internal to true stems.

herent set of scheme-clusters which it believes model the inflectional paradigms of a language.

The remainder of this section assesses the quality of ParaMor's Spanish paradigm induction performance after the full set of pipelined induction steps has been taken. Section 4.5.1 concludes the in-depth examination of ParaMor's Spanish paradigm models that began in Section 4.1.1: Where Section 4.1.1 discussed the initial schemes that ParaMor's

bottom-up search procedure produces, Section 4.5.1 takes a detailed look at ParaMor's final set of filtered Spanish scheme-clusters. Section 4.5.2 then presents an experiment over Spanish data that measures the robustness of ParaMor's paradigm building pipeline to a reduction in the size of the induction corpus. Section 4.6 in this chapter, together with Chapters 5 and 6, will broaden the evaluation of ParaMor to include English, German, Finnish, Turkish, and Arabic.

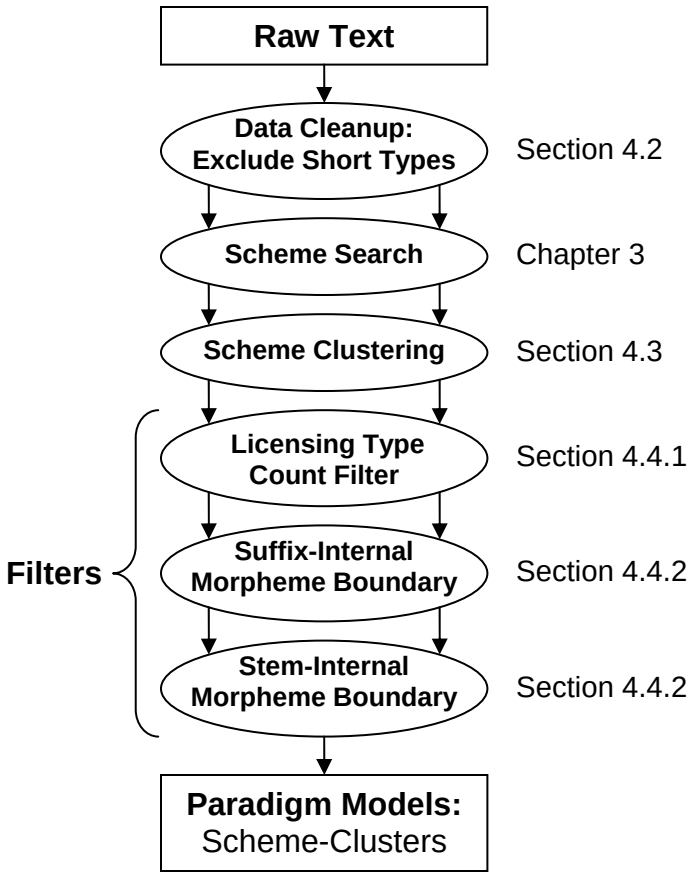

Figure 4.14: The steps of ParaMor's data processing pipeline: Together scheme search, clustering, and filtering transform raw natural language text into models of inflectional paradigms consisting of clusters of schemes.

4.5.1 ParaMor's Final Scheme-Clusters as Viable Models of Paradigms

This section evaluates the final scheme-clusters that ParaMor produces over a Spanish newswire corpus of 50,000. As described in Section 4.4.1, after scheme search, scheme clustering, and the filtering out of small scheme-clusters, ParaMor retains 150 scheme-clusters. Of these 150, the stem-internal and suffix-internal morpheme boundary error filters remove all but 42; And of the 108 scheme-clusters which hypothesized an incorrect morpheme boundary only 12 are not discarded by the morpheme boundary filters. Unfortunately, ParaMor's morpheme boundary error filtering does have collateral damage: Recall of string-unique Spanish suffixes drops from 81.6% to 69.0%. All together, 11 unique c-suffixes that are string identical to Spanish inflectional suffixes given in Appendix A no longer appear in any cluster that ParaMor retains.

Four of these unique c-suffixes were only found in clusters which did not model a Spanish paradigm. Consider the fate of the c-suffix **iste** which is string identical to the verbal **er**/**ir** suffix that marks *2nd Person Singular Past Indicative*. The only cluster in which the c-suffix **iste** occurs is **iste.isten.istencia.istente.istentes.istiendo.istir.-istió.istía**—a cluster which ParaMor's stem-internal morpheme boundary error filter correctly deletes. This **iste**-containing cluster is an incorrect segmentation of verb forms whose stems end in the string **ist**. Among the licensing types of this **iste**-containing cluster are **consiste**, **existe**, **persiste**, etc. Where this scheme-cluster hypothesizes the segmentations are **cons +iste**, **ex +iste**, etc., the correct segmentations of these words removes only the word-final **e**, an inflectional morpheme which marks *3rd Person Singular Present Indicative*.

But, unfortunately, 7 of the 11 lost unique c-suffixes model true Spanish suffixes. All 7 of these lost c-suffixes model Spanish pronominal clitics. And all 7 were lost when the only cluster which modeled these Spanish clitics was removed by the suffix-internal morpheme boundary error filter. The specific cluster that was removed is: **Ø.a.emos.la.-las.le.lo.los.me.on.se.á.án.ía.ían**. In this cluster, the c-suffixes **la**, **las**, **le**, **lo**, **los**, **me**, and **se** are all pronominal clitics, the c-suffix **Ø** correctly captures the fact that not all Spanish verbs occur with a clitic pronoun, and the remaining c-suffixes are incorrect segmentations of verbal inflectional suffixes.

While it is worrisome that an entire category of Spanish suffix can be discarded with a single mistake, Spanish clitics had two counts against them. First, ParaMor was not designed to retrieve rare paradigms, but pronominal clitics are very rare in formal Spanish newswire text. And second, the pronominal clitics which do occur in newswire text almost exclusively occur after an infinitive morpheme, usually **ar**. Exacerbating the problem, the c-suffixes which appear alongside the clitics in this cluster are incorrect suffix-internal segmentations whose c-stems also end in **r**. As so many c-stems in the **Ø.a.-emos.la.las.le.lo.los.me.on.se.á.án.ía.ían** cluster end in **r**, the suffix-internal boundary-error filter believes this cluster to hypothesize an erroneous morpheme boundary. ParaMor's bias toward preferring the left-most plausible morpheme boundary will fail whenever the c-suffixes of a cluster consistently follow the same suffix, or even when they consistently follow the same set of suffixes which all happen to end with the same character. This is a weakness of ParaMor's current algorithm. Note however, that ParaMor retains clusters that contain c-suffixes that model cross-product sequences of verbal inflectional suffix + clitic. For example, turning to Figure 4.15, one cluster that ParaMor retains is the scheme-cluster on the 4th row of Figure 4.15. This 4th row cluster contains such inflectional suffix + clitic cross-product c-suffixes as: **arla**, **arse**, and **ándose**.

Figure 4.15 is a reprint of Figure 4.4 from p. 118 which contained a sampling of 17 clusters that ParaMor constructed from initially selected schemes before any filtering. Keeping in mind ParaMor's full paradigm induction pipeline, here are the fates of the scheme-clusters first introduced in Figure 4.4: The cluster on the top row of Figure 4.4 (or Figure 4.15) models the most prevalent inflection class of ***Number*** on Spanish nouns and adjectives, containing the scheme **Ø.s**. This 1st cluster is correctly retained after all filtering steps. The 2nd scheme-cluster in Figure 4.15 incorrectly places a morpheme boundary after the epenthetic vowel **a** which leads off most suffixes in the **ar** inflection class. ParaMor's suffix-internal morpheme boundary error filter correctly and successfully removes this 2nd cluster. ParaMor correctly retains the scheme-clusters on the 3rd, 4th, 7th, and 8th rows of Figure 4.15. These clusters have respectively the 3rd, 4th, 11th, and 17th most licensing types of any cluster that ParaMor builds. The 3rd scheme-cluster cov-

ers the scheme, **a.as.o.os**, which models the cross-product of ***Gender*** and ***Number*** on Spanish adjectives, and which was the 2nd selected scheme during ParaMor's initial search. The other candidate suffixes in this cluster include a version of the adverbial suffix (**a**)**mente**, and a number of derivational suffixes that convert adjectives to nouns. The 4th, 11th, and 17th scheme-clusters in Figure 4.15 are correct collections of inflectional and derivational suffixes that correspond to, respectively, the verbal **ar**, **er**, and **ir** paradigms.

The 5th scheme-cluster in Figure 4.15 segments a Spanish nominalization internally. But ParaMor's morpheme boundary filters are unsuccessful at removing this scheme-cluster because this Spanish nominalization suffix has four allomorphs: **sion**, **cion**, **sión**, and **ción**. The 5th scheme-cluster places a morpheme boundary immediately before the **i** in these allomorphs. The suffix-internal morpheme boundary error filter is unable to remove the cluster because some c-stems end in **s** while others end in **c**, increasing the leftward link entropy. But the stem-internal boundary error filter is also unable to remove the cluster, because, from a majority of the schemes of this cluster, after following a rightward link through the initial **i** of these c-suffixes, ParaMor's stem-internal error filter reaches a scheme that, although still morpheme-internal, has two rightward trie-style paths, one following the character **o** and one following the character **ó**.

In fact, the largest class of errors in ParaMor's remaining 42 scheme-clusters consists of clusters which, like the 5th cluster, involve allomorphic variation in their c-stem or c-suffix sets: Thirteen final clusters contain allomorphic errors. In Figure 4.15, in addition to the 5th cluster, the cluster on the 9th row, ranked 21st for the size of its set of licensing types, is also led astray by an allomorphic alternation—this time it is the stem that has multiple allomorphic surface forms. In the orthography of some Spanish verbs, stem-final **c** allomorphically becomes **zc**. The 21st cluster hypothesizes a morpheme boundary to the left of the true stem boundary, erroneously including stem-final characters within its c-suffixes. The only way ParaMor can model an allomorphic stem change is by expanding the c-suffixes of a scheme or cluster to include the variable portion of the verb stems. Both the 5th cluster and the 21st cluster are marked in the **ALLO**. column of Figure 4.15.

Rank	Corresponds to Figure 4.1	Licensing Types	# of Schemes	Model of N	Model of Adj	Model of Verb ar	Model of Verb er ir	Deriv. Allo.	Good	Err. Stem	Err. Suffix	C-Suffixes	C-Stems
1	1	11055	3	●	●			●	●			**4** Ø menente mente s	**5401** apoyada barata hombro inevitable segura …
2	2	4740	23			●		●			●	**37** Ø ba ban cion ciones ción da das do dor dora doras dores dos miento mos n ndo r ra ran remos rla rlas rle rles rlo rlos rme rnos ron rse rá rán ré ría rían	**1805** apoya compra disputa genera lanza lleva reclama senti utiliza visite …
3	3, 11	4142	4		●			●	●			**11** a amente as illa illas o or ora oras ores os	**976** apoyad captad frank guerr negociad …
4	5 12 (400)	1909	23			●		●	●			**41** a aba aban acion aciones ación ada adas ado ador adora adoras adores ados amos an ando ante antes ar ara aran aremos arla arlas arlo arlos arme aron arse ará arán aré aría arían ase e en ándose é ó	**113** apoy celebr desarroll genera hall lanz llev public realiz termin utiliz visit …
5	-	1095	16					● ●				**22** ion iona ionaba ionada ionadas ionado ionados ionamiento ionamientos ionan ionando ionantes ionar ionario ionaron ionará ionarán ione ionen iones ionó ión	**418** administrac condic cuest emoc gest les ocas pres reacc sanc ses vinculac …
10	10	722	4		●			●		●		**7** ta tamente tas titud to tos tísimo	**324** cier direc gra insóli len modes sangrien ...
11	30 127 (135)	720	17			●		●	●			**29** e edor edora edoras edores en er erlo erlos erse erá erán ería erían ida idas ido idos iendo iera ieran ieron imiento imientos iéndose ió í ía ían	**62** abastec aparec crec cumpl desconoc es-cond ofrec ocurr permit pertenec recib reun sal suspend transmit vend …
17	1592	384	11				●	●	●			**20** ida idas ido idor idores idos imos ir iremos irle irlo irlos irse irá irán iré iría irían ía ían	**39** abat conclu cumpl distribu eleg exclu permit persegu recib reun segu transmit …
21	1000	344	18			●		● ●				**29** ce cedores cemos cen cer cerlo cerlos cerse cerá cerán cería cida cidas cido cidos ciendo ciera cieran cieron cimiento cimientos cimos ció cí cía cían zca zcan zco	**26** abaste apare agrade compare conven est-able fortale ofre pertene recono ven …
100	-	82	2	●						●		**4** eta etas eto etos	**33** atl bol concr libr metrall papel secr tarj …
122	(40)	58	2	●								**6** Ø es idad idades mente ísima	**15** casual fort important irregular primer …
200	-	35	1							●		**5** staba stado star staron stará	**7** co conte entrevi gu in manife pre
300	-	30	1									**5** pondrán pone ponen poner puesta	**6** com dis ex im pro su
1000	-	15	1							●		**3** dismo dista distas	**5** bu ovie parti perio perre
2000	-	12	1							●		**3** ral rales ralismo	**4** electo libe neolibe plu
3000	-	8	1							●		**2** unas unos	**4** alg fa oport vac
5000	-	6	1							●		**2** ntra ntrarse	**3** ade ce conce
…		…										**…** …	**…** …

Figure 4.15: Figure 4.4 Revisited: Typical clusters produced by ParaMor over a Spanish corpus of 50,000 types.

Nearing the end of Figure 4.15, the scheme-cluster on the 6th row of Figure 4.15, with the 10th most licensing types and the scheme-cluster on the 10th row, with the 100th most licensing types hypothesize morpheme boundaries in adjectives at stem-internal positions. Both the 10th and the 100th clusters are rightly removed by ParaMor's stem-internal morpheme boundary error filter. Correctly, neither the suffix-internal nor the stem-internal morpheme boundary filter removes the scheme-cluster with the 122nd most licensing types, which models *Plural* ***Number*** on nouns. Finally, as mentioned in Section 4.4.1, the last six scheme-clusters in Figure 4.15 are correctly removed by ParaMor's cluster size filter that looks at the number of licensing types in a cluster.

In summary, of the scheme-clusters in Figure 4.15 (and Figure 4.4), ParaMor retains only those clusters with the 1st, 3rd, 4th, 5th, 11th, 17th, 21st, and 122nd most licensing word types; and of these, all but the 5th and 21st exactly model true Spanish morphological paradigm. Overall, the scheme-clusters that ParaMor produces as models of Spanish paradigms are generally quite reasonable. In the course of ParaMor's paradigm induction pipeline, ParaMor builds models of all major inflectional paradigms of Spanish including: both major sub-paradigms marking ***Number*** on nouns, the paradigm cross-product of ***Gender*** and ***Number*** on adjectives, and all three sub-paradigms of Spanish verbs. In Chapter 5, ParaMor will take the paradigms that are built for any particular natural language and morphologically segment word-forms of that language.

4.5.2 Paradigm Learning and Vocabulary Size

As a final examination of the paradigms that ParaMor builds from a Spanish corpus, this section considers the lower limit of the vocabulary size from which ParaMor can confidently learn morphological paradigm structure. Where the majority of this and the previous chapter held the vocabulary size constant and focused on devising strong algorithms for identifying the paradigms of a language in an unsupervised fashion, this section takes ParaMor's full paradigm identification pipeline as given (see Figure 4.14 on p. 144) and investigates the effects of limiting the size of the vocabulary.

While ParaMor's algorithms are accepted as given, this vocabulary experiment still requires careful consideration of how to set ParaMor's free parameters. ParaMor has four free parameters: one parameter thresholds the scheme c-stem ratios in ParaMor's bottom-up initial search phase; one parameter sets the entropy value at which ParaMor discards scheme-clusters that seem to model incorrect morpheme boundaries; one parameter sets the minimum word length for the vocabulary; and finally, one parameter thresholds the minimum number of word types which must license a scheme-cluster. Of these four, the only parameter which need vary with vocabulary size is the cutoff on the minimum number of word types that a cluster must contain. In thresholding a ratio of c-stems, the parameter controlling ParaMor bottom-up search is inherently agnostic to absolute counts; as described in Section 4.4.2 the threshold over morpheme boundary errors is already set conservatively; and this experiment will leave unchanged the requirement that all input words be at least five characters long. The cluster size threshold, on the other hand *is* sensitive to vocabulary size—with fewer types in the induction vocabulary, a scheme built from the same c-suffix set will likely have fewer adherent c-stems. And, indeed empirically, when the cluster size threshold is left at the value set in Section 4.4.1, namely 37, ParaMor removes an unreasonable number of scheme-clusters at lower vocabulary sizes. Hence, for this experiment, the cluster size threshold is linearly scaled with vocabulary size such that at a vocabulary of zero types the cluster size threshold would also be zero.

This section uses two metrics to evaluate ParaMor's performance at identifying the suffixes of true paradigms. First, to gauge success at uncovering suffixes, ParaMor's final scheme-clusters are measured for global recall of Spanish inflectional suffixes, where global recall is defined as the recall in any scheme-cluster of any string-unique suffix from a true paradigm. Second, to evaluate ParaMor's ability to succinctly group suffixes into paradigms, this section reports the final number of clusters ParaMor identifies.

It is reasonable to expect suffix recall to drop somewhat as the size of the induction vocabulary falls simply because in a smaller vocabulary the more rare suffixes may simply not occur. The more serious question for ParaMor is whether there is a vocabulary size at which ParaMor's morphology induction algorithms break down; a lower limit on vocabulary size where ParaMor is unable to identify even those suffixes which *do* occur

in the data. Similarly, in measuring the number of clusters ParaMor produces, the objective is to discover a vocabulary size at which ParaMor begins to fail at reliably grouping suffixes into paradigms. ParaMor's final cluster count would reveal a failure to form paradigms if, as vocabulary size decreased, the number of clusters grew unreasonably.

Figure 4.16 plots ParaMor's paradigm identification performance over a corpus of Spanish as the vocabulary size varies down from 50,000 types to 5,000. Three separate series are plotted in Figure 4.16. The bottom, dashed, line reports ParaMor's final cluster count at each vocabulary size, while the two upper lines report recall scores. The central,

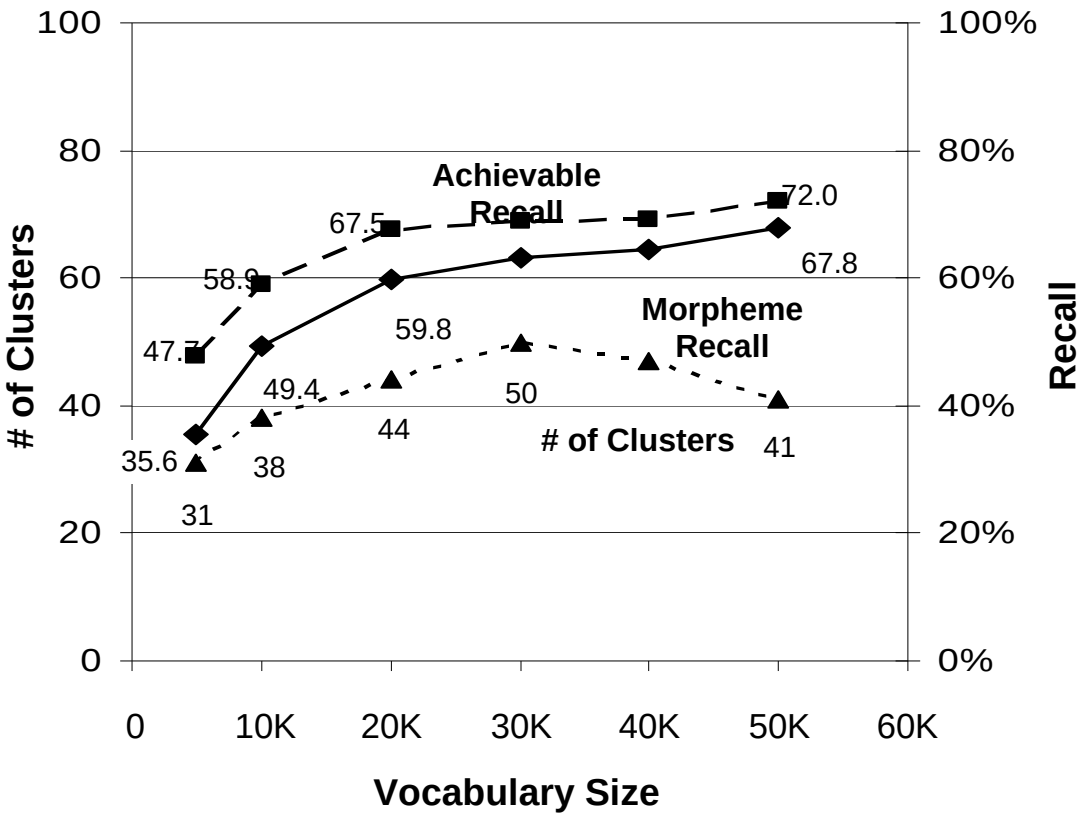

Figure 4.16: A learning curve – Spanish morpheme recall and the number of paradigm-clusters ParaMor identifies as the vocabulary size varies from 50,000 down to 5,000. Achievable Recall is calculated out of those valid Spanish suffixes which occur at least twice in the induction vocabulary as word-final strings.

solid, line is ParaMor's global recall of all string-unique Spanish suffixes. The upper, dashed, line gives ParaMor's global recall of all suffixes that occurred as a word-final string on at least two types in that vocabulary—it seems reasonable to allow any morphology induction system to discount as noise those morphemes that only occur once in a corpus. Note, however, that when the induction vocabulary is varied, as in this experiment, the set of suffixes that occur at least twice in the vocabulary will change. In particular, because the smaller vocabularies in this experiment are subsets of the larger vocabularies, the set of achievable suffixes can only decrease as the vocabulary size shrinks. Also, since the number of achievable suffixes will never be larger than the full set of suffixes in a language, achievable recall will never fall below the strict global recall.

Look first at the central solid global suffix recall curve in Figure 4.16. While suffix recall does fall as vocabulary size decreases, the fall is initially gradual. Reducing the vocabulary size from 50,000 types to 20,000 results in only an 8.0% absolute drop in global recall: from 67.8% to 59.8%. Reducing the vocabulary size further begins a more precipitous decline in recall. Moving from 20,000 to 10,000 vocabulary types reduces recall by 10.4% absolute, down to 49.4%; and then moving to 5,000 reduces recall by an additional 13.8% absolute, falling all the way to 35.6%.

ParaMor's performance at *achievable* recall supports the conclusion that ParaMor's morphology induction algorithms are less robust at finding Spanish suffixes when the vocabulary size is below 20,000 types. From 50,000 types down to 20,000 achievable recall declines only 4.5% absolute. But then, like global recall, when provided with 10,000 unique types, achievable recall falls significantly more steeply, by 8.6%; and again by an additional 11.2% when moving to 5,000 types. In all, ParaMor identifies less than half, just 47.7%, of all true suffix strings that occurred two or more times in the corpus of 5,000 unique types. At these smaller vocabulary sizes, ParaMor's algorithms are no longer able to find even those suffixes that *are* present in the corpus.

For ParaMor to find a suffix there must be paradigmatic and syntagmatic evidence for that suffix. At a minimum, ParaMor's initial search strategy cannot identify any c-suffix as likely modeling a true suffix without:

1. The syntagmatic evidence of the c-suffix appearing attached to at least three c-stems, and without
2. The paradigmatic evidence of those same three c-stems all accepting some other c-suffix.

At vocabulary sizes below 20,000 types in Spanish, these paradigmatic and syntagmatic requirements are just not met.

Turn now to ParaMor's performance at grouping suffixes into paradigms. Empirically, as the vocabulary size decreases from 50,000 types, the number of clusters that ParaMor groups the selected c-suffixes into first moves up from 41 to 50 and then back down to 31. But examining this empirical behavior in two halves reveals an interesting behavior: When the vocabulary size is above 20,000 types ParaMor's cluster count is relatively stable between 41 and 50; Then, as suffix recall falls off at vocabulary sizes below 20,000, the number of clusters that ParaMor identifies also drops. At the smaller vocabulary sizes, the number of clusters roughly follows the number of identified suffixes. It may be that ParaMor is reasonably grouping into paradigms those suffixes that ParaMor is able to identify. Indeed, although individual clusters contain fewer c-suffixes and c-stems when induced from a small vocabulary, a qualitative analysis of ParaMor's Spanish scheme-clusters could find no significant difference in the paradigmatic coherence of individual clusters formed from 50,000 vocabulary types and those formed from 5,000.

Although paradigmatic coherence of scheme-clusters does not significantly change at smaller vocabulary sizes, there is one crucial difference between the clusters formed from large and those formed at small vocabularies—but this difference is a failure of suffix identification not a failure of paradigm grouping. At smaller vocabulary sizes, clusters for some paradigms are simply missing. At 10,000 types there is no paradigm that corresponds to the **Ø.es** paradigm that marks ***Number*** on Spanish nouns, and at 5,000 types ParaMor also finds no scheme-cluster to model those suffixes of the Spanish **ir** verbal paradigm that are not shared with the **er** paradigm.

In summary then, the preceding analysis finds that, at least for Spanish, to identify a significant fraction of the suffixes of a language requires a corpus of at least 20,000

unique types. However, with some confidence, ParaMor will organize the suffixes that *are* identified into reasonable paradigm-like structures, even for very small vocabulary sizes. All of the analysis in this section has been over Spanish. For languages with a morphological system different than that of Spanish, it may be that a vocabulary larger than 20,000 is necessary. Section 6.2 investigates the question of the necessary vocabulary size for English and German.

4.6 Scheme-Clusters in Languages Beyond Spanish

Thus far, this thesis has described ParaMor's performance at paradigm identification by closely examining the paradigm models that ParaMor constructs when analyzing Spanish text. But ParaMor's unsupervised induction methods are intended to enable paradigm discovery in any language. Quantitatively assessing the quality of ParaMor's induced paradigms for a language, as has been done for Spanish throughout Chapters 3 and 4, requires compiling by hand a definitive set of the paradigms of a language. Deciding on a single set of productive inflectional paradigms can be difficult even for a language with relatively straightforward morphology such. Appendix A describes the challenge of deciding whether Spanish pronominal clitics are inflectional paradigms. Moreover, for an agglutinative language like Turkish, the combinatorial number of potential suffix sequences makes a single list of paradigm cross-products extremely unwieldy. Hence, rather than separately define paradigm sets for each language that ParaMor analyzes in this thesis, Chapter 5 will apply ParaMor's induced paradigm models to the task of word-to-morpheme segmentation. And Chapter 6 will empirically evaluate ParaMor's morphological segmentations in English, German, Finnish, Turkish, and Arabic.

Nevertheless, it is enlightening to look at a few global statistics describing the kinds of scheme-clusters that ParaMor builds over a range of languages. Figure 4.17 tabulates, for all six natural languages examined in this thesis, the counts of schemes (or scheme-clusters) that ParaMor constructs after each step in ParaMor's paradigm induction pipeline.

# of Schemes / Clusters	Spanish	English	German	Finnish	Turkish	Arabic
Search	8339	8767	10778	10774	10340	12528
Excluding Short Types	6909	6733	9851	10495	9618	14247
Clustering	6087	6396	9145	8797	6898	12744
Small Cluster Filtering	150	193	206	280	258	87
Morpheme Boundary Filtering	42	40	48	116	100	27

Figure 4.17: The number of schemes or scheme-clusters that ParaMor produces for six languages after each search, clustering, or filtering step in ParaMor's paradigm induction pipeline.

The six languages in Figure 4.17, Spanish, English, German, Finnish, Turkish, and Arabic, fall into three general categories. First, Spanish, English, and German have relatively simple morphological structure that is primarily suffixing. Of these first three languages, English has the least inflectional morphology, and German has the added complication of productive, written, compounding. The second group of languages examined for this thesis encompasses Finnish and Turkish: These two languages have rich morphological structures of agglutinative suffixes. In agglutinative languages multiple affixes (suffixes in the case of Finnish and Turkish) can occur on a single surface word form one after the next. Finally, among this set of six languages Arabic is in a class by itself. Like Finnish and Turkish, Arabic has a rich morphological structure. But Arabic words contain not only suffixes, but also prefixes and templatic morphology that interleaves sequences of vowels and consonants to form new words.

The scheme and cluster counts of Figure 4.17 reflect this three-way classification of these six languages. The counts for Spanish, English, and German are roughly similar. In

particular, the final sets of scheme-clusters that ParaMor produces for these three languages are in a similar range, 42 for Spanish, 40 in English, and 48 for German. The higher counts of German schemes and scheme-clusters that occur on the first three rows of Figure 4.17, before ParaMor's 'Small Cluster Filtering' (Section 4.4.1) and 'Morpheme Boundary Filtering' (Section 4.4.2) steps, are primarily due to schemes that model morpheme boundaries between compound nouns. But noun compounding does not exhibit the same paradigmatic regularities as inflectional morphology and many of the compounding scheme-clusters are discarded during ParaMor's filtering steps.

The scheme and cluster counts for the agglutinative languages Finnish and Turkish are also quite similar to one another. ParaMor's processing of both languages finishes with cluster counts about double those for Spanish, English, or German, at or just above 100. Considering that Finnish and Turkish have more complex morphological structures than Spanish, English, or German, it is to be expected that ParaMor requires additional scheme-clusters to adequately describe the morphology of these two languages.

Finally, ParaMor's scheme-cluster counts for Arabic follow a pattern unlike that for any of the other languages. Initially, ParaMor's bottom-up scheme-search procedure selects more schemes for Arabic than for any other language. But most of these schemes are discarded in the 4th row of Figure 4.17 by ParaMor's 'Small Cluster Filtering' (Section 4.4.1). And ParaMor ultimately retains the fewest scheme-clusters for Arabic out of any language, just 27. The many clusters that ParaMor discards for Arabic primarily attempt to model morpheme boundaries between *prefixes* and stems. These erroneous prefix models place prefixes into schemes' c-stem sets and stems into schemes' c-suffix sets. While ParaMor's initial search finds enough 'paradigmatic' evidence to select these prefixational schemes, ParaMor's paradigmatic clustering algorithm is unable to meaningfully group them, and consequently the vast majority of these schemes are removed by ParaMor's filtering steps.

This preliminary look at ParaMor's performance at paradigm identification for languages beyond Spanish is preparation for their more extensive evaluation in the next two chapters. Chapter 5 will lay the groundwork, defining a methodology to segment words into constituent morphemes that uses the scheme-cluster models of paradigms that Pa-

raMor produces when following the paradigm induction pipeline described in Chapters 3 and 4. Chapter 6 then reports quantitative evaluations of ParaMor's morphological segmentations.

Chapter 5: Morphological Segmentation

Chapters 3 and 4 presented the steps of ParaMor's paradigm discovery pipeline. This chapter and the next will apply ParaMor's induced paradigms to the natural language processing task of morphological word segmentation. The job of a morphological segmentation algorithm is to break full form words at morpheme boundaries. For example, the correct segmentation of the Spanish word **apoyar** would be **apoy +ar**. This **apoy +ar** segmentation is correct because in the word **apoyar** '*to support*', the stem **apoy** carries the lexical meaning of '*support*', while the **ar** marks *Infinitive*. Crucially, however, a morphological segmentation algorithm is not required to associate morphosyntactic features with the morpheme pieces that it proposes: When segmenting the word **apoyar** a word-to-morpheme algorithm is not asked to notate the meaning of the stem **apoy**, nor must a segmentation algorithm state that **ar** signifies the *Infinitive*.

While not a full morphological analysis from a linguistic perspective, morphological segmentation can nonetheless advance performance in a variety of natural language processing tasks. Oflazer and El-Kahlout (2007) improve a Turkish-English statistical ma-

chine translation system by morphologically segmenting Turkish. The morphological segmentation system that Oflazer and El-Kahlout use is hand-built. In contrast, Creutz (2006) significantly improves a Finnish speech recognition system using an unsupervised morphology induction system called Morfessor to morphologically segment the training data for the speech system's language model. Linguistically naïve word stemming, a crude form of morphological segmentation, is also standard in information retrieval. And Section 6.4.2 of this thesis discusses a simple embedding of ParaMor's unsupervised morphological segmentations into an information retrieval system that gives promising results.

For two reasons, the paradigm models that ParaMor produces are well-suited to the task of word-to-morpheme segmentation. First, while ParaMor can identify morpheme segments within word forms, ParaMor is unable to provide a more complete morphosyntactic analysis of words: ParaMor does not associate morphosyntactic features with the c-suffixes in each discovered cluster. ParaMor might know that the c-suffix **ar** attaches to a class of c-stems including **apoy**, but ParaMor does not know that **apoy** carries the lexical meaning of '*support*', or even that the **apoy**-class of c-stems models Spanish verbs; nor does ParaMor know that the **ar** suffix forms the *Infinitive*. Second, ParaMor's paradigm models are best suited to morphological tasks, like segmentation, that analyze word forms, as opposed to generation tasks that propose novel surface forms. As noted in Section 4.3.3, most of ParaMor's clusters contain c-suffixes which do not form valid surface words with all the c-stems in the cluster. These invalid (c-stem, c-suffix) pairs are incorrect generalizations that would hurt performance in a morphological generation task.

Despite the suitability of ParaMor's scheme-cluster models of paradigms to the task of morphological segmentation, ParaMor's segmentation algorithm must be carefully devised so as to segment a wide range of words. Even though clustering introduces sometimes erroneous model generalization, the scheme-clusters that ParaMor produces remain highly specific. By associating a set of c-suffixes with a particular set of c-stems, a strict interpretation of ParaMor's paradigm models would constrain ParaMor's analyses only to the word types covered by each (c-stem, c-suffix) pair in the scheme-cluster. Section 5.1 proposes a segmentation algorithm that successfully generalizes overly specific scheme-

clusters to morphologically segment the wide range of word forms that occur in a large corpus.

5.1 The Segmentation Algorithm

Two principles guide ParaMor's approach to morphological segmentation. First, ParaMor only segments word forms when the discovered scheme-clusters hold paradigmatic evidence of a morpheme boundary. Second, ParaMor's segmentation algorithm must generalize beyond the specific set of types which license individual scheme-clusters. In particular, ParaMor will be able to segment word types which did not occur in the data from which ParaMor induced its scheme-clusters.

ParaMor's segmentation algorithm is perhaps the most simple paradigm inspired segmentation algorithm possible that can generalize beyond the specific set of licensing types in each scheme-cluster. To segment any word, *w*, ParaMor examines all segmentations of *w* into an initial non-null stem, *t*, and a final non-null suffix, *f*, such that $t.f = w$. For each such stem + suffix segmentation of *w*, ParaMor identifies all scheme-clusters, *C*, that contain *f* as a c-suffix. If there is some second c-suffix, f', in *C* such that $t.f'$ is a word form found either in the corpus from which ParaMor induced its scheme-clusters or else found in the corpus of text that ParaMor is to segment, then ParaMor segments *w* between *t* and *f*.

The rationale behind ParaMor's segmentation algorithm is that since *f* and f' are mutually substitutable suffixes from the same induced paradigm model, ParaMor has found paradigmatic evidence of a morpheme boundary. Note that the c-suffix f' need not arise from the same original scheme as *f*, but merely from the same scheme-cluster *C*.

If ParaMor finds no complex analysis, then ParaMor proposes *w* itself as the analysis of the word. On the other hand, if ParaMor discovers more than one potential morpheme boundary in *w*, ParaMor accepts them all—producing a single analyzed form of *w* containing multiple morpheme boundaries. Figure 5.1 gives a pseudo-code implementation of ParaMor's word-to-morpheme segmentation algorithm.

```
// the substring(String, startIndex, endIndex) method works as follows:
//   substring("abcde", 0, 2) yields "ab"
//   substring("abcde", 2, 5) yields "cde"
//
segmentWords(
    words, schemeClusters, paradigmInductionCorpus, corpusToSegment) {

  foreach (word in words) {
    morphemeBoundaryIndexes = emptySet;
    wordLength = word.length();

    for (charIndex = 1; charIndex <= wordLength-1; charIndex++) {
      stem = substring(word, 0, charIndex);
      suffix = substring(word, charIndex, word.length());
      foreach (schemeCluster in schemeClusters) {
        foreach (cSuffix in schemeCluster) {
          if (cSuffix == suffix) {
            foreach (cSuffixPrime in schemeCluster) {
              if (cSuffixPrime != cSuffix) {
                possibleWord = stem + cSuffixPrime;
                if (paradigmInductionCorpus.contains(possibleWord) ||
                    corpusToSegment.contains(possibleWord)) {
                  morphemeBoundaryIndexes.add(charIndex);
    } } } } } } } // end 'for (charIndex = 1...'

    // segmentedWords is a hash on each word to an array which holds
    // the morphemes of the word.
    startIndex = 0;
    foreach (boundaryIndex in morphemeBoundaries) {
      morpheme = substring(word, startIndex, boundaryIndex);
      segmentedWords{word}.add(morpheme);
      startIndex = boundaryIndex;
    }

    // And add the final segment of the word as a final morpheme
    morpheme = substring(word, startIndex, word.length();
    segmentedWords{word}.add(morpheme);
  }

  return segmentedWords;
}
```

Figure 5.1: Pseudo-code implementing ParaMor's word segmentation algorithm.

5.2 A Sample of ParaMor's Word Segmentations

Figure 5.2 contains examples of Spanish word-to-morpheme segmentations that ParaMor produces. ParaMor segmented the word forms of Figure 5.2 using scheme-cluster models of paradigms that were induced over the same newswire corpus of 50,000 types used throughout Chapters 3 and 4. But the segmented words in Figure 5.2 come from a larger newswire corpus of 100,000 types which includes the 50,000 type corpus as a subset. Each row of Figure 5.2 contains segmentation information about a single word. The word forms of the first thirteen rows of Figure 5.2 were hand selected to illustrate the range of analyses that ParaMor's segmentation algorithm is capable of. And then, to provide a flavor for the typical segmentations that ParaMor outputs, the forms in the last six rows of Figure 5.2 were randomly selected from the words in ParaMor's segmentation corpus of 100,000 unique word forms.

Starting with the leftmost column, each row of Figure 5.2 specifies:

1. The row number;
2. A particular Spanish word which ParaMor segmented;
3. An English gloss for that word form;
4. The word's correct morphological segmentation;
5. A full morphosyntactic analysis of the Spanish word form;
6. Each separate morpheme boundary that ParaMor matches;
7. ParaMor's full segmentation of the word; and
8. The rank of one or more scheme-clusters which provide paradigmatic support for ParaMor's segmentation of that row's word form.

Elaborating on the 8th column of Figure 5.2: Whenever a scheme-cluster from Figure 4.4 on p. 118 leads ParaMor to propose a morpheme boundary in Figure 5.2, then the rank of the Figure 4.4 cluster is given in the final column of Figure 5.2. In the few cases where no Figure 4.4 cluster supports a morpheme boundary that is proposed in Figure 5.2, then the rank of a supporting cluster appears in parentheses. Many morpheme boundaries that Pa-

	Row	Word Form	Gloss	Correct Segmentation	Morphosyntactic Analysis	Individual Boundaries	ParaMor's Segmentation	Cluster Rank
Illustrative Word Forms	1	sacerdote	priest	sacerdote	sacerdote	sacerdote	sacerdote	-
	2	sacerdotes	priests	sacerdote +s	sacerdote +pl	sacerdote + s	sacerdote +s	1
	3	regulares	ordinary	regular +es	ordinary +pl	regular +es	regular +es	122
	4	chancho	filthy	chanch +o	chancho +masc	chanch +o	chanch +o	3
	5	incógnitas	unknown	incógnit +a +s	incognito +fem +pl	incógnit +as, incógnita +s	incógnit +a +s	1, 3
	6	descarrilaremos	we will be derailed	descarril +aremos	descarrilar +1pl.fut.indic	descarril +aremos	descarril +aremos	4
	7	accidentarse	to have an accident	accident +ar +se	accidentarse +inf +reflex	accident +arse	accident +arse	4
	8	errados	wrong, mistaken (Masculine Plural)	err +ad +o +s	errar +adj +masc +pl	err +ados errad +os errado +s	err +ad +o +s	1, 3, 4
	9	agradezco	I thank	agradec +o	agradecer +1sg.pres.indic	agrade +zco	agrade +zco	21
	10	agradecimos	we thank	agradec +imos	agradecer +1pl.past.indic	agrade +cimos agradec +imos	agrade +c +imos	17, 21
	11	antelación	(in) advance	antel(+)ación	antelar +ción.N	antel +ación antelac +ión	antel +ac +ión	4, 5
	12	tanteador	storekeeper	tante(+)ador	tantear +ador.N	tante +ador	tante +ador	4
	13	vete	he/she should veto	vet +e	vetar +3sg.pres.subjunc	vet +e	vet +e	4
Randomly Selected	14	bambamg	not a Spanish word	bambamg	bambamg	bambamg	bambamg	-
	15	clausurará	he/she will con-clude	clausur +ará	clausurar +3sg.fut.indic	clausur +ará	clausur +ará	4
	16	hospital	hospital	hospital	hospital	hospit +al hospital +l	hospit +a +l	(7), (41)
	17	investido	invested (Masculine Singular)	invest +id +o	investor +adj +masc	invest +ido investi +do invested +o	invest +i +d +o	3, 11, 17, (28)
	18	pacíficamente	peaceably	pacífic +amente	pacíficamente	pacific +amente pacifíca +mente	pacific +a +mente	1, 3, 122
	19	sabiduría	wisdom	sabiduría	sabiduría	sabiduría	sabiduría	-

Figure 5.2: Morphological segmentations of some Spanish word forms produced by ParaMor over a corpus of 100,000 types.

raMor proposes gain paradigmatic support from two or more scheme-clusters. For any particular morpheme boundary, Figure 5.2 only lists the rank of more than one supporting cluster when each supporting cluster appears in Figure 4.4.

The 1st row of Figure 5.2 contains ParaMor's segmentation of the monomorphemic word form **sacerdote** '*priest*'. ParaMor's segmentation algorithm correctly analyzes **sacerdote** as containing no morpheme boundaries. The 2nd row of Figure 5.2 segments **sacerdotes** '*priests*'. Since both the word forms **sacerdote** and **sacerdotes** occurred in the Spanish corpora, and because ParaMor contains a scheme-cluster which contains both of the c-suffixes **s** and **Ø**, namely the cluster from Figure 4.4 with rank 1, ParaMor detects sufficient paradigmatic evidence to successfully suggest a morpheme boundary before the final **s** in **sacerdotes**, giving **sacerdote +s**. ParaMor similarly correctly segments the form **regulares** '*ordinary* (*Plural*)' before the final **es**, giving **regular +es**, using the rank 122 scheme-cluster; and the form **chancho** '*filthy* (*Masculine Singular*)' before the final **o**, yielding **chanch +o**, by drawing on the rank 3 cluster.

The particular forms **sacerdote**, **sacerdotes**, **regulares**, and **chancho** of the first four rows in Figure 5.2 illustrate the ability of ParaMor's segmentation algorithms to correctly generalize. The forms **sacerdote** and **sacerdotes** directly contribute to the **Ø.s** scheme that participates in the rank 1 scheme-cluster of Figure 4.4. And so to segment **sacerdotes** required no generalization whatsoever. On the other hand, the c-stem **regular** does not occur in the rank 122 scheme-cluster, and the c-stem **chanch** does not occur in the rank 3 cluster, but ParaMor was yet able to generalize from the rank 122 cluster and the rank 3 cluster to properly segment the word forms **regulares** and **chancho** respectively. ParaMor segmented **regulares** because: 1. The rank 122 scheme-cluster contains the **Ø** and the **es** c-suffixes; and 2. The word forms **regular** and **regulares** both occur in the corpus from which ParaMor learned its scheme-clusters. The occurrence of **regular** and **regulares** provides the paradigmatic evidence that ParaMor requires to suggest segmentation. ParaMor's justification for segmenting **chancho** is similar to the reasoning behind **regulares** but takes generalization one step further—the form **chancho** did not occur in the corpus from which ParaMor induced paradigms, but only occurred in the larger corpus of 100,000 word types that ParaMor segments.

The 5th row of Figure 5.2 illustrates ParaMor's ability to segment a single word into more than two morphemes. The fifth row contains the *Plural Feminine* form of the Spanish adjective **incógnito** '*unknown*': **incógnitas**. The ***Gender*** is marked by the **a** in this form, while *Plural* ***Number*** is marked in the final **s**. And so, the correct segmentation contains three morphemes and two morpheme boundaries, **incógnit +a +s**. ParaMor's scheme-cluster models of paradigms successfully identify both morpheme boundaries in the word **incógnitas**. Unfortunately, ParaMor's segmentation algorithm is not always so perfect when it proposes multiple morpheme boundaries. In the segmentation of the word form **agradecimos** '*we thank*' as **agrade +c +imos**, on row ten of Figure 5.2, while the morpheme boundary before the final **imos** is reasonable, the character **c** is rightfully part of this verb's stem and should not be segmented off as a suffix.

The 6th row of Figure 5.2 gives an example of the rank 4 scheme-cluster from Figure 4.4 correctly segmenting a Spanish verb; the rank 4 scheme-cluster, capturing suffixes from the Spanish verbal **ar** paradigm, contains more c-suffixes than any other cluster that ParaMor induces over this Spanish corpus. The 7th row in Figure 5.2 is an example of ParaMor's failure to analyze Spanish pronominal clitics. As discussed in Section 4.5.1, ParaMor's suffix-internal morpheme boundary error filter mistakenly discards the scheme-cluster which contains the majority of Spanish clitics. Where there should be a morpheme boundary before the reflexive clitic **se**, in **accidentar +se**, ParaMor places none. ParaMor's correct segmentation of the adjectival verb, **errados** '*wrong* (*Masculine Plural*)' as **err +ad +o +s**, on row 8 of Figure 5.2 contrasts with the incorrect oversegmentation of another adjectival verb, **investido** '*invested* (*Masculine Singular*)' as **invest +i +d +o**, on the table's 17th row. In the segmentation of **investido**, there should be no morpheme boundary between the characters **i** and **d**.

The 9th, 10th, and 11th rows of Figure 5.2 illustrate some of the incorrect segmentations that ParaMor produces from scheme-clusters which involve allomorphic variation. The 9th and 10th rows contain the *1st Person Singular Present Indicative* and the *1st Person Plural Past Indicative* forms, respectively, of the Spanish verb **agradecer** '*to thank*'. The stem of the verb **agradecer** has two written surface forms: before suffixes which begin with the back vowels **a** and **o**, the stem **agradezc** occurs; but, before the front vowels

e and **i**, the stem is **agradec**. In fact, this **c**/**zc** alternation is a feature of a reasonably sized minority of Spanish verbs, and hence, the rank 21 cluster of Figure 4.4 attempts to model this stem alternation by fusing the variable stem material to the front of c-suffixes. Thus, where the *1st Person Singular Present Indicative* morpheme is **o**, ParaMor removes **zco**, giving **agrade +zco**; and where the morpheme marking *1st Person Plural Past Indicative* in **er** verbs (like **agradecer**) is **imos**, ParaMor identifies both a c-suffix that matches the correct **imos** morpheme but also a c-suffix that matches **cimos**, ultimately producing the oversegmented form **agrade +c +imos**.

In contrast to the varying stems of the word forms in the 9th and 10th rows of Figure 5.2, it is the suffix of **antelación**, on the 11th row, that appears in distinct allomorphic surface forms in different words. As discussed in Section 4.5.1, the four related suffixes **ción**, **sión**, **cion**, and **sion** confound ParaMor's morpheme boundary error filters such that ParaMor is unable to remove the 5th ranked scheme-cluster which contains the c-suffix **ión**. The consequence of ParaMor's failure to detect this morpheme boundary error in the scheme-cluster paradigm models is that the word **antelación** is incorrectly oversegmented before the word-final string **ión**, yielding **antel +ac +ión**.

ParaMor was designed to identify inflectional morphemes. Consequently, most of ParaMor's segmentations in Figure 5.2 split off inflectional suffixes. The segmentations that ParaMor gives for the word forms of the 11th and 12th rows of Figure 5.2, however, segment the derivational morphemes **ación** and **ador** respectively. The suffix **ación** forms abstract nouns from verbs, while **ador** is the agentive.

Also, the short word form **vete**, 'veto (*3rd Person Singular Present Subjunctive*)' on the table's 13th row, is correctly segmented by ParaMor even though its four characters excluded it from the corpus from which ParaMor induced scheme-clusters.

The final six rows of Figure 5.2 place ParaMor in the wild, giving segmentations of a small random sample of Spanish words. ParaMor successfully leaves unsegmented the Indonesian proper name **bambamg** and the monomorphemic Spanish word **sabiduría** '*wisdom*'. ParaMor correctly segments the verbal form **clausurará** '*conclude* (*3rd Person Singular Future Indicative*)' as **clausur +ará**; but oversegments the three forms **hospital** '*hospital*', **investido**, '*invested* (*Adjectival Masculine Singular*)' and **pacificamente**

‘*peaceably*’. The oversegmentations of both **investido** and **pacificamente** occur because of morpheme boundary errors that ParaMor makes while attempting to model legitimate morphemes: the *Adjectival Masculine Singular* suffix **ido** on **er** and **ir** verbs, and the productive adjectival suffix **amente**, respectively.

But the double mis-segmentation of the monomorphemic **hospital**, as **hospit +a +l**, is conspicuous. Both of the mis-hypothesized morpheme boundaries in the word **hospital**, that before **al** and that before the final character **l** (el), occur because of an incorrect overgeneralization in ParaMor’s segmentation algorithm. The boundary before the word-final string **al** occurs when ParaMor incorrectly applies a cluster that contains, among other c-suffixes, the adjectival c-suffixes **al** ‘*Adjectival Singular*’ and **ales** ‘*Adjectival Plural*’. This adjectival cluster is derived from Spanish words that include **accidental** ‘*accidental* (*Adjectival Singular*)’ and **accidentales** ‘*accidental* (*Adjectival Plural*)’). Because the Spanish noun **hospital** happens to end in the character sequence **al**, and because the plural form of **hospital** is **hospitales**, ParaMor’s segmentation algorithm overgeneralizes to apply this *adjectival* cluster to segment a *noun*. In a similar fashion, ParaMor places a boundary before the final character **l** in **hospital** because of a cluster that contains incorrect suffix-internal segmentations of these same adjectival **al** and **ales** suffixes—the incorrect cluster contains the c-suffixes **l** and **les**.

5.3 Morpheme Segmentation Enables Evaluation

ParaMor’s word-to-morpheme segmentation algorithm, defined in this chapter, provides a practical way to evaluate, for a range of languages, the quality of ParaMor’s scheme-cluster models of natural language morphological paradigms. Cannonical morphological analyses are available for the words of many natural languages; And ParaMor’s morphological segmentations can be directly compared to these cannonical analyses. Moreover, for a number of natural language processing applications, it is possible to replace raw word-forms with morphological segmentations and to then measure the (hopefully positive) impact of using the segmented forms. Chapter 6 will evaluate ParaMor’s induced scheme-cluster models of paradigms both against cannonical

morphological analyses as well as by embedding ParaMor's morphological segmentations in an information retrieval system.

Chapter 6: ParaMor and Morpho Challenge

With the development, in Chapter 5, of an unsupervised morphological segmentation algorithm, the ParaMor morphology induction system can now analyze the morphology of individual words. To evaluate ParaMor's word-to-morpheme segmentations, ParaMor competed in two years of the Morpho Challenge competition series. In May 2007 and again in June 2008, the Adaptive Informatics Research Center at the Helsinki University of Technology sponsored a Morpho Challenge. This series of Challenges pits against one another algorithms that, like ParaMor, are designed to discover the morphological structure of natural languages from nothing more than raw text (Kurimo, Turunen, and Varjokallio, 2008). Evaluating ParaMor through participation in the Morpho Challenge competitions permits direct comparison of ParaMor's morphological analyses to the analyses produced by other state-of-the-art unsupervised morphology induction systems.

The remainder of this chapter is structured as follows: Section 6.1 will describe the evaluation methodology used in the 2007 and 2008 Morpho Challenge competitions.

Then, using the evaluation methodology of Morpho Challenge, Sections 6.2 and 6.3 examine ParaMor's performance at morphological segmentation from two perspectives: Section 6.2, an ablation study, weighs the contributions that each of ParaMor's major sub-algorithms separately make toward the final morpheme segmentations that ParaMor produces; And Section 6.3 considers ParaMor's ability to identify inflectional, as opposed to derivational, morphology. Finally, Section 6.4 presents ParaMor's performance in the Morpho Challenge competitions proper.

6.1 Evaluation Methodology at Morpho Challenge 2007/2008

The Morpho Challenge competitions of 2007 and 2008 appraised participating algorithms on their morphological analyses of five languages: English, German, Finnish, Turkish, and Arabic. As the ParaMor algorithm was developed while analyzing Spanish data, participation in the Morpho Challenge competitions will allow ParaMor to show language independence.

ParaMor's induction algorithms were specifically designed around the natural organizing structure of inflectional morphology: the paradigm (see Chapters 1 and 3). Hence, ParaMor will likely be able to learn the inflectional structure of languages other than Spanish. However, ParaMor has several free parameters, and it is conceivable that parameter settings that produce strong paradigm models for Spanish will yield imperfect paradigms for other languages. Nevertheless, setting ParaMor's parameters anew for each separate language would void ParaMor status as an unsupervised algorithm. Hence, for all languages, ParaMor holds parameters at the settings which produce reasonable *Spanish* suffix sets, as determined in Chapters 3 and 4. Section 6.4 will demonstrate that these parameter settings do, in fact, reasonably transfer to the languages of the Morpho Challenge competitions.

For each of the five language tracks in the competition, the Morpho Challenge organizing committee provided text corpora with vocabulary sizes much larger than the 50,000 Spanish types that the ParaMor algorithms were developed over. The English corpus contains nearly 385,000 unique types; the German corpus, 1.26 million types; Fin-

nish, 2.21 million; Turkish, 617,000; while the Arabic corpus is the smallest, with just under 143,000 unique types. To increase the likelihood that ParaMor's parameter settings will transfer from Spanish to other language corpora, ParaMor keeps the size of the paradigm induction vocabulary constant at 50,000 types. As ParaMor has a choice over which types to place in the paradigm induction vocabulary, ParaMor selects, from each larger language corpus, the 50,000 most frequent word types that pass ParaMor's string-length criterion (Section 4.2). Once ParaMor has learned paradigm models of a language from the 50,000 most frequent types, ParaMor segments *all* the word types in the Morpho Challenge corpus for that language, following the methodology of Chapter 5.

The Morpho Challenge competitions held in 2007 and 2008 scored each contending algorithm's morphological analyses in two ways: First, a Linguistic Evaluation measured a system's morpheme identification against an answer key of morphologically analyzed word forms (Kurimo and Varjokallio, 2008; Kurimo, Creutz, and Varjokallio, 2008). And second, a Task-Based Evaluation embedded each algorithm's morphological analyses in an information retrieval (IR) system (Kurimo and Turunen, 2008; Kurimo, Creutz, and Turunen, 2007). Sections 6.1.1 and 6.1.2 describe the Linguistic and Task-Based Evaluation procedures of the Morpho Challenge competitions in turn.

6.1.1 The Linguistic Evaluation of Morpho Challenge 2007/2008

While the majority of the unsupervised morphology induction systems, including ParaMor, which have participated in Morpho Challenge competitions, perform simple morphological segmentation, the Linguistic Evaluation of the 2007 and 2008 Morpho Challenge competitions was not purely a word segmentation task. In both years, the Linguistic Evaluation compared each system's automatic morphological analyses against an answer key containing the full morphological analysis of each word form. Although a more challenging standard than word-to-morpheme segmentation, evaluating an unsupervised morphology induction system against full morphosyntactic analyses is less arbitrary than evaluating against an artificial segmentation standard—Only in an idealized world are morphemes consistently strung together in a purely concatenative fashion. In actual natu-

WORD FORM	GLOSS	MORPHOSYNTACTIC ANALYSIS	PARAMOR'S SEGMENTATION
sacerdote	priest	sacerdote	sacerdote
sacerdotes	priests	sacerdote +pl	sacerdote +s
regulares	ordinary	ordinary +pl	regular +es
agradezco	I thank	agradecer +1sg.pres.indic	agrade +zco
agradecimos	we thank	agradecer +1pl.past.indic	agrade +c +imos

Figure 6.1: Full morphosyntactic analyses in the style found in the answer keys of the Linguistic Evaluation at Morpho Challenge, together with ParaMor's Morphological segmentations of five Spanish word forms. ParaMor produced these segmentations when analyzing a Spanish corpus of 100,000 unique types.

ral languages, morphophonological processes often change the surface form of both stem and affix morphemes at the time of affixation; and moreover, in a language like Arabic, morphemes are often non-concatenative to begin with, rendering the idea of a gold-standard morphological segmentation meaningless.

The morphosyntactic answer keys of Morpho Challenge are in the same format as the analyses found in the **MORPHOSYNTACTIC ANALYSIS** column of Figure 6.1. Extracted from Figure 5.2, the rows and columns of Figure 6.1 contain morphological analyses of five Spanish words. The analyses in the **MORPHOSYNTACTIC ANALYSIS** column of Figure 6.1, and the analyses of words in a Morpho Challenge answer key, contain one or more lexical stems and zero or more inflectional or derivational morpheme feature markers. Both stems and feature markers are strings: feature markers have a leading '**+**'. In addition to the Morpho Challenge style morphosyntactic analyses for each of five Spanish words, Figure 6.1 gives an English gloss of the Spanish word and the morphological segmentation that ParaMor produces for the word.

As a morphosyntactic answer key, distinct surface forms of the same morpheme are marked with the same lexical stem or feature marker. For example, Spanish builds the *Plural* of **sacerdote** 'priest' by appending an **s**, while *Plural* is marked on the Spanish form **regular** with **es**. But in both cases, a Morpho Challenge style morphosyntactic an-

swer key would mark *Plural* with the same feature marker—**+pl** in Figure 6.1. The organizing committee of Morpho Challenge designed the linguistic answer keys for each language to contain feature markers for all and only morphosyntactic features that are overtly marked in a word form. Since *Singular* forms are unmarked on Spanish nouns, the Morpho Challenge style analysis of the word **sacerdote** in Figure 6.1 does not contain a feature marker indicating that **sacerdote** is *Singular*. Also important for the ParaMor algorithm is that the morphosyntactic answer keys for the Linguistic Evaluation of Morpho Challenge analyze not only inflectional but also derivational morphology (See Section 6.3).

Against each morphosyntactic answer key, the Linguistic Evaluation of Morpho Challenge assessed systems' precision and recall at identifying the stems and feature markers of each word form. But to calculate these precision and recall scores, the Morpho Challenge Linguistic Evaluation must account for the fact that label names assigned to stems and to feature markers are arbitrary. In Figure 6.1, morphosyntactic analyses mark *Plural **Number*** with the space-saving feature marker **+pl**, but another human annotator might have preferred the more verbose **+plural**—in fact, any unique string would suffice.

Since the names of feature markers, and stems, are arbitrary, the Linguistic Evaluation of Morpho Challenge does not require each unsupervised morphology analysis system to guess the particular names used in the answer key. Instead, to measure recall, the automatic Linguistic Evaluation selects a large number of word pairs such that each word pair shares a morpheme in the answer key. The fraction of these word pairs which also share a morpheme in a system's automatic analyses is the Morpho Challenge recall score for that system. Precision is measured analogously: a large number of word pairs are selected where each pair shares a morpheme in the automatically analyzed words. Out of these pairs, the number of pairs that share a morpheme in the answer key is the precision.

To illustrate the scoring methodology of the Linguistic Evaluation of Morpho Challenge, consider a recall evaluation of the Spanish words in Figure 6.1. To calculate recall, the Linguistic Evaluation routine might select the pairs of words: (**agradezco**, **agradecimos**) and (**sacerdotes**, **regulares**) for sharing in the answer key the stem **agradecer** and the feature marker **+pl**, respectively. ParaMor would get recall credit for its '**agrade**

its '**agrade +zco**' and '**agrade +c +imos**' segmentations as these segmentations share the morpheme string **agrade**. Note here that the stem in the answer key, **agradecer**, is different from the stem ParaMor suggests, **agrade**, but ParaMor still receives recall credit. On the other hand, ParaMor would not get recall credit for the (**sacerdotes**, **regulares**) pair, as ParaMor's segmentations '**sacerdote +s**' and '**regular +es**' do not contain any common pieces.

It is possible for a word pair to share more than one morpheme. In this case the Linguistic Evaluation of Morpho Challenge credits a system's precision or recall with a fractional count of correctly identified morphemes. For example, suppose the morphological answer key for a language states that a word pair (w_1, w_2) shares two morphemes in common, but a particular morphological analysis system analyzes w_1 and w_2 to share just one morpheme. In this situation, the denominator of the recall calculation increments by a full word pair, while the numerator is incremented by one-half a pair—for finding one of the two morphemes.

The Linguistic Evaluation of Morpho Challenge also normalizes precision and recall scores when a surface word has multiple analyses. It is not uncommon for a single word type to be ambiguous between two or more morphological analyses: consider '**he dances**' where **dances** is a *3rd Person Singular* verb and '**the dances**' where **dances** is a *Plural* noun. Because morphology can be inherently ambiguous, the Morpho Challenge Linguistic Evaluation permits for each word:

1. Multiple morphological analyses in the answer key for a language, and
2. A morphological analysis system to propose more than one analysis.

The probability with which the Morpho Challenge Linguistic Evaluation selects any particular word, *w*, for participation in a morpheme-sharing word pair during the calculations of precision and recall is proportional to the number of morphological analyses that *w* has—the more ambiguous morphological analyses there are of *w*, the more pairs that *w* will appear in. To reduce the influence that ambiguous word forms have on precision and recall, the Linguistic Evaluation down-weights word-pairs that contain the ambiguous word *w* by a factor again proportional to the number of analyses that exist for *w*.

Once a precision and a recall score has been calculated, the Morpho Challenge Linguistic Evaluation uses F_1, the harmonic mean of precision and recall, as a single overall performance measure for each algorithm and each language. The official specification of the Linguistic Evaluation procedure for the 2007 and 2008 Morpho Challenge competitions appears in Kurimo, Creutz, and Varjokallio (2008).

6.1.2 The Task-Based Evaluation of Morpho Challenge 2007/2008

The 2007 and 2008 Morpho Challenge competitions balanced the Linguistic Evaluation described in the previous sub-section against a Task-Based Evaluation in which the morphological analyses of each competing unsupervised morphology induction system are embedded in an information retrieval (IR) system. The Task-Based IR Evaluation consists of queries over a language specific collection of newswire articles. To measure the effect that a particular morphological analysis algorithm has on newswire IR, the Task-Based Evaluation replaces all word forms in all queries and all documents with their morphological decompositions, according to that analysis algorithm.

Separate IR tasks were run for English, German, and Finnish, but not Turkish or Arabic. For each language, the IR task made at least 50 queries over collections ranging in size from 55,000 (Finnish) to 300,000 (German) articles. The evaluation data included 20,000 or more binary relevance assessments for each language. The IR Evaluation employed the LEMUR toolkit (Ogilvie and Callan, 2002), a state-of-the-art retrieval suite; and used okapi term weighting (Robertson, 1994). To account for stopwords, terms in each run with a frequency above a threshold, 75,000 for Finnish, 150,000 for English and German, were discarded. The performance of each IR run was measured with Uninterpolated Average Precision. For additional details on the IR Evaluation of Morpho Challenge please reference Kurimo and Turunen (2008) and Kurimo, Creutz, and Turunen (2007).

6.2 An Ablation Study

Before delving into the official results of the Morpho Challenge competitions from 2007 and 2008, this section uses the Morpho Challenge evaluation methodology to explore the contributions that each major sub-piece of ParaMor's paradigm induction algorithm makes toward ParaMor's final word segmentations. ParaMor's algorithm for unsupervised paradigm induction consists of the pipelined sub-algorithms described in Chapters 3 and 4, and summarized in Section 4.5. At a high level, ParaMor's paradigm induction breaks down into three major steps:

1. The initial search for scheme-models of paradigms, as described in Chapter 3;
2. The agglomerative scheme-clustering algorithm, detailed in Section 4.3; and
3. The filtering procedures, Sections 4.2 and 4.4, that were designed to both limit the creation of and to then discard unlikely paradigm models.

To examine the separate impacts that the search, clustering, and filtering algorithms each have on morphological word segmentations, this section evaluates the English and German word segmentations that four configurations of these three sub-algorithms produce. All four configurations begin with ParaMor's initial search: ParaMor's scheme search procedure is prerequisite to both clustering and filtering. The first of the four configurations consists solely of the initial search, skipping entirely both the clustering and all filtering procedures. The second configuration clusters schemes, but does not filter out the poorer candidates. The third configuration filters out unlikely candidate paradigms, but does not cluster the initially selected schemes into coherent paradigmatic groups. And the fourth configuration applies the full suite of algorithms: first searching for candidate schemes, then clustering the candidates into consolidated paradigms, and finally filtering out the least promising clusters.

A few further specifics on the experimental setup are in order. To begin, although Chapter 4 presented four distinct procedures designed to filter out less desirable paradigm models, this section only measures their aggregate effect. In addition to the cluster-size filter that removes clusters with support from few licensing types (Section 4.4.1), and the

two filters designed to detect and discard schemes which hypothesize incorrect morpheme boundaries (Section 4.4.2), this experiment counts as a filtering algorithm the technique described in Section 4.2 that requires all word types in the paradigm induction corpus to consist of more than a threshold number of characters. The corpus word length requirement is categorized as a filtering algorithm here because it was specifically designed to eliminate (i.e. filter) the production of schemes that result from accidental string similarities between corpus word types, see Section 4.2.

Also concerning experimental setup: The order in which ParaMor invokes the scheme-search, filtering, and clustering algorithms is the fixed order presented in Figure 4.14 of Section 4.5. Specifically, the filtering step of excluding short types from the paradigm induction vocabulary always occurs immediately before ParaMor's scheme search; ParaMor's clustering algorithm is run just following scheme search; and after clustering, ParaMor removes scheme-clusters that have support from few licensing word types, discards scheme-clusters that propose morpheme boundaries suffix-internally, and discards clusters whose boundaries fall stem-internally—in that order. When a specific experimental configuration omits the clustering or filtering steps, the clustering algorithm or all filtering procedures are simply skipped in the pipeline.

Figure 6.2 tabulates ParaMor's Morpho Challenge-style precision, recall, and F_1 scores for word segmentation of English and German under each of the four search-cluster-filter configurations. After each precision, recall, or F_1 value, Figure 6.2 gives (in parentheses) the standard deviation for that value. The standard deviation values were obtained by calculating precision, recall, and F_1 scores on multiple non-overlapping sets of 1000 feature-sharing word pairs. For each experimental-configuration of Figure 6.2 and for each language, ParaMor applied the paradigm induction pipeline to a corpus of 50,000 unique word types. With the resultant models of morphological paradigms, ParaMor then segmented the same full data used in the Morpho Challenge 2007/2008 competitions: an English corpus containing almost 385,000 unique words, and a German corpus of 1.26 million types.

The top two lines of Figure 6.2 are the experimental configurations that exclude ParaMor's filtering steps, while the lower two rows include filtering. And the top-most row

			English			German		
Search	**Cluster**	**Filter**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
•			14.2 (±0.5)	82.8 (±1.0)	24.2 (±0.7)	13.1 (±0.4)	78.6 (±1.0)	22.5 (±0.5)
•	•		14.0 (±0.5)	83.0 (±1.0)	24.0 (±0.7)	13.1 (±0.4)	78.6 (±1.0)	22.5 (±0.6)
•		•	**63.0** (±1.0)	47.8 (±1.3)	**54.3** (±0.9)	48.4 (±0.9)	**41.6** (±1.0)	44.7 (±0.6)
•	•	•	57.2 (±1.1)	**48.8** (±1.3)	52.7 (±0.9)	**54.1** (±1.0)	38.9 (±1.0)	**45.2** (±0.6)

Figure 6.2: An ablation study. ParaMor's **P**recision, **R**ecall, and **F_1** scores (and their standard deviations in parentheses) at word-to-morpheme segmentation using four configurations of ParaMor's scheme search, clustering, and filtering algorithms. A dot in the Search, Cluster, or Filter columns indicates that the relevant step(s) in ParaMor's paradigm induction pipeline took part in that configuration. Paradigm models were induced using corpora of 50,000 unique word types.

of Figure 6.2 gives the word-to-morpheme segmentation performance that ParaMor attains in the absence of both the clustering and all filtering procedures. In this search-only configuration ParaMor's scheme search algorithm identifies paradigm models that have high recall and low precision: recall near 80% with precision in the mid-teens. That ParaMor's initially selected schemes trade high morpheme boundary recall for low precision is not unexpected. In beginning separate search paths from all individual c-suffixes, a bias toward high recall, at the expense of precision, was intentionally built into ParaMor's search procedure (See Section 3.2).

Now consider the effect that scheme clustering has on ParaMor's performance at morphological segmentation. ParaMor's clustering algorithm was designed to group together all c-suffixes that belong to the same paradigm. Particularly relevant for this word segmentation experiment is the fact that clustering can join pairs of c-suffixes into a single paradigm that did not co-occur in any individual scheme. Recall that ParaMor segments words exactly when a pair of c-suffixes that co-occur in a scheme-cluster are mu-

tually interchangeable on some c-stem from a corpus. Therefore, increasing the number of c-suffixes which co-occur in paradigm models can increase the number of morpheme boundaries that ParaMor places. More frequent segmentation may translate into higher recall but may also lower precision. With ParaMor's segmentation recall already quite high, it is questionable how much clustering could improve recall. Empirically, in the absence of filtering, ParaMor's clustering algorithm has a negligible effect on both precision and recall. Apparently, most common pairs of English and German c-suffixes already co-occur in some scheme.

In contrast to ParaMor's clustering step, scheme filtering has a significant effect on ParaMor's morpheme segmentation performance. The paradigm filtration steps were designed to increase the initially low precision of ParaMor's selected schemes (See Section 4.3.3). ParaMor's filtering algorithms reduce the number of morpheme boundaries that ParaMor hypothesizes; And with such low initial precision, as long as a reasonable majority of the morpheme boundaries that ParaMor drops are the incorrect ones, precision will increase. ParaMor's filtering algorithms will decrease the number of morpheme boundaries that ParaMor proposes in two ways. First, individual erroneous c-suffixes can be entirely eliminated from ParaMor's paradigm models when all schemes or scheme-clusters that contain a particular c-suffix are removed—if a c-suffix doesn't exit, then it can't match against the tail of a word. Second, just as clustering can join pairs of c-suffixes into a paradigmatic relationship, deleting a scheme or a scheme-cluster can remove the hypothesis that a particular pair of c-suffixes are paradigmatically related. And lacking paradigmatic evidence, ParaMor will be unable to propose certain morpheme boundaries. Of course, ParaMor's filtration algorithms are imperfect, so it is likely that some correct schemes and some correct c-suffixes will be discarded—decreasing recall.

The 1st and 3rd rows of Figure 6.2 contain the configurations of ParaMor's paradigm induction steps that contrast in paradigm filtering in the absence of scheme clustering. In both English and German, precision rises significantly when scheme filtering is added: from 14.2% to 63.0% in English, and from 13.1% to 48.4% in German: These are improvements of 48.8% and 35.3% absolute for English and German respectively. Unfortunately, among the paradigm models that are discarded during the scheme filtering steps

are a significant number of models that propose correct morpheme boundaries: recall declines from 82.8% to 47.8% in English and from 78.6% to 41.6% in German, absolute percentage falls of 35.0% and 37.0% respectively. On balance, however, after filtering, the harmonic mean of precision and recall immensely improves for both languages: F_1 approximately doubles in each case to 54.3% for English and to 44.7% for German.

Now compare the final two rows of Figure 6.2. Both of these experimental configurations invoke both the search and the filtering routines, but the next-to-last row has no clustering while the final row does merge schemes. In English, as was the case when moving from the non-clustering to the clustering configuration without filtering (top two rows), precision falls and recall rises—although the magnitudes of the changes in recall and particularly precision are significantly greater in the presence of ParaMor's filtering routines.

However, while in English precision falls and recall rises when introducing ParaMor's scheme-clustering algorithm in the presence of the filtering routines, interactions between ParaMor's clustering algorithm and filtering algorithm make it possible for recall to fall and precision to rise, as happens in this corpus of German. During clustering, a scheme, *C*, which alone is not removed by ParaMor's morpheme boundary error filters, can form a cluster with schemes that ParaMor does believe mark morpheme boundaries. When ParaMor's morpheme boundary error filters flag 50% or more of the schemes in a cluster as hypothesizing a morpheme boundary, the cluster is removed—consequently, in the presence of ParaMor's scheme clustering, the *C* scheme will now be filtered from ParaMor's set of paradigm models. If *C* was the basis of hypothesized morpheme boundaries, then these boundaries will not be proposed when ParaMor's filtering algorithms are run *after* clustering. Hence, recall can drop and precision rise.

In German, the scheme **Ø.n** fills the role of the *C* scheme in the previous paragraph. A word-final **n** can mark a variety of morphosyntactic features in German including *Plural* ***Number*** and/or *Dative* ***Case*** on nouns. The stems of many, but not all, German words which inflect with a final **n** end in **e**, c.f. **Auge** '*eye* (*Singular*)' becomes **Augen** '*eye* (*Plural*)', but **Fenster** '*window* (*Singular Nominative*)' can inflect to **Fenstern** '*window* (*Plural Dative*)'. Because the c-stems of the **Ø.n** scheme end in a sufficient variety of

characters, the **Ø.n** scheme is not flagged as modeling an incorrect morpheme boundary by ParaMor's suffix-internal morpheme boundary error filter. However, during ParaMor's scheme-clustering phase, the **Ø.n** scheme is merged with several more-specialized schemes whose sets of c-stems end exclusively in the character **e**. ParaMor flags these now-merged **e**-schemes as incorrectly hypothesizing morpheme boundaries that lie internal to true suffixes, and the **Ø.n** scheme is consequently removed.

Significantly for German, the **Ø.n** scheme introduces some 133,891 morpheme boundaries into the 1.26 million unique German words that ParaMor segments! More than 10% of German words end in **n** and alternate with a surface form that lacks the **n**. Thus, the choice to include or discard this single **Ø.n** scheme from among ParaMor's paradigm models has a significant impact on the precision and recall scores that ParaMor receives for morphological segmentation. Comparing the two experimental configurations on the bottom two rows of Figure 6.2: when ParaMor filters but does not cluster schemes (thus retaining the **Ø.n** scheme) recall lies at 41.6% and precision at 48.4%; but when ParaMor both clusters schemes and filters the resulting clusters (in the process losing the **Ø.n** scheme), recall of course falls to 38.9% (as ParaMor is unable to correctly analyze words like **Augen**) but precision also rises considerably, to 54.1%.

ParaMor's precision significantly increases when the **Ø.n** scheme is removed because in many German words a final **n** does not constitute a morpheme. One common error caused by the **Ø.n** scheme occurs in German verbs. Consider three inflected forms of one particular verb: **spielen** '*play* (*Infinitive*)', **spiele** '*play* (*1st Person Present Indicative*)', and **spielt** '*play* (*3rd Person Past Indicative*)'. The correct morphological analysis of the verb **spielen** treats the string **spiel** as the verb stem and segments off the infinitive morpheme **en**, i.e. **spiel +en**. However, the existence of the form **spiele** incorrectly allows the **Ø.n** scheme to segment **spilen** as **spiele +n**. Overall, the significant increase in ParaMor's precision leads to a slight increase in F_1 for German when schemes are clustered in addition to being filtered.

While scheme clustering increases F_1 in German, in English, ParaMor's clustering algorithm actually lowers F_1 in the presence of scheme filtering, from 54.3% down to 52.7%. The scheme-clustering procedure manages to raise English recall only slightly,

from 47.8% to 48.8%, while significantly lowering precision through the introduction of many incorrect overgeneralized morpheme boundaries. Precision falls from 63.0% to 57.2%

As the ultimate effect of ParaMor's scheme-clustering algorithm on F_1 can differ from language to language, it is difficult to discern whether scheme clustering is fully appropriate for the word segmentation task. However, Section 4.3.3 demonstrated that scheme clustering significantly reduces the number of separate partial paradigm models that ParaMor produces. Because of this clear reduction in paradigm fragmentation, ParaMor's word segmentation submissions to the Morpho Challenge competition (Section 6.4) are produced when including the scheme-clustering algorithm in ParaMor's paradigm induction pipeline.

Limiting the Vocabulary

As a second look at the individual contributions that ParaMor's major sub-algorithms make toward morphological segmentation, consider ParaMor's performance when inducing paradigm models from a much smaller corpus. Where all other word-to-morpheme segmentation experiments in this thesis run ParaMor's paradigm induction algorithms over corpora of 50,000 unique word types, the results in Figure 6.3 were obtained from a corpus of 20,000 types. Like Figure 6.2, Figure 6.3 reports precision, recall, and F_1 scores (and the value of one standard deviation in parentheses) for morphological segmentations of English and German for four configurations of ParaMor's scheme-search, clustering, and filtering algorithms. The smaller corpus size of 20,000 unique types was chosen because the experiments in Section 4.5.2 demonstrated that in Spanish the quality of ParaMor's induced paradigm models begins to quickly degrade below a vocabulary size of 20,000.

Before turning to the experimental results, it is important to keep in mind that both this experiment over corpora of 20,000 types and that of Figure 6.2 over 50,000 types, as well as all other word segmentation experiments reported in this thesis, segment words over corpora that are much larger than the corpora from which paradigms are initially learned. The large size of the segmentation corpus is important because the more unique

			English			German		
Search	**Cluster**	**Filter**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
•			18.1 (±0.6)	75.1 (±1.3)	29.1 (±0.7)	16.2 (±0.5)	72.7 (±1.1)	26.5 (±0.6)
•	•		17.9 (±0.6)	75.5 (±1.3)	28.9 (±0.7)	16.2 (±0.5)	73.0 (±1.1)	26.5 (±0.7)
•		•	**58.0** (±1.0)	48.4 (±1.2)	52.8 (±0.8)	**55.4** (±1.0)	36.0 (±1.0)	43.6 (±0.7)
•	•	•	57.6 (±1.1)	**50.6** (±1.2)	**53.8** (±0.8)	44.0 (±0.9)	**45.4** (±1.0)	**44.7** (±0.6)

Figure 6.3: An ablation study when inducing paradigm models over 20,000 unique word types. ParaMor's **P**recision, **R**ecall, and **F_1** scores (and their standard deviations in parentheses) at word-to-morpheme segmentation using four configurations of ParaMor's scheme search, clustering, and filtering algorithms. (See also Figure 6.2).

word types that are present in the segmentation corpus the more likely that a particular lexeme will occur in more than one inflected form, providing the paradigmatic evidence that ParaMor requires to propose a morpheme boundary. By holding the segmentation corpus fixed, this experiment evaluates the quality of ParaMor's induced paradigms, as applied to the word-segmentation task, when the size of the paradigm induction vocabulary is reduced.

Overall, ParaMor's word segmentation algorithm is remarkably resilient to a reduction in the size of the paradigm induction corpus. In particular, the F_1 scores for the full ParaMor algorithm of search, clustering, and filtering at a vocabulary of 20,000 types (Figure 6.3) are within two standard deviations of the F_1 scores for paradigms trained over 50,000 types (Figure 6.2). In German, F_1 is slightly lower at 44.7% vs. 45.2%; And in English, F_1 is actually higher when paradigms are learned from the smaller vocabulary of 20,000 types, 53.8% vs. 52.7%.

This increase in English F_1 at word segmentation reflects small increases in both precision (from 57.2% to 57.6%) and in recall (from 48.8% to 50.6%) when learning paradigms from a smaller corpus. At first blush, increases in recall, let alone precision and F_1,

are counterintuitive for paradigm induction over a smaller data set. It would seem that ParaMor should find fewer total c-suffixes from a smaller corpus, and that this smaller set of c-suffixes should propose fewer morpheme boundaries. If this smaller set of discovered c-suffixes is more focused, then precision might increase, but surely recall should go down. Indeed, a consistent pattern of lower recall scores at smaller vocabulary sizes does occur among the experimental configurations that do not include ParaMor's filtering algorithms: Recall is at least 5% absolute higher for both English and German in the top two rows of Figure 6.2 than in the top two rows of Figure 6.3. However, in three of the four experiments that include ParaMor's filtering algorithms (the bottom two rows of Figures 6.2 and 6.3), the morpheme recall is higher when vocabulary size is smaller.

These recall statistics suggest that ParaMor's filtering algorithms are behaving in a less aggressive fashion at lower vocabulary sizes. As discussed in Section 4.5.2, ParaMor has one parameter that is not invariant to vocabulary size. The filtering algorithm that discards scheme-clusters which are not licensed by a sufficient number of word forms (Section 4.4.1) employs a threshold that must be adjusted with the size of the vocabulary. To compensate for the change in vocabulary size, the experiments in this section set the cluster-size threshold using the same procedure as the experiments in Section 4.5.2: the cluster-size threshold is linearly scaled with the vocabulary. In the Spanish experiments of Section 4.5.2, the linear adjustment of the cluster-size threshold was sufficient to ensure that the number of unique correct c-suffixes discovered by ParaMor should decrease with vocabulary size. But empirically, a linear scaling of the cluster-size filtering threshold does not prevent the inventory of discovered c-suffixes in English and German from increasing at lower vocabulary sizes. And hence, recall increases.

Now look beyond the morphology segmentation scores that the full ParaMor algorithm achieves in the small-vocabulary scenario of Figure 6.3 to the experimental configurations that omit the scheme clustering or filtering steps, or both. The segmentation scores of these other experimental configurations reveal a pattern broadly similar to that of the larger vocabulary experiments from Figure 6.2: When filtering is omitted, ParaMor's segmentation recall is high, precision low, and clustering has little effect. And again, as when learning from the larger vocabulary, invoking ParaMor's filtering algo-

rithms brings recall and precision into closer balance. At the smaller vocabulary size of 20,000 words types, ParaMor attains a maximum F_1 score, for both English and German, when ParaMor's full suite of search, clustering, and filtering algorithms is applied.

6.3 Inflectional vs. Derivational Morphology

As mentioned in Section 6.1.1 the Linguistic Evaluation of Morpho Challenge explicitly requires analyzing both inflectional and derivational morphology. But ParaMor is designed to discover paradigms—the organizational structure of *inflectional* morphology. The experiment of Figure 6.4 makes concrete ParaMor's relative strength at identifying inflectional morphology and relative weakness at analyzing derivational morphology.

Figure 6.4 contains Morpho Challenge style linguistic evaluations of English and German. For English and German, the official answer keys used in the 2007 and 2008 Morpho Challenge competitions were created from the widely available Celex morphological database (Burnage, 1990). To create the official Morpho Challenge answer keys, the Morpho Challenge organization extracted from Celex both the inflectional and the derivational structure of word forms. For the experiment in Figure 6.4, I constructed from Celex two Morpho Challenge style answer keys for English and two for German. First, because the Morpho Challenge organization did not release their official answer key, I built an answer key for each language very similar to the official Morpho Challenge answer keys where each word form is analyzed for both inflectional and derivational morphology. Second, I constructed from Celex answer keys for both English and Germen which contain analyses of only inflectional morphology.

From the 50,000 most frequent types in the Morpho Challenge English and German data that pass ParaMor's word-length restriction (see Section 4.2), ParaMor built scheme-cluster models of paradigms. And then Figure 6.4 evaluates ParaMor's morphological segmentations against both the answer key which contains only inflectional morphology and against the answer key which contains inflectional and derivational morphology. As described in Section 6.2, a minor modification to the Morpho Challenge scoring script

allowed the calculation of standard deviations. The standard deviation of F_1 is reported in the σ column of Figure 6.4.

Figure 6.4 reveals that ParaMor attains remarkably high recall of inflectional morphemes for both English, at 70.1%, and for German, at 66.5%. When evaluated against analyses which include both inflectional and derivational morphemes, ParaMor's morpheme recall scores are 20 to 30 absolute percentage points lower, English: 48.8% and German: 38.9%.

In addition to evaluating ParaMor's segmentations, Figure 6.4 evaluates segmentations produced by a morphology analysis system called Morfessor Categories-MAP v0.9.2 (Creutz, 2006). Morfessor is a state-of-the-art minimally supervised morphology induction algorithm that has no bias toward identifying inflectional morphology. To obtain Morfessor's segmentations of the English and German Morpho Challenge data used in this experiment, I downloaded the freely available Morfessor program and ran Morfessor over the data myself. Morfessor has a single free parameter. To make for stiff competition, Figure 6.4 reports results for Morfessor at that parameter setting which maximized F_1 in each separate evaluation scenario.

	Inflectional Only								**Inflectional & Derivational**							
	English				**German**				**English**				**German**			
	P	**R**	**F_1**	**σ**	**P**	**R**	**F_1**	**σ**	**P**	**R**	**F_1**	**σ**	**P**	**R**	**F_1**	**σ**
ParaMor	40.2	**70.1**	**51.0**	0.9	37.6	**66.5**	**48.0**	0.8	57.2	48.8	**52.7**	0.9	54.1	38.9	45.2	0.6
Morfessor	53.3	47.0	49.9	1.3	38.7	44.2	41.2	0.8	**73.6**	34.0	46.5	1.1	**66.9**	37.1	**47.7**	0.7

Figure 6.4: ParaMor segmentations compared to Morfessor's (Creutz, 2006) evaluated for **P**recision, **R**ecall, **F_1**, and standard deviation of F_1, **σ**, in four scenarios. Segmentations over English and German are each evaluated against correct morphological analyses consisting, on the left, of inflectional morphology only, and on the right, of both inflectional and derivational morphology.

Morfessor's unsupervised morphology induction algorithms, described briefly in Chapter 2, are quite different from ParaMor's. While ParaMor focuses on identifying productive paradigms of usually inflectional suffixes, Morfessor is designed to identify agglutinative sequences of morphemes. Looking at Figure 6.4, Morfessor's strength emerges to be the accurate identification of morphemes: both inflectional and derivational. In English and in German, Morfessor's precision against the answer key that contains both inflectional and derivational morphology is significantly higher than ParaMor's. And, as compared with ParaMor, a significant portion of the morphemes that Morfessor identifies are derivational. Morfessor's relative strength at identifying derivational morphemes is particularly clear in German. Against the German answer key of inflectional and derivational morphology, Morfessor's precision is higher than ParaMor's; but ParaMor has a precision comparable to Morfessor's when identifying just inflectional morphemes—indicating that many of the morphemes Morfessor correctly identifies are derivational. Similarly, while both ParaMor and Morfessor score lower at recall when required to identify derivational morphology in addition to inflectional; Morfessor's recall falls much less than ParaMor's—indicating that many of Morfessor's suggested segmentations which were dragging down precision against the inflection-only answer key were actually modeling valid derivational morphemes.

Clearly, the ParaMor and Morfessor morphology induction systems focus on very different areas of morphology. These two systems' complementary nature suggests pooling their morphological segmentations—a suggestion that will be realized in Section 6.4.1.

6.4 Morpho Challenge 2007/2008

Where the ablation study of Section 6.2 clarified the contributions of the subcomponents of the ParaMor algorithm toward morphological segmentation, and Section 6.3 examined ParaMor's relative performance at learning inflectional and derivational morphological structure, this section focuses on the full ParaMor algorithm under the strict requirements of the Morpho Challenge competitions held in 2007 and 2008. In

these two years of the Morpho Challenge, systems competed in a Linguistic Evaluation of up to five languages: English, German, Finnish, Turkish, and Arabic; and in a Task-Based Information Retrieval (IR) Evaluation in up to three languages: English, German, and Finnish. The scores from both the Linguistic and Task-Based Evaluations of the 2008 competition are directly comparable to scores from the 2007 Challenge because:

1. Both the Linguistic and Task-Based IR Evaluations used the same methodology in the 2007 and 2008 competitions; and moreover,
2. The more recent 2008 challenge scored systems over the same corpora and against the same answer keys as the 2007 competition.

While comparing systems from the 2007 and the 2008 Morpho Challenge competitions is reasonable, comparing the scores of even the same system across different languages is meaningless. Each language has radically different morphological structure and the Linguistic Evaluation uses different morphologically annotated answer keys for each language.

An early prototype of the ParaMor morphology induction algorithm competed in Morpho Challenge 2007 (Monson et al., 2008a). And the fully developed ParaMor algorithm participated in the 2008 Challenge (Monson et al., 2008b). Only ParaMor's improved performance from the 2008 Challenge is reported for ParaMor in this thesis. In the Morpho Challenge held in 2008, ParaMor took part in all five language tracks of the Linguistic Evaluation, and in all three language tracks of the Task-Based IR Evaluation. The next two sub-sections detail ParaMor's performance in the Linguistic and Task-Based Evaluations respectively.

6.4.1 Linguistic Evaluation Results from Morpho Challenge 2007/2008

Figure 6.5 summarizes ParaMor's performance in the Linguistic Evaluation of Morpho Challenge 2008 and places ParaMor's results into the context of the best performing systems from the 2007 and 2008 Challenges. Figure 6.5 contains the precision (P), recall (R), and F_1 scores of nine individual unsupervised morphology induction algorithms for

the five languages of the Linguistic Evaluation. Six of the nine systems in Figure 6.5 competed in Morpho Challenge 2008, while three systems participated in the 2007 challenge. Of the six systems from Figure 6.5 which competed in the 2008 challenge, three are systems that Monson et al. (2008b) submitted, while three are systems built and submitted by others. The three systems which Monson et al. (2008b) entered in Morpho Challenge 2008 are:

1. The ParaMor system alone, (2nd column of Figure 6.5),
2. An instance of the unsupervised morphology induction system Morfessor (Creutz, 2006) which I trained myself (3rd column), and
3. A joint ParaMor-Morfessor system which combines the analyses of 1. and 2., (1st column of Figure 6.5).

The joint ParaMor-Morfessor system was developed to leverage the complementary strengths of the ParaMor and Morfessor systems. As uncovered in Section 6.3: ParaMor excels at identifying inflectional morphology, while the Morfessor system discovers the most regular and frequent morphological structures of a language, whether inflectional or derivational. And as discussed in Section 6.1.1, the Morpho Challenge competition permits a system to submit more than one (ostensibly ambiguous) analysis of a single word. The ParaMor-Morfessor system joins analyses from ParaMor and Morfessor by simply adding Morfessor's segmentation of each word to ParaMor's segmentation as a separate, ambiguous, analysis.

The ParaMor algorithm has several free parameters that control the paradigm discovery phase. These parameters were set to values that produced reasonable Spanish paradigms. The parameters were then frozen before learning the morphology of the languages in the Morpho Challenge. In the experiments which adjoin ParaMor and Morfessor analyses, Morfessor's single free parameter was optimized for F_1 separately for English, German, and Turkish. To optimize Morfessor's parameter for these three languages, morphological answer keys were constructed from pre-existing morphological data and tools. The source for the English and German morphological answer keys was the Celex data-

		Monson et al. (2008b)			Other Authors					
		2008			2008			2007		
		ParaMor + Morfessor	ParaMor	Morfessor	Morfessor MAP	Zeman	Kohonen	Bernhard	Bordag	Pitler
English	P	50.6	58.5	77.2	82.2	53.0	83.4	61.6	59.7	74.7
	R	63.3	48.1	34.0	33.1	42.1	13.4	60.0	32.1	40.6
	F_1	56.3	52.8	47.2	47.2	46.9	23.1	**60.8**	41.8	52.6
German	P	49.5	53.4	67.2	67.6	53.1	87.9	49.1	60.5	-
	R	59.5	38.2	36.8	36.9	28.4	7.4	57.4	41.6	-
	F_1	**54.1**	44.5	47.6	47.8	37.0	13.7	52.9	49.3	-
Finnish	P	49.8	46.4	77.4	76.8	58.5	92.6	59.7	71.3	-
	R	47.3	34.4	21.5	27.5	20.5	6.9	40.4	24.4	-
	F_1	**48.5**	39.5	33.7	40.6	30.3	12.8	**48.2**	36.4	-
Turkish	P	51.9	56.7	73.9	76.4	65.8	93.3	73.7	81.3	-
	R	52.1	39.4	26.1	24.5	18.8	6.2	14.8	17.6	-
	F_1	**52.0**	46.5	38.5	37.1	29.2	11.5	24.7	28.9	-
Arabic	P	79.8	78.6	90.4	90.2	77.2	-	-	-	-
	R	27.5	8.5	21.0	21.0	12.7	-	-	-	-
	F_1	**40.9**	15.4	34.0	34.0	21.9	-	-	-	-

Figure 6.5: Results from the Linguistic Evaluation of Morpho Challenge. The unsupervised morphology induction systems which appear in this table are the nine best-performing systems from the 2008 and 2007 challenges. Systems participated in up to 5 language tracks. In each language track all participating systems were scored at **P**recision, **R**ecall, and $\mathbf{F_1}$ of morpheme identification. The ground truth against which Morpho Challenge compares systems in the Linguistic Evaluation is a morphologically analyzed answer key that includes both inflectional and derivational morphology. For each language track, the system or systems which place first at F_1 by a statistically significant margin appear in **bold**.

base (Burnage, 1990); while in Turkish, a hand-built morphological analyzer provided by Kemal Oflazer was used as the basis of a morphological answer key. Having limited access to morphologically annotated Finnish and Arabic data, Morfessor's parameter was not directly optimized for these languages. Instead, as Finnish and Arabic both have rich morphological systems, Morfessor's segmentations for these two languages were generated using the parameter value which performed best on the morphologically complex Turkish language.

The six systems from Figure 6.5 that were prepared by groups other than Monson et al. (2008b) are those systems with the top competing performances in the Linguistic Evaluation of the Morpho Challenge competitions from 2007 and 2008. The system labeled Morfessor MAP is a second instance of the same Morfessor algorithm that I trained, submitted, and joined with ParaMor. But the Morfessor MAP submission was prepared by Kurimo and Varjokallio (2008) and likely uses a different parameter setting for each language from the settings used in my Morfessor submissions. A change in parameter setting can sometimes result in quite different performance for Morfessor, c.f. Finnish. The remaining five systems, found in the right-most columns of Figure 6.5, bear the names of their principle authors. If multiple versions of a single algorithm competed in the 2007 and/or 2008 Morpho Challenge, then the scores reported in Figure 6.5 are from the algorithm variant which attained the highest F_1 score.

Because the number of morpheme-sharing word pairs that were used to calculate the precision and recall scores in the Linguistic Evaluation of Morpho Challenge was quite large, most score differences between systems in Figure 6.5 are statistically significant. All F_1 differences of more than 0.5 between systems which competed in Morpho Challenge 2007 are statistically significant (Kurimo, Creutz, and Varjokallio, 2008); and similarly, all F_1 differences of more than 0.5 among the Morpho Challenge 2008 systems are also statistically significant (Kurimo and Varjokallio, 2008).

To begin the examination of the results of the Linguistic Evaluation of Morpho Challenge, turn to the 2nd column of Figure 6.5 and examine the precision, recall, and F_1 scores that the ParaMor algorithm achieves alone. In all language tracks but Arabic, ParaMor holds its own against the state-of-the-art unsupervised morphology induction sys-

tems which competed in Morpho Challenge. Against the six best-performing systems from the 2007 and 2008 Morpho Challenge competitions that were not prepared by Monson et al. (2008b), and looking at F_1, the ParaMor algorithm places first in Turkish with 46.5%, second in English with 52.8%, third in Finnish at 39.5%, and fourth in German with 44.5%.

Although ParaMor's recall scores are consistently lower than precision for each language (in English precision is 58.5% vs. a recall of 48.1%, in German precision is 53.4% and recall 38.2%, etc.), the precision and recall scores of the lone ParaMor algorithm are often more balanced than the scores of other systems. ParaMor's balance in precision and recall is particularly noticeable in Turkish, where the morpheme recall of other unsupervised systems is anomalously low. In Turkish, the spread between ParaMor's precision, 56.7%, and recall, 39.4%, is 17.3 percentage points. In contrast, the smallest spread in any competing system is 47.0 percentage points in the Zeman (2008) system. ParaMor's focus on a recall-centric morphology induction procedure (see Chapter 3) and on a segmentation procedure that can propose more than one morpheme boundary in a single word (see Chapter 5) pays strong dividends when analyzing the highly agglutinative Turkish language.

Although ParaMor alone performs respectably, it is when ParaMor's analyses are adjoined with Morfessor's that ParaMor shines. In German, Finnish, Turkish, and Arabic, the combined ParaMor-Morfessor system achieves the highest F_1 of any system which competed in the 2007 or 2008 Challenges. And in English, the joint ParaMor-Morfessor system places a strong second. It is the precision score of the joint ParaMor-Morfessor system that drags the English F_1 under that of the first place system, Bernhard (2008). In Finnish, the Bernhard system's F_1 is likely not statistically different from that of the ParaMor-Morfessor system.

The joint ParaMor-Morfessor system attains its higher F_1 scores by balancing precision and recall. The trend for precision to be above recall that was noted for ParaMor is even more pronounced in the Morfessor system: Morfessor's lowest precision and highest recall scores both occur in German: 67.2% precision and 47.6% recall. Because of the

elevated precision scores in both ParaMor and Morfessor, a boost in the harmonic mean of precision and recall can be achieved by sacrificing precision for a decent gain in recall.

This tradeoff of precision for recall is exactly accomplished by adjoining the morphological segmentations that are proposed by the ParaMor and Morfessor systems. Combining the analyses of any two systems will increase the total number of morphemes in the analysis of each word—likely lowering precision but possibly increasing recall. But Section 6.3 suggests that the morphological analyses that the ParaMor and Morfessor systems propose are particularly suited for joining: the mostly inflectional morphemes that ParaMor identifies differ substantially from the mix of inflectional and derivational morphemes that Morfessor induces. Hence, although combining ParaMor's analyses with those from Morfessor almost always hurts precision, in all five languages, the improvement in recall significantly boosts F_1 over that of either ParaMor or Morfessor alone.

The performance of the ParaMor-Morfessor system is particularly striking in Turkish and in Arabic. In Turkish, the recall of the joint ParaMor-Morfessor system is double that of all non-ParaMor Turkish systems. This high recall leads to an improvement in F_1 over the next best system, Morfessor alone, of 13.5% absolute or 22.0% relative.

Contrasting with ParaMor's strong performance at Turkish, the language that the lone ParaMor algorithm performs most poorly at is Arabic. New to Morpho Challenge in 2008, Arabic's morphology is distinctly different from that of the other four languages in the challenge. Arabic morphology differs most notably in possessing templatic morphology, where a consonantal root is interleaved with vowels to produce specific surface forms. Equally important, from ParaMor's perspective, is that Arabic is the only language in Morpho Challenge with significant prefixation: Not only does Arabic verbal morphology include inflectional prefixes, but Arabic orthography also attaches a number of common determiners and prepositions directly onto the written form of the following word. These attached function words act as prepositions in text. With no strategy for identifying prefixes, let alone templatic morphology, the ParaMor algorithm recovers just 8.5% of the morphemes in the Arabic words. Such low recall pulls ParaMor's F_1 down to 15.4%. It is some small consolation that no lone unsupervised morphology induction system recovered even a quarter of the morphemes in the Arabic answer key.

Despite ParaMor's own poor performance at uncovering the morphological structure of Arabic, when ParaMor's analyses are presented in combination with Morfessor's the increase in recall between the two systems is practically additive. Morfessor's Arabic Recall is 21.0%, ParaMor's is 8.5%, and the recall of the joint ParaMor-Morfessor system is 27.5%. This significant jump in recall implies very little overlap between the morphemes which the ParaMor and Morfessor systems identify. When recall scores are depressed, an increase in recall implies an increase in F_1. And indeed, the ParaMor-Morfessor system receives the highest F_1 of any system which analyzed Arabic morphology.

In the near term, since prefixes are the mirror image of suffixes, a simple augmentation of ParaMor's algorithms could allow analysis of prefixation. The ability to identify prefixes would not only improve morpheme recall in Arabic, but help identify German verbal prefixes, and English derivational prefixes as well.

6.4.2 The Task-Based Evaluation of Morpho Challenge 2007/2008

The Task-Based Information Retrieval (IR) Evaluations of the 2007 and the 2008 Challenge covered three languages: English, German, and Finnish. To measure the impact that a morphology analysis system has on an IR system, the Morpho Challenge competition replaced all words in all documents and queries from each IR corpus with the morphological analyses the morphology system suggests for that language. In both years of the Morpho Challenge the same information retrieval corpus and query set were used, making results from 2007 comparable with results from 2008. The Morpho Challenge organizing committee did not measure the statistical significance of average precision scores in the IR Evaluation. Additional details on the procedure used in the Task-Based IR Evaluation of Morpho Challenge are given in Section 6.1.2 as well as in Kurimo and Turunen (2008) and Kurimo, Creutz, and Turunen (2007).

Figures 6.6 and 6.7 contain the results of the Task-Based IR Evaluations of Morpho Challenge 2007 and 2008. Figure 6.6 contains the average precision IR scores for the eight best performing systems from the 2007 and 2008 challenges; while Figure 6.7 contains average precision scores for four baseline metrics, namely:

1. **NO MORPHOLOGY:** IR experiments run over the raw documents and queries;
2. **SNOWBALL (PORTER):** All words in each document and query are stemmed using the Snowball package of language stemmers. In the case of English, the Snowball stemmer is the Porter stemmer;
3. **ANSWER KEY:** Document and query words are replaced with their morphological analyses from the answer keys that were used in the Linguistic Evaluation of Morpho Challenge. Note that the answer keys used in the Linguistic Evaluations contain only a subset of the full set of word types found in the IR corpora; and
4. **TWO-LEVEL:** Each word is replaced with the morphological analysis provided by a hand-built rule-based morphological analysis system. No hand-built morphological analysis system was evaluated for German.

In the IR Evaluation of Morpho Challenge, the ParaMor system alone placed third in English, while the combined ParaMor-Morfessor system placed first in English and German, and fourth in Finnish. The IR Evaluation is a black-box experiment, and so it is not completely clear why the ParaMor-Morfessor system fared worse in the Finnish track. The most likely explanation is that replacing each word in each document and query with *both* the ParaMor *and* the Morfessor analyses is inappropriate for a language with complex morphology such as Finnish. It is unfortunate that Morpho Challenge did not conduct IR experiments for the morphologically complex Turkish and Arabic languages. It would be particularly interesting to see ParaMor's IR performance on Turkish, which, like Finnish, is agglutinative.

The ParaMor systems also perform well in comparison to the baseline algorithms of Figure 6.7. Most notably, in all languages, both the lone ParaMor algorithm and the joint ParaMor-Morfessor system improve on the average precision scores the IR system achieves when no morphological analysis is performed. With no morphological analysis, the IR system scores 32.9% average precision for English, 35.1% for German and 35.2% for Finnish; while the joint ParaMor-Morfessor system scores 39.9% for English, 47.3% for German, and 46.7% for Finnish.

	Monson et al. (2008) 2008			Other Authors 2008			2007	
	ParaMor + Morfessor	ParaMor	Morfessor	Morfessor MAP	Morfessor Baseline	McNamee	Bernhard	Bordag
English	**39.9**	39.3	36.4	37.1	38.6	36.3	39.4	34.3
German	**47.3**	36.3	46.7	46.4	46.6	43.9	**47.3**	43.1
Finnish	46.7	39.7	46.8	44.4	44.3	**49.2**	**49.2**	43.1

Figure 6.6: Average precision scores for unsupervised morphology induction systems which participated in the Information Retrieval (IR) Evaluation of Morpho Challenge. The unsupervised morphology induction systems which appear in this table are the eight best systems from the 2008 and 2007 challenges. Systems participated in up to three language tracks. The best performing system(s) for each track appear in **bold** font.

	No Morphology	Snowball (Porter)	Answer Key	Two- Level
English	32.9	40.8	37.3	39.6
German	35.1	38.7	33.5	-
Finnish	35.2	42.8	43.1	49.8

Figure 6.7: Average precision scores of four reference algorithms for the Information Retrieval (IR) Evaluation of Morpho Challenge.

The best performing unsupervised systems, including the ParaMor-Morfessor system, also outperform the baseline **ANSWER KEY** scenario: demonstrating that imperfect morphological analysis performed by unsupervised morphology induction systems can trump perfect analysis of a subset of the words found in a task. ParaMor and the other unsupervised systems face stiffer competition in the two hand-built morphological baselines that have some generalization capacity, **SNOWBALL (PORTER)** and **TWO-LEVEL**. The Porter stemmer has the best average precision of any method against English; but unsupervised systems, the joint ParaMor-Morfessor system among them, outperform the Snowball rule-based stemmers for both German and Finnish. And finally, although the hand-built two-level morphological analyzer improves average precision more than any unsupervised induction method does in Finnish; the joint ParaMor-Morfessor system edges out the hand-built English morphological analysis system.

Chapter 7: Conclusions and Future Work

ParaMor, the unsupervised induction system developed for this thesis automatically discovers the morphological paradigms of natural language from unannotated text in a three step process. First, ParaMor lays out a space of candidate partial paradigms and applies a recall-centric search strategy to that space (Chapter 3). Second, ParaMor merges candidate paradigms that likely describe portions of the same true paradigm (Section 4.3). And third, ParaMor culls out the least likely candidates (Sections 4.2 4.2and 4.4). With a firm grasp on the paradigmatic structure of a particular language, ParaMor then segments individual words of that language exactly when there is paradigmatic evidence that a word adheres to a discovered paradigm (Chapter 5).

ParaMor's identified paradigmatic models organize inflectional morphology by grouping mutually substitutable suffixes into paradigm-like structures. With its focus on inflectional paradigms, ParaMor contrasts with other unsupervised morphology induction systems, such as Morfessor (Creutz, 2006), which seek to identify all morpheme types whether inflectional or derivational. With their emphasis on very different aspects of morphology, ParaMor's morphological analyses are largely complementary to those of a

system like Morfessor. And this thesis leverages the individual strengths of the general purpose morphology induction system Morfessor and the inflection specific system ParaMor by combining the analyses from the two systems into a single joint analysis.

The combined ParaMor-Morfessor system competed in the 2007 and 2008 Morpho Challenge competitions, which evaluated unsupervised morphology induction systems at morpheme identification. In four of the five language tracks of the 2007/2008 Morpho Challenge competitions, in German, Finnish, Turkish, and in Arabic, the ParaMor-Morfessor system achieved an F_1 score for morpheme identification at or above that of all other systems which competed. The primary reason for the combined ParaMor-Morfessor system's success at morpheme identification is its strong morpheme recall. In Turkish, the combined system's recall, at 52.1%, is twice that of the next highest system.

Results from the Morpho Challenge competitions also suggest that the ParaMor morphology analysis system is helpful in higher-level natural language processing tasks. Augmenting an Information Retrieval (IR) system with the morphological analyses that are proposed by the ParaMor system alone significantly improves average precision over a morphologically naïve baseline. ParaMor's improvement over the baseline IR scores occurs in all three languages of the Task-Based IR Evaluation at Morpho Challenge. Furthermore, the IR average precision scores of the joint ParaMor-Morfessor system place first among all competing systems in English and German.

Despite ParaMor's successes, this thesis, naturally, could not exhaustively investigate all areas of the ParaMor algorithm. Nor could this thesis explore every aspect of unsupervised morphology induction in general. The next two sections outline particular suggestions for extending ParaMor and specific recommendations to all who will pursue unsupervised morphology induction in the future.

7.1 Improving the Core ParaMor Algorithms

Developing ParaMor's large suite of algorithms necessitated prioritizing implementation. The foremost priority for this thesis was to create a complete system that could both identify paradigms and then segment word forms. Unfortunately, focusing on the full Pa-

raMor system required limiting the investigation of both the paradigm identification algorithm as well as the segmentation algorithm individually. Beginning with the paradigm identification algorithm and moving toward the segmentation algorithm, five specific pieces of ParaMor's current algorithms warrant further work.

First, although ParaMor's scheme-clustering algorithms of Section 4.3 significantly reduce the fragmentation of true paradigms across ParaMor's paradigm models, the final scheme-clusters remain disjointed. For example, among the 42 scheme-clusters that ParaMor outputs over a training corpus of 50,000 Spanish types, nine separate clusters model portions of the **ar** inflection class of Spanish verbs. But ParaMor was unable to merge these nine clusters because ParaMor's c-suffix based discriminative restriction on clustering, described in Section 4.3.1, prevents any cluster from containing a pair of c-suffixes that are not mutually substitutable on at least one c-stem. The verbal c-suffixes **ándoles** and **arme**, for example, which are each built of a non-finite verb suffix in combination with a pronominal clitic, did not both attach to any single c-stem in this Spanish corpus. Although both the **ándoles** and the **arme** c-suffixes belong to the **ar** inflection class, any one ParaMor cluster is prevented from containing them simultaneously. In failing to merge models which clearly describe portions of the same paradigm, ParaMor does not capture the full generality and power of paradigms.

In a similar vein, ParaMor's clustering restrictions prevent distinct inflection classes of the same paradigm from coalescing. For example, in Spanish ParaMor keeps separate those scheme-clusters that model the **ar** verbal inflection class from those clusters that model the **er** inflection class of verbs, which are separated again from models of the **ir** inflection class. On the one hand, it was an intentional choice to prevent ParaMor from merging distinct inflection classes of the same underlying paradigm: Although all verbs in Spanish inflect for the same morphosyntactic features, verbs which adhere to distinct inflection classes use distinct suffixes to mark the same feature sets, and sometimes even use the same surface suffix to mark distinct sets of features (see Chapter 4). While it is clearly important, then, to somehow distinguish between distinct inflection classes, ParaMor's separated models prevent the segmentations of the *3rd Person Singular Present*

Indicative Spanish word forms **bebe** *'she drinks'* and **habla** *'she speaks'*, as **beb + e** and **habl + a** respectively, from containing matching suffixes.

The second area of ParaMor that warrants further work is a scaling up of ParaMor's paradigm identification algorithms from the relatively small vocabulary size of 50,000 unique types to the orders of magnitude larger vocabularies used in the Morpho Challenge competitions. While ParaMor's segmentation algorithms can segment arbitrarily large corpora, in this thesis ParaMor included at most 50,000 types from any corpus in the paradigm induction vocabulary. There is no significant barrier to scaling up ParaMor's paradigm identification algorithms to larger vocabulary sizes. Each step in ParaMor's paradigm induction pipeline will scale with larger vocabularies: ParaMor's on-demand instantiation of the morphology scheme network allows the scheme search procedure to scale reasonably with vocabulary size both in time and space complexity. And bottom-up agglomerative scheme clustering is at worst cubic in time with the number of initial schemes. The primary reason this thesis did not scale up ParaMor's paradigm identification algorithms is that ParaMor's parameters were set over the smaller vocabulary size, and time did not permit empirical adjustment of the parameter settings to accommodate larger vocabularies.

In contrast to scaling up ParaMor's paradigm identification phase, the third ParaMor-specific question this thesis does not answer is how ParaMor's segmentation algorithms scale down to smaller vocabulary sizes. The larger the vocabulary that ParaMor segments, the more likely ParaMor will be to find evidence that a word belongs to a particular paradigm. The vocabularies of the language corpora in the Morpho Challenge 2007 and 2008 competitions were very large. But an unsupervised morphology induction system might be most useful for languages with limited machine readable data available. Thus, it will be important to scale ParaMor's segmentation algorithms down to smaller vocabulary sizes. In ParaMor's current morphological segmentation algorithm, a word-final string, *f*, of a particular word form, *w*, must be mutually substitutable with another c-suffix of a paradigm before ParaMor will place a morpheme boundary in *w* before *f*. Perhaps in scenarios where more limited data is available, ParaMor's morphological

segmentation algorithm should require less evidence that a stem adheres to a particular paradigm before segmenting.

A fourth area that this thesis did not fully investigate is the effect that ParaMor's four free parameters have on ParaMor's ultimate morphological segmentations. ParaMor's free parameters directly control the induced paradigms. ParaMor's first parameter is the cutoff in the morphology scheme network search on when a parent scheme is deemed likely to correctly extend the coverage of the current scheme (Chapter 3). The second parameter is the lower boundary on the string length of the word forms used in the paradigm induction corpus (Section 4.2). The third parameter decides when to discard a scheme-cluster based on the number of word types a scheme-cluster covers (Section 4.4.1). And the fourth parameter governs the suffix-internal and stem-internal morpheme boundary error filters (Section 4.4.2). ParaMor's four free parameters were set not by examining word-to-morpheme segmentations, but by examining the candidate paradigms, i.e. schemes and scheme-clusters, that ParaMor created.

The chronological development of ParaMor's core algorithms largely followed the order of ParaMor's processing pipeline. In particular, ParaMor's word-to-morpheme segmentation algorithm was only developed after ParaMor's paradigm identification algorithms were in place. Hence, no experiments directly measured changes in word segmentation quality as ParaMor's parameters varied. As described in Section 6.3, ParaMor's free parameters were intentionally set to maximize the recall of inflectional morphemes. Only empirical experimentation could determine if a more restrictive search parameter or more aggressive filtering of scheme-clusters would significantly improve ParaMor's precision at the word-to-morpheme segmentation task without severely hurting recall.

Finally, this thesis leaves to future work the development of a more satisfactory approach for combining the segmentations of ParaMor and Morfessor. The current algorithm does not attempt, for any particular word, to merge the morphological segmentations from the two systems. Instead, the procedure described in this thesis simply suggests each system's segmentation as an alternative analysis. A more sophisticated system could select from among the morpheme boundaries that are suggested by the two sys-

tems, and produce a single unified segmentation. In fact, a general purpose solution to the problem of combining segmentations that come from multiple morphology analysis algorithms could be used to combine segmentations produced by additional unsupervised morphology induction systems beyond ParaMor and Morfessor.

7.2 The Future of Unsupervised Morphology Induction

This section looks beyond specific extensions to the ParaMor algorithm to the broader picture of the steps future unsupervised morphology induction systems must take. The field of unsupervised morphology induction is still in its infancy. Although unsupervised morphology induction is a large and complex problem, implemented systems almost universally focus on a narrow slice of the most simple forms of morphology. Three areas where very little work has been done on morphology induction from an unsupervised perspective are:

1. Morphological processes other than suffixation,
2. Morphophonemics, and
3. Mapping from morphemes to morphosyntactic features.

The next three sub-sections examine these underserved areas in turn.

7.2.1 Beyond Suffixation

This thesis joins most other unsupervised morphology induction work in addressing the most prevalent morphological process, suffixation. While other morphological processes play an important role in many languages, suffixation plays a significant role in the morphology of nearly all the world's languages, including those that also employ other morphological processes (Dryer, 2008). Still, suffixation is only the most common morphological process among many, including: prefixation, infixation, reduplication, vowel

and consonant mutation, templatic morphology (as in Arabic), as well as suprasegmental changes such as tone or stress.

The next step in the field of unsupervised morphology induction will be to address these additional morphological processes. To date, the most promising work on unsupervised induction of non-concatenative morphology is Wicentowski (2002), which explicitly assumes an underlying morphology that is not purely concatenative. However, most work on unsupervised induction of morphology has focused on concatenative processes. Morfessor (Creutz, 2006), for example, agnostically discovers any concatenative morphological processes. Indeed, ParaMor could be easily extended to analyze the concatenative operation of prefixation by treating initial substrings of words as candidate prefixes in paradigmatic relationships and final substrings as the syntagmatic candidate stems.

7.2.2 Morphophonological Change

Not only does ParaMor restrict morphological analysis to suffixation, ParaMor also requires the modeled suffixation to be purely concatenative. But in fact, suffixation is often not strictly concatenative. Suffixation, like all morphological processes, can be accompanied by phonologic changes. Even English's limited morphology is rife with examples of morphophonology: many nouns voice a final fricative before the plural, i.e. *wolf* becomes *wolves* not **wolfs*. But more can change than just the word final consonant: both Finnish and Turkish, evaluated in the 2007 and 2008 Morpho Challenge competitions, have vowel harmony where vowels in a suffix change to match vowels in the stem.

Although Wicentowski (2002), again ahead of its time, and Goldwater and Johnson (2004) begin to approach the unsupervised learning of morphophonology, most current generation unsupervised morphology induction systems do not address morphophonology. In particular, neither ParaMor nor Morfessor (Creutz, 2006) model morphophonology in a systematic fashion. A logical next step for the ParaMor algorithm would be to extend the definition of a scheme from a strict concatenation of c-stems and c-suffixes to allow for phonological change when a stem and affix are joined.

7.2.3 Mapping Morphemes to Features

Finally, isolating the stem of a word by identifying and separating off inflectional suffixes, as ParaMor does, constitutes only the first step of full morphological analysis. A full analysis must also specify the set of morphosyntactic features marked by each identified affix. Probst (2003) is the only prior attempt I am aware of to automatically associate induced suffixes with morphosyntactic features such as ***Person***, ***Number***, or ***Tense***. Being able to map morphosyntactic features onto discovered morphemes would be particularly useful for syntactic transfer based machine translation, where unification rules over morphosyntactic features restrict the set of applicable translation rules.

Deriving a mapping from surface morphemes to morphosyntactic features requires knowledge of the grammatical features that exist in the specific language. Ongoing work in the Language Technologies Institute at Carnegie Mellon University (Clark, Frederking, and Levin, 2008) plans to acquire this knowledge of grammatical features through a process of feature detection. Figure 7.1 pictures the general process of feature detection, where a bilingual informant aligns the words in pairs of translated of sentences. In this example, a bilingual informant translates the two English sentences “The tree fell” and “The trees fell” into Spanish. The feature detection system then compares sets of features that are associated with the English sentences. In Figure 7.1, the feature detection system would find that the morphosyntactic feature structures associated with this pair of sentences are identical except for one feature value—the ***Subject Number***. To discover whether ***Subject Number*** is marked in Spanish, the feature detection system compares the Spanish words which are aligned to the head, to the dependent, and to the governor of the sentence subject. During these comparisons the feature detections system learns that ***Subject Number*** is marked in all three possible locations, i.e. on the head **árbol** vs. **árboles**, on the dependent, **El** vs. **Los**, and on the governor **cayó** vs. **cayeron**, because each of these surface string pairs differ. And now, with knowledge of what features are marked in Spanish, c-suffixes like **es** on **árbol<u>es</u>**, and **Ø** on **árbol(<u>Ø</u>)** can be associated with *Singular* and *Plural* ***Number***.

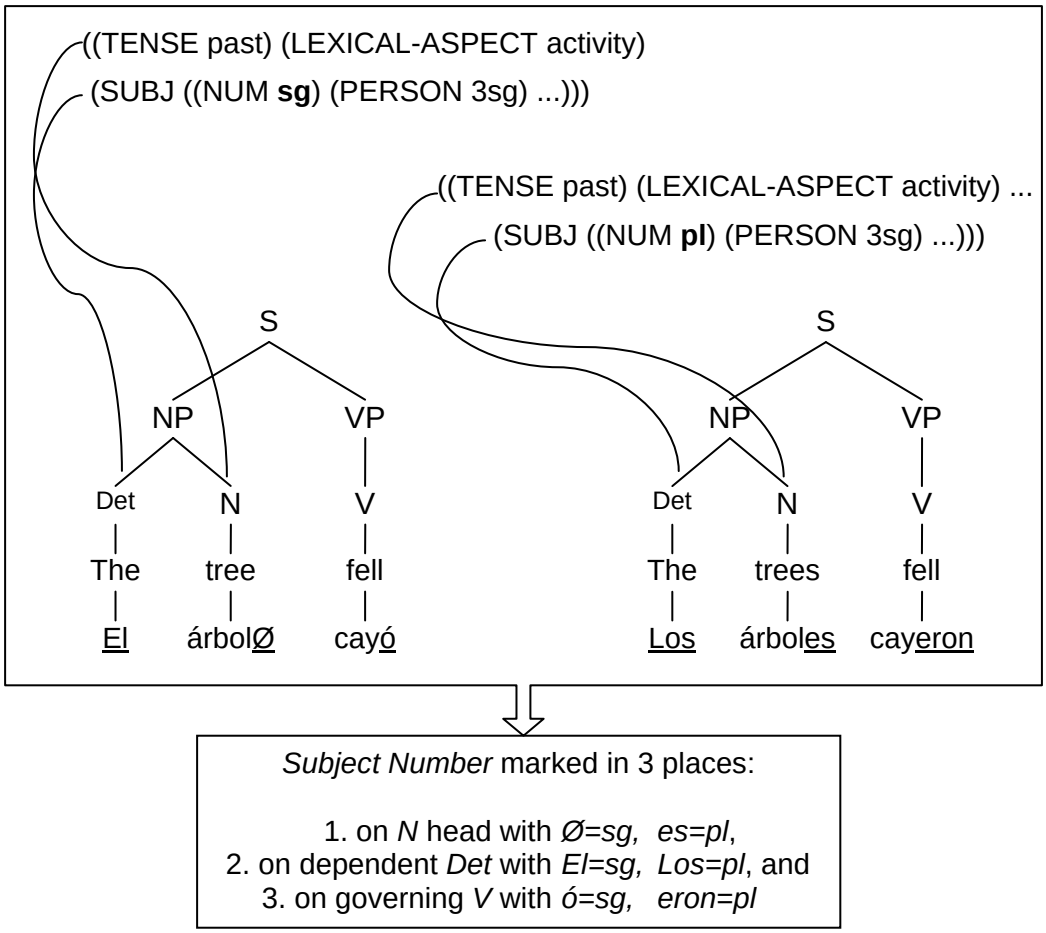

Figure 7.1: A General schema for feature detection from word-aligned translated sentences and associated feature structures.

7.3 ParaMor: A Successful Morphology Induction Algorithm

This thesis has empirically shown that the inherent paradigmatic organization of inflectional morphemes can be leveraged to induce the morphological structure of natural languages in an unsupervised fashion. With algorithms specifically tailored to the unique structure of natural language morphology, as the scheme-search and scheme-clustering procedures of Chapters 3 and 4 are, the ParaMor system is able to recover models of the paradigmatic relationships that exist between the inflectional morphemes of a language.

This thesis validated the quality of the discovered paradigmatic models both by direct comparison of the induced paradigm models against hand-crafted paradigm descriptions (Chapter 4) and by ParaMor's improvements on state-of-the-art performance in the Morpho Challenge competitions held in 2007 and 2008 (Chapter 6).

Bibliography

Altun, Yasemin, and Mark Johnson. 2001. Inducing SFA with ϵ-Transitions Using Minimum Description Length. *Finite State Methods in Natural Language Processing Workshop at ESSLLI*, Helsinki, Finland.

The American Heritage® Dictionary of the English Language. 2000. Fourth Edition, Houghton Mifflin Company.

Angluin, Dana. 1982. Inference of Reversible Languages. *Journal of the ACM*, 29.3, 741-765.

Baroni, Marco. 2000. *Distributional Cues in Morpheme Discovery: A Computational Model and Empirical Evidence*. Ph.D. Thesis in Linguistics, Los Angeles, University of California, Los Angeles.

Baroni, Marco, Johannes Matiasek, and Harald Trost. 2002. Unsupervised Discovery of Morphologically Related Words Based on Orthographic and Semantic Similarity. *ACL Special Interest Group in Computational Phonology in Cooperation with the ACL Special Interest Group in Natural Language Learning (SIGPHON/SIGNLL)*, Philadelphia, Pennsylvania, 48-57.

Baroni, Marco. 2003. Distribution-driven Morpheme Discovery: A Computational/Experimental Study. *Yearbook of Morphology*.

Beesley, Kenneth R., and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Palo Alto, California.

Bernhard, Delphine. 2008. Simple Morpheme Labelling in Unsupervised Morpheme Analysis. *Lecture Notes in Computer Science: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers*, Budapest, Hungary, Springer, 5152/2008, 873-880.

Biermann, A. W., and J. A. Feldman. 1972. On the Synthesis of Finite-State Machines from Samples of Their Behavior. *IEEE Transactions on Computers*, C-21.6, 592-597.

Bordag, Stefan. 2008. Unsupervised and Knowledge-Free Morpheme Segmentation and Analysis. *Lecture Notes in Computer Science: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers*, Budapest, Hungary, Springer, 5152/2008, 881-891.

Brent, Michael. 1993. Minimal Generative Explanations: A Middle Ground between Neurons and Triggers. *Cognitive Science Society*, University of Colorado-Boulder, Lawrence Erlbaum Associates, Inc., 28-36.

Brent, Michael R., Sreerama K. Murthy, and Andrew Lundberg. 1995. Discovering Morphemic Suffixes: A Case Study in MDL Induction. *The Fifth International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, Florida.

Burnage, Gavin. 1990. *Celex—A Guide for Users*. Springer, Centre for Lexical information, Nijmegen, the Netherlands.

Casella, George, and Roger L. Berger. 2002. *Statistical Inference*. Second Edition, Duxbury, Thomson Learning Inc.

Chan, Erwin. 2006. Learning Probabilistic Paradigms for Morphology in a Latent Class Model. *ACL Special Interest Group on Computational Phonology*, New York City, New York, 69-78.

Clark, Jonathan H., Robert Frederking, and Lori Levin. 2008. Inductive Detection of Language Features via Clustering Minimal Pairs: Toward Feature-Rich Grammars in Machine Translation. *Workshop on Syntax and Structure in Translation (SSST) at the Association for Computational Linguistics (ACL)*, Columbus, Ohio, 78-86.

Creutz, Mathias, and Krista Lagus. 2002. Unsupervised Discovery of Morphemes. *ACL Special Interest Group in Computational Phonology in cooperation with the ACL Special Interest Group in Natural Language Learning (SIGPHON/SIGNLL)*, Philadelphia, Pennsylvania, 21-30.

Creutz, Mathias. 2003. Unsupervised Segmentation of Words Using Prior Distributions of Morph Length and Frequency. *The Association for Computations Linguistics (ACL)*, Sapporo, Japan.

Creutz, Mathias, and Krista Lagus. 2004. Induction of a Simple Morphology for Highly-Inflecting Languages. *ACL Special Interest Group in Computational Phonology*, Barcelona, Spain, 43-51.

Creutz, Mathias. 2006. *Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition*. Ph.D. Thesis, Computer and Information Science, Report D13, Helsinki, University of Technology, Espoo, Finland.

DeGroot, Morris H. 1989. *Probability and Statistics*. Second Edition, Reading, Massachusetts, Addison-Wesley Publishing Company.

Déjean, Hervé. 1998. Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. *New Methods in Language Processing and Computational Natural Language Learning (NeMLaP/CoNLL) Workshop on Paradigms and Grounding in Language Learning*, Ed. Powers, David M. W., 295-298.

Demberg, Vera, Helmut Schmid, and Gregor Möhler. 2007. Phonological Constraints and Morphological Preprocessing for Grapheme-to-Phoneme Conversion. *Association for Computational Linguistics (ACL)*, Prague, Czech Republic.

Dryer, Matthew S. 2008. Prefixing vs. Suffixing in Inflectional Morphology. In: Haspelmath, Martin, David Gil, and Bernard Comrie (Eds.) *The World Atlas of Language Structures Online*. Munich: Max Planck Digital Library, Chapter 26. Available online at <http://wals.info/features/26>. Accessed on December 17, 2008.

Francis, W. Nelson. 1964. *A Standard Sample of Present-Day English for Use with Digital Computers*. Report to the U.S. Office of Education on Cooperative Research Project No. E-007, Brown University, Providence, Rhode Island.

Gaussier, Éric. 1999. Unsupervised Learning of Derivational Morphology from Inflectional Lexicons. *Unsupervised Learning in Natural Language Processing an ACL'99 Workshop*, University of Maryland.

Goldsmith, John. 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27.2, 153-198.

Goldsmith, John. 2006. An Algorithm for the Unsupervised Learning of Morphology. *Natural Language Engineering*. 12.4, 335-351.

Goldwater, Sharon, and Mark Johnson. 2004. Priors in Bayesian Learning of Phonological Rules. *ACL Special Interest Group in Computational Phonology*, Barcelona, Spain, 35-42.

Goldwater, Sharon, and David McClosky. 2005. Improving Statistical MT through Morphological Analysis. *Empirical Methods in Natural Language Processing*, Vancouver, Canada.

Gordon, Ronni L., and David M. Stillman. 1999. *The Ultimate Spanish Review and Practice*. Passport Books, Chicago, Illinois.

Hafer, Margaret A., and Stephen F. Weiss. 1974. Word Segmentation by Letter Successor Varieties. *Information Storage and Retrieval*, 10.11/12, 371-385.

Hammarström, Harald. 2006a. A Naive Theory of Affixation and an Algorithm for Extraction. *ACL Special Interest Group on Computational Phonology*, New York City, New York, 79-88.

Hammarström, Harald. 2006b. Poor Man's Stemming: Unsupervised Recognition of Same-Stem Words. *Information Retrieval Technology: Proceedings of the Third Asia Information Retrieval Symposium, AIRS 2006*, Singapore, 323-337.

Hammarström, Harald. 2007. *Unsupervised Learning of Morphology: Survey, Model, Algorithm and Experiments*. Thesis for the Degree of Licentiate of Engineering, Department of Computing Science, Chalmers University of Technology and Göteborg University, Göteborg, Sweden.

Harris, Zellig. 1955. From Phoneme to Morpheme. *Language*, 31.2, 190-222, Reprinted in Harris (1970).

Harris, Zellig. 1967. Morpheme Boundaries within Words: Report on a Computer Test. *Transformations and Discourse Analysis Papers*. Department of Linguistics, University of Pennsylvania, Reprinted in Harris (1970).

Harris, Zellig. 1970. *Papers in Structural and Transformational Linguists*. Ed. D. Reidel, Dordrecht.

Jacquemin, Christian. 1997. Guessing Morphology from Terms and Corpora. *SIGIR '97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Philadelphia, Pennsylvania, ACM Press, 156-165.

Johnson, Howard, and Joel Martin. 2003. Unsupervised Learning of Morphology for English and Inuktitut. *Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Edmonton, Canada.

Kurimo, Mikko, Mathias Creutz, and Ville Turunen. 2007. Unsupervised Morpheme Analysis Evaluation by IR Experiments – Morpho Challenge 2007. *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

Kurimo, Mikko, Mathias Creutz, and Matti Varjokallio. 2008. Morpho Challenge Evaluation Using a Linguistic Gold Standard. *Lecture Notes in Computer Science: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers*, Budapest, Hungary, Springer, 5152/2008, 864-872.

Kurimo, Mikko, and Matti Varjokallio. 2008. Unsupervised Morpheme Analysis Evaluation by a Comparison to a Linguistic Gold Standard – Morpho Challenge 2008. *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

Kurimo, Mikko, and Ville Turunen. 2008. Unsupervised Morpheme Analysis Evaluation by IR Experiments – Morpho Challenge 2008. *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

Kurimo, Mikko, Ville Turunen, and Matti Varjokallio. 2008. *Unsupervised Morpheme Analysis—Morpho Challenge 2008.* <http://www.cis.hut.fi/morphochallenge2008/>, Accessed on December 10, 2008.

Lang, Kevin J., Barak A. Pearlmutter, and Rodney A. Price. 1998. Results of the Abbadingo One DFA Learning Competition and New Evidence Driven State Merging Algorithm. *ICGI '98: The 4th International Colloquium on Grammatical Inference*, Springer-Verlag.

Manning, Christopher D., and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

McNamee, Paul. 2008. Retrieval Experiments at Morpho Challenge 2008. *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

Miclet, Laurent. 1980. Regular Inference with a Tail-Clustering Method. *IEEE Transactions on Systems, Man, and Cybernetics SMC*-10.11, 737-743.

Monson, Christian, Alon Lavie, Jaime Carbonell, and Lori Levin. 2004. Unsupervised Induction of Natural Language Morphology Inflection Classes. *ACL Special Interest Group in Computational Phonology (SIGPHON)*, Barcelona, Spain, 52-61.

Monson, Christian, Jaime Carbonell, Alon Lavie, and Lori Levin. 2007. ParaMor: Minimally Supervised Induction of Paradigm Structure and Morphological Analysis. *ACL Special Interest Group in Computational Morphology and Phonology (SIGMORPHON)*, Prague, Czech Republic, 117-125.

Monson, Christian, Jaime Carbonell, Alon Lavie, and Lori Levin. 2008a. ParaMor: Finding Paradigms across Morphology. *Lecture Notes in Computer Science: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Revised Selected Papers*, Budapest, Hungary, Springer, 5152/2008, 900-907.

Monson, Christian, Jaime Carbonell, Alon Lavie, and Lori Levin. 2008b. ParaMor and Morpho Challenge 2008. *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

Monson, Christian, Alon Lavie, Jaime Carbonell, and Lori Levin. 2008c. Evaluating an Agglutinative Segmentation Model for ParaMor. *ACL Special Interest Group in Computational Morphology and Phonology (SIGMORPHON)*, Columbus, Ohio, 49-58.

Oflazer, Kemal, and İlknur Durgar El-Kahlout. 2007. Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. *Statistical Machine Translation Workshop at ACL*, Prague, Czech Republic.

Ogilvie, Paul, and Jamie Callan. 2002. Experiments Using the Lemur Toolkit. *The 2001 Text Retrieval Conference (TREC 2001)*, 103-108, National Institute of Standards and Technology, Special Publication, 500-250.

Probst, Katharina. 2003. Using 'Smart' Bilingual Projection to Feature-Tag a Monolingual Dictionary. *Computational Natural Language Learning (CoNLL)*, Edmonton, Canada.

Roark, Brian, and Richard Sproat. 2007. *Computational Approaches to Morphology and Syntax*. Oxford University Press Inc., New York.

Robertson, S. E., S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1994. Okapi at TREC-3. *The Third Text Retrieval Conference (TREC-3)*.

Schone, Patrick, and Daniel Jurafsky. 2000. Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. *Computational Language Learning (CoNLL)*, Lisbon, Portugal, 67-72.

Schone, Patrick, and Daniel Jurafsky. 2001. Knowledge-Free Induction of Inflectional Morphologies. *North American Chapter of the Association for Computational Linguistics (NAACL)*, Pittsburgh, Pennsylvania, 183-191.

Snover, Matthew G., and Michael R. Brent. 2001. A Bayesian Model for Morpheme and Paradigm Identification. *Association for Computational Linguistics (ACL)*, Toulouse, France, Morgan Kaufmann.

Snover, Matthew G. 2002. *An Unsupervised Knowledge Free Algorithm for the Learning of Morphology in Natural Languages*. Sever Institute of Technology, Computer Science, Saint Louis, Missouri, Washington University, M.S. Thesis.

Snover, Matthew G., Gaja E. Jarosz, and Michael R. Brent. 2002. Unsupervised Learning of Morphology Using a Novel Directed Search Algorithm: Taking the First Step. *ACL Special Interest Group in Computational Phonology in cooperation with the ACL Special Interest Group in Natural Language Learning (SIGPHON/SIGNLL)*, University of Pennsylvania, Philadelphia, Pennsylvania, Association for Computational Linguistics, 11-20.

Snover, Matthew G., and Michael R. Brent. 2002. A Probabilistic Model for Learning Concatenative Morphology. *Neural Information Processing Systems Foundation (NIPS)*.

Sproat, Richard. (Ed.). 1997. *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Wicentowski, Richard. 2002. *Modelling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. Ph.D. Thesis, Baltimore, Maryland, The Johns Hopkins University.

Xanthos, Aris. 2007. *Apprentissage automatique de la morphologie: Le cas des structures racine-schème*. Ph.D. Thesis, Université de Lausanne.

Xu, Jinxi, and W. Bruce Croft. 1998. Corpus-Based Stemming Using Co-occurrence of Word Variants. *ACM Transactions on Information Systems (TOIS)*, 16.1, 61-81.

Yarowsky, David, and Richard Wicentowski. 2000. Minimally Supervised Morphological Analysis by Multimodal Alignment. *Association for Computational Linguistics (ACL)*.

Yarowsky, David, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. *Human Language Technology Research (HLT)*.

Zeman, Daniel. 2008. Using Unsupervised Paradigm Acquisition for Prefixes. *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark.

Appendix A:

A Summary of Common Spanish Suffixes

This appendix provides the basic knowledge of Spanish morphology needed to understand the examples which appear in the main body of this thesis. Spanish has both inflectional and derivational morphology. Since ParaMor is designed to identify the basic organizational unit of inflectional morphology, namely the paradigm, this appendix primarily describes the inflectional morphology of Spanish. However, Figure A.8, at the end of this appendix, lists a few common derivational suffixes of Spanish, including most derivational suffixes which occur in examples in this thesis. This guide to Spanish morphological paradigms is based on Gordon and Stillman (1999), an intermediate Spanish textbook.

Figures A.1 through A.7 present sets of inflectional suffixes and word-final clitics that together describe the inflectional morphology of Spanish. Figures A.1 through A.3 describe the inflectional morphology of Spanish verbs. Verbal suffixes in Spanish mark

combinations of ***Tense***, ***Aspect***, and ***Mood*** together with ***Subject Number***, ***Subject Person***, and in the case of past participles a ***Gender*** feature. The verbal paradigm has three inflection classes. The inflection classes of the verbal paradigm are morphologically stipulated—neither phonology nor syntax determines which inflection class a particular lexical verb belongs to.

Figures A.4 and A.5 give the two phonologically conditioned inflection classes of the ***Number*** paradigm on Spanish nouns. These two inflection classes mark *Singular* verses *Plural* on nouns, adjectives, and past participles. Figure A.6 contains the four suffixes which constitute the cross-product of the adjectival ***Number*** and ***Gender*** paradigms.

In addition to the inflectional suffixes of Figures A.1 through A.6, Spanish has a system of pronominal clitics which mimic true inflectional morphemes in Spanish orthography. Spanish pronominal clitics are written as a single orthographic word when they occur immediately following a non-finite verb form. There are three sets of pronominal clitics in Spanish which each masquerade as a separate paradigm: one set of clitics marks *Accusative* ***Case***, another *Dative* ***Case***, and a third clitic set contains reflexive pronouns. Figure A.7 presents the three sets of Spanish pronominal clitics. Although rare, it is possible for a single non-finite Spanish verb form to attach clitics from each separate set of pronominal clitics. In a Spanish written word that contained a clitic from each pronominal clitic set, the order of the clitics would be: Accusative, Dative, Reflexive—this is the order in which the columns of clitics appear in Figure A.7.

At times the main body of this thesis refers to the set of string unique inflectional Spanish suffixes (see for example Section 4.4.1). The Spanish suffixes which contribute to this set of inflectional suffixes are the surface forms which appear in Figures A.1 through A.7.

Infinitive	ar
Present Participle	ando

		Singular	**Plural**
Past Participle	**Feminine**	ada	adas
	Masculine	ado	ados

	1st Person Singular	**2nd Person Singular**	**3rd Person Singular**	**1st Person Plural**	**2nd or 3rd Person Plural**
Present Indicative	o	as	a	amos	an
Past Indicative Perfect	é	aste	ó	amos	aron
Past Indicative Imperfect	aba	abas	aba	ábamos	aban
Future Indicative	aré	arás	ará	aremos	arán
Conditional	aría	arías	aría	aríamos	arían
Subjunctive Perfect	e	es	e	emos	en
Subjunctive Imperfect	ara	aras	ara	áramos	aran

Figure A.1: The suffixes of the **ar** inflection class of Spanish verbs

Infinitive	er
Present Participle	iendo

		Singular	**Plural**
Past Participle	**Feminine**	ida	idas
	Masculine	ido	idos

	1st Person Singular	**2nd Person Singular**	**3rd Person Singular**	**1st Person Plural**	**2nd or 3rd Person Plural**
Present Indicative	o	es	e	emos	en
Past Indicative Perfect	í	iste	ió	imos	ieron
Past Indicative Imperfect	ía	ías	ía	íamos	ían
Future Indicative	eré	erás	erá	eremos	erán
Conditional	ería	erías	ería	eríamos	erían
Subjunctive Perfect	a	as	a	amos	an
Subjunctive Imperfect	iera	ieras	iera	iéramos	ieran

Figure A.2: The suffixes of the **er** inflection class of Spanish verbs

Infinitive	ir
Present Participle	iendo

		Singular	**Plural**
Past Participle	**Feminine**	ida	idas
	Masculine	ido	idos

	1st Person Singular	**2nd Person Singular**	**3rd Person Singular**	**1st Person Plural**	**2nd or 3rd Person Plural**
Present Indicative	o	es	e	imos	en
Past Indicative Perfect	í	iste	ió	imos	ieron
Past Indicative Imperfect	ía	ías	ía	íamos	ían
Future Indicative	iré	irás	irá	iremos	irán
Conditional	iría	irías	iría	iríamos	irían
Subjunctive Perfect	a	as	a	amos	an
Subjunctive Imperfect	iera	ieras	iera	iéramos	ieran

Figure A.3: The suffixes of the **ir** inflection class of Spanish verbs

Singular	Ø
Plural	s

Singular	Ø
Plural	es

Figures A.4 and A.5: The suffixes of the two inflection classes of the Spanish paradigm for ***Number*** on nouns and adjectives

	Masculine	**Feminine**
Singular	o	a
Plural	os	as

Figure A.6: The suffixes of the cross-product of the adjectival ***Gender*** and ***Number*** paradigms.

		Accusative	Dative	Reflexive
	1st Person Singular	me	me	me
	2nd Person Singular	te	te	te
Masculine	3rd Person Singular	lo	le	se
Feminine		la		
	1st Person Plural	nos	nos	nos
Masculine	2nd or 3rd Person Plural	los	les	se
Feminine		las		

Figure A.7: The pronominal clitics which appear in Spanish orthography as three separate paradigms of suffixes.

Derivational Suffix	Inflected Surface Forms that Appear in Examples in this Thesis	Meaning
ador	ador, adores, adora, adoras	Verb → Noun, Agentive of *ar* verbs
idor	idor, idores, idora, idoras	Verb → Noun, Agentive of *er* and *ir* verbs
ación	ación, ción, sión, aciones, etc.	Verb → Noun, Abstract noun
amente	amente, mente	Adjective → Adverb
idad	idad, idades	Adjective → Noun
izar	ización	Noun → Verb

Figure A.8: A few of the most frequent derivational suffixes of Spanish.

Appendix B:

Scheme Clustering, the Pseudo-Code

Pseudo-code implementing ParaMor's scheme-clustering algorithm as described in Section 4.3. At base ParaMor employs a bottom-up agglomerative clustering algorithm. However, Sections 4.3.1 and 4.3.2 describe several adaptations to the basic agglomerative algorithm that aid the clustering of scheme-models of paradigms. All of the adaptations discussed in Sections 4.3.1 and 4.3.2 are present in the pseudo-code that begins on the next page.

```
//////////////////////////////////////////
// The Data Structure of a Scheme-Cluster //
//////////////////////////////////////////

// Each cluster is one of two types:
//   1) A leaf cluster is a wrapper for a scheme
//   2) A compound cluster is a cluster that results from the merger of
//      two child clusters. Each child cluster of a compound cluster can
//      either be a leaf cluster or a compound cluster itself.
//
// Leaf clusters and compound clusters inherit common properties from
// the base Cluster struct.
struct Cluster {

  // For every 'small' scheme a CompoundCluster contains, that
  // CompoundCluster must contain one 'large' scheme (See Section
  // 4.3.2). Hence, each cluster, whether a Leaf or a CompoundCluster,
  // keeps track of the number of 'large' and small' schemes it
  // contains.
  largeSchemeCount = 0;
  smallSchemeCount = 0;

  // Each Cluster keeps track of the set of morpheme-boundary annotated
  // word types that license the Cluster
  licensingBoundaryAnnotatedTypes = null;
};

struct Leaf subStructOf Cluster {
  scheme = null;
};

struct CompoundCluster subStructOf Cluster {
  childA = null;
  childB = null;
};


////////////////////////////////////////////////////////////
// The Bottom-Up Agglomerative Scheme-Clustering Algorithm //
////////////////////////////////////////////////////////////

clusterBottomUp(schemes) {
  clusters = buildLeafClusters(schemes);
  while (true) {
    [bestClusterA, bestClusterB] = findClustersToMerge(clusters);

    // Stop clustering when there is no pair of clusters that are
    // permitted to merge.
    if (bestClusterA == null)
      return clusters;

    newCluster = merge(bestClusterA, bestClusterB);
    clusters.removeAll(bestClusterA, bestClusterB);
    clusters.add(newCluster);
  }
}
```

```
buildLeafClusters(schemes) {
  foreach (scheme in schemes) {
    leaf = new Leaf();
    leaf.scheme = scheme;
    if (scheme.numberOfLicensingWords >= threshold) {
      leaf.largeSchemeCount = 1;
    } else {
      leaf.smallSchemeCount = 1;
    }
    foreach (cSuffix in scheme.cSuffixes) {
      foreach (cStem in scheme.cStems) {
        leaf.licensingBoundaryAnnotatedTypes.add(cStem.'+'.cSuffix);
      }
    }
    leafClusters.add(leaf);
  }
  return leafClusters;
}

findClustersToMerge(clusters) {
  bestClusterScore = null;
  clusterToMergeA, clusterToMergeB = null;
  foreach (clusterA in clusters) {
    foreach (clusterB greaterThan clusterA in clusters) {
      if (isMergePermitted(clusterA, clusterB)) {
        score =
          cosine(clusterA.licensingBoundaryAnnotatedTypes,
                 clusterB.licensingBoundaryAnnotatedTypes);
        if (score > bestClusterScore) {
          bestClusterScore = score;
          clusterToMergeA = clusterA;
          clusterToMergeB = clusterB;
        }
      }
    }
  }
  return [clusterToMergeA, clusterToMergeB];
}

merge(clusterA, clusterB) {
  newCluster = new CompoundCluster();

  newCluster.childA = clusterA;
  newCluster.childB = clusterB;
  newCluster.largeSchemeCount =
    clusterA.largeSchemeCount + clusterB.largeSchemeCount;
  newCluster.smallSchemeCount =
    clusterA.smallSchemeCount + clusterB.smallSchemeCount;
  newCluster.licensingBoundaryAnnotatedTypes =
    union(clusterA.licensingBoundaryAnnotatedTypes,
          clusterB.licensingBoundaryAnnotatedTypes);

  return newCluster;
}
```

```
isMergePermitted(clusterA, clusterB) {

  // There are two restrictions on clustering
  //
  // Restriction 1: ParaMor discriminatively requires that each pair
  //   of c-suffixes in each cluster be able to form words that
  //   occurred in the paradigm induction corpus by separately
  //   attaching to a common c-stem.
  foreach (cSuffixA in clusterA) {
    foreach (cSuffixB in clusterB) {
      schemeOfTwoCSuffixes =
        dynamicSchemeNetwork.generateScheme(cSuffixA, cSuffixB);
      if (schemeOfTwoCSuffixes.cStems.size == 0)
        return false;
    }
  }

  // Restriction 2: ParaMor requires that there be at least as many
  //   'large' schemes as there are 'small' schemes in each cluster.
  if ((clusterA.smallSchemeCount + clusterB.smallSchemeCount) >
      (clusterA.largeSchemeCount + clusterB.largeSchemeCount)) {

    return false;
  }

  return true;
}
```