Morphology: Making a lexicon

4

4.1 General remarks on morphology

The field of morphology has as its domain the study of internal word structure, and in practice that has meant the study of three relatively autonomous aspects of natural language, which one can identify as morphophonology, morphosyntax, and morphological decomposition. To explain what each covers, we must introduce the notion of *morph*—a natural, but not entirely uncontroversial notion. If we consider the written English words *jump, jumps, jumped*, and *jumping*, we note that they all begin with the string *jump*, and three of them are formed by following *jump* by *s, ed*, or *ing*. When words can be decomposed directly into such pieces, and when the pieces recur in a functionally regular way, we call those pieces *morphs*.

- Morphophonology. It is often the case that two (or more) morphs are similar in form, play a nearly identical role in the language, and each can be analytically understood as the realization of a single abstract element—abstract merely in the sense that it characterizes a particular grammatical function, and abstracts away from one or more changes in spelling or pronunciation. For example, the regular way in which nouns form a plural in English is with a suffixal *-s*, but words ending in *s, sh*, and *ch* form their plurals with a suffixal *-es*. Both *-s* and *-es* are thus morphs in English, and we may consider them as forming a class which we call a *morpheme*: *s, -es* whose grammatical function is to mark plural nouns. The principles that are involved in determining which morph is used as the correct realization of a morpheme in any given case is the responsibility of morphophonology. Morphophonology is, in a real sense, the shared responsibility of the disciplines of phonology and morphology.

- Morphosyntax. Syntax is the domain of language analysis responsible for the analysis of sentence formation, given an account of the words of a language. In the very simplest case, the syntactic structure of a well-formed sentence could conceivably be described as **noun-verb-noun**, where the first noun is the subject and the second the object, but grammar is never that simple; in reality, the morphs that appear in one word (for example, verbal suffixes) may also specify information about the subject or the object (for example, the verbal suffix *-s* in *Sincerity frightens John* specifies that the subject of the verb is grammatically singular). Morphosyntax is the shared responsibility of the disciplines of syntax and morphology.

- Morphological decomposition. While English has many words which contain only a single morpheme (e.g., *while, class, change*), it also has many words that are decomposable into morphs, with one or more suffixes (*help-ful*, *thought-less-ness*), one or more prefixes (*out-last*,) or combinations (*un-help-ful*). But English is rather on the tame side as natural languages go; many languages regularly have several affixes in their nouns, adjectives, and even more often, their verbs. (e.g., Spanish *bon-it-a-s*).

Three interrelated questions:

- Word segmentation: How can we develop a *language-independent* algorithm that takes as input a large sequence of symbols representing letters or phonemes and provides as output that same sequence with an indication of how the sequence is divided into words?
- How can we develop a language-independent algorithm that takes as input a list of words and provides as output a segmentation of the words into morphemes, appropriately labeled as prefix, stem, or suffix—in sum, a morphology of the language that produced the word list?
- How can we implement our knowledge of morphology in computational systems in order to improve performance in natural language processing?

General comments here.

Morphological decomposition. Conversion; compounding.

Inflectional and derivational morphology. A useful distinction is generally made between derivational and inflectional morphology. The distinction falls squarely on whether the phenomenon one is considering is relevant to morphosyntax or not. If it is relevant, then it is considered inflectional morphology, and otherwise it is considered derivational morphology.

Users of natural languages (which is to say, all of us) need no persuasion that words are naturally occurring units. We may quibble as to whether expressions like "of course" should be treated as one word or two, but there is no disagreement about the notion that sentences can be analytically broken down into component words.

In all, or virtually all, languages, it is appropriate to analytically break words down into component pieces, called morphemes; such an analysis is called a morphology, and is the central subject of this chapter. Morphologies are motivated by three considerations: (1) the discovery of regularities and redundancies in the lexicon of a language (such as the pattern in *walk:walks:walking :: jump:jumps:jumping*); (2) the need to predict the occurrences of words not found in a training corpus (e.g.); and (3) the usefulness of breaking words into parts in order to achieve better models for statistical translation and other models particularly sensitive to the meaning of a message.(explain).

Thus morphological models offer a level of segmentation that is typically larger than the individual $letter$, and typically smaller than the $word$. For example, the English word $unhelpful$ can be analyzed as a single word, as a sequence of nine letters, or from a morphological point of view as a sequence of the prefix un, the stem $help$, and the suffix ful.

4.2 Big Picture question

[1]

Can we build a picture of linguistics in which the goal is to specify a function mapping from the spaces of corpora $\times$ space of grammars such that for a fixed corpus, the optimal value of the function identifies the grammar that is in some *linguistic* sense correct? $g^* = \arg\max_g F(C, g)$, where C is a given set of observations ("corpus"), and $g \in \mathcal{G}$: how much is gained by restricting the set $\mathcal{G}$? Such restrictions amount to an assumption about innate knowledge/Univeral Grammar. An alternative strategy is (following Rissanen) to choose a Universal Turing Machine (UTM), and assign a probability to a grammar equal to $2^{-|l(g)|}$, where $|l(g)|$ is the length of the shortest implementation of grammar g on this particular UTM. Does it matter that (1) this statement does not offer any hope that we can recognize the shortest implementation when we see it, or (2) we have no way to choose among UTMs: how do we determine whether UTM-choice matters, in a world of finite data and in which limits may not be taken?

[2] If we want to tackle the problem of discovering linguistic structure, both phonology and syntax have the problem that their structure is heavily influenced by the nature of sound and perception (in the case of phonology) and of meaning and logical structure, in the case of syntax. Morphology is less influenced by such matters, and it is possible to emphasize both cross-linguistic variation and formal simplicity. *It is a good test case for language-learning from a computational point of view.*

[3] The design of an appropriate objective function—explicating what the description length of a morphology is—is half the project; the other half is designing appropriate and workable discovery heuristics.

[4] The goal is not to provide a morphology of English: it is to develop a language-independent morphology learner. Standard orthography (when it departs from phonemic representations) has rules that are similar to (and of the same type, in general) as the rules we find in phonology.

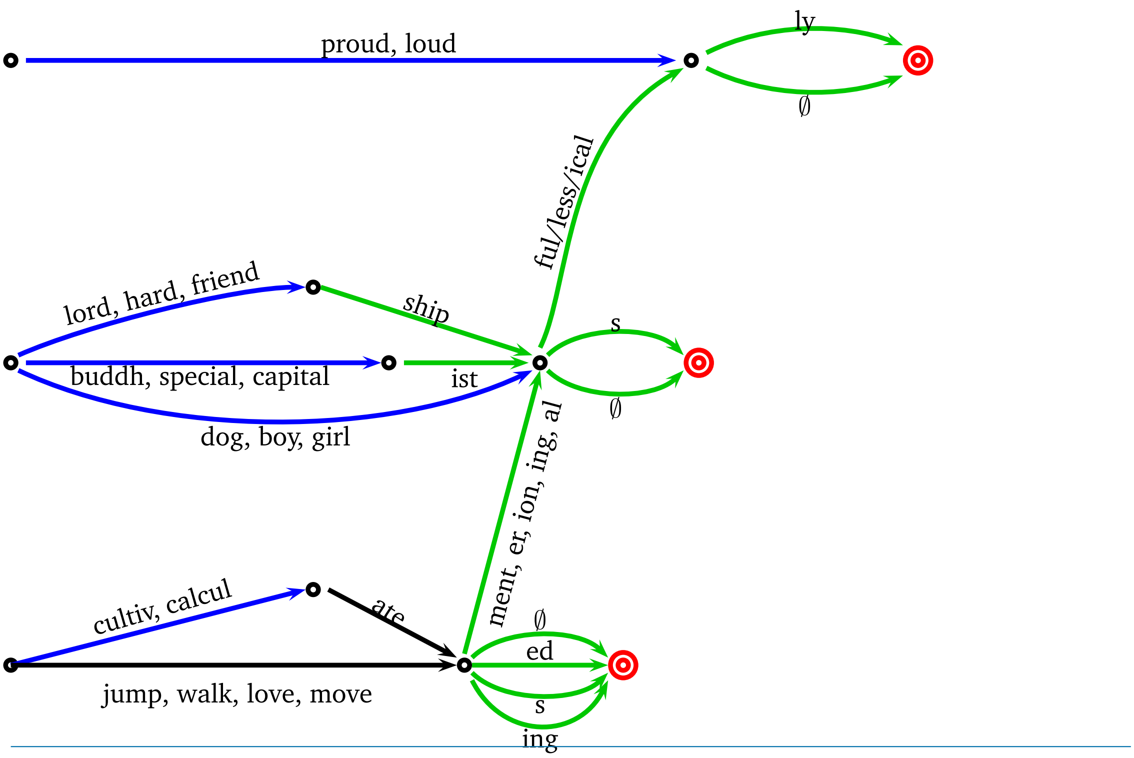


Figure 4.3.1 English morphology: morphemes associated with nodes of an FSA

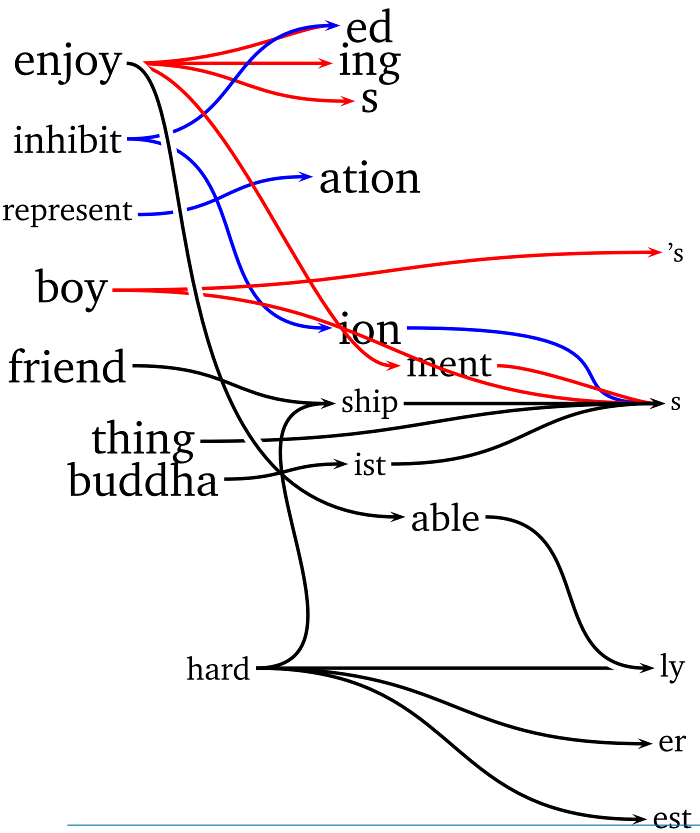
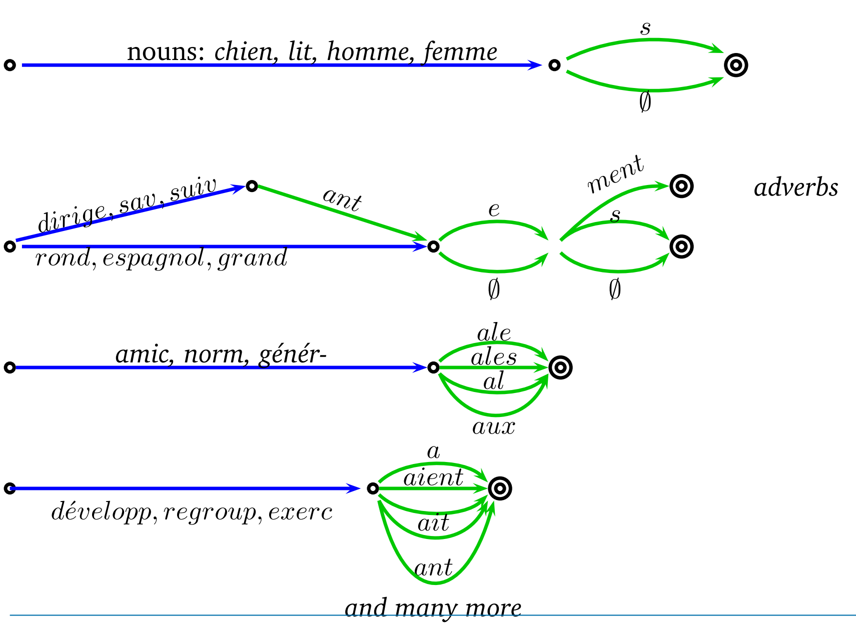
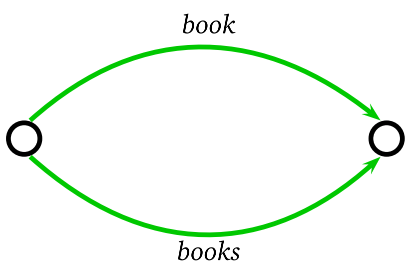


Figure 4.3.2 French



4.3 Morph discovery: breaking words into pieces

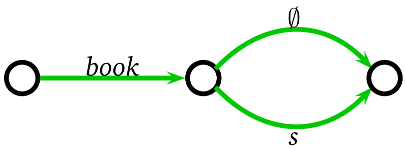


States	
number	'pointer to me'
0	0
1	1
	2
sum	65 bits

Edges			
number	states	encoding of states	'pointer to me'
0	(0,1)	0 1	0
1	(0,1)	0 1	1
		4	2

Labels	
edge ptr.	label
0	book#
1	books#
2	55

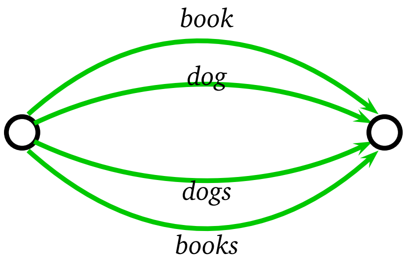
[1] $g^* = \arg\max_g F(C, g)$, where C is a given set of observations ("corpus"). Classical MDL offers the joint probability of the data and model as its candidate for F.
[2] Why **morphology**?
[3] 2 goals: objective function and learning heuristics
[4] Why conventional orthography? Why not phonemes?



States	
number	‘pointer to me’
0	0
1	10
2	11
	5
sum	66 bits

Edges			
number	states	encoding of states	‘pointer to me’
0	(0,1)	0 10	0
1	(1,2)	10 11	10
2	(1,2)	10 11	11
		11	5

Labels	
edge ptr.	label
0	book#
10	#
11	s#
5	40

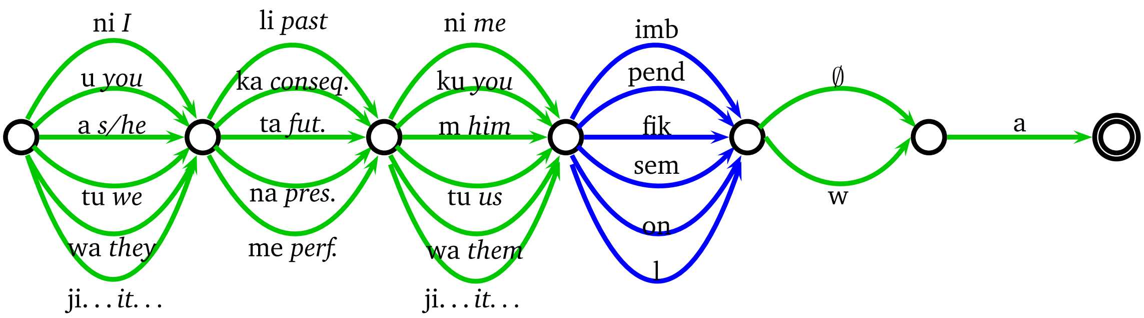


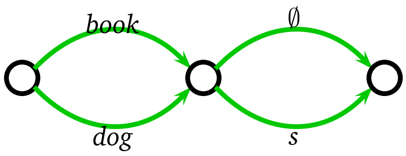
States	
number	‘pointer to me’
0	0
1	1
	2
sum	126 bits

Edges			
number	states	encoding of states	‘pointer to me’
0	(0,1)	0 1	00
1	(0,1)	0 1	01
2	(0,1)	0 1	10
3	(0,1)	0 1	11
		8	8

Labels	
edge ptr.	label
00	dog#
10	dogs#
10	book#
11	books#
8	100

Figure 4.3.3 Swahili verbal morphology





States

number	'pointer to me'
0	0
1	10
2	11
	5
sum	95 bits

Edges

number	states	encoding of states	'pointer to me'
0	(0,1)	0 10	00
1	(0,1)	0 10	01
2	(1,2)	10 11	10
3	(1,2)	10 11	11
		14	8

Labels

edge ptr.	label
00	dog#
01	book#
10	#
11	s#
8	60

- How do we choose a morphology (algorithmically)? We want one that endows the data with structure, but not too much. We want to extract redundancy in the data, but not spurious redundancy. In short: how do we find the boundary between real and spurious generalizations regarding word-internal structure?

Figure 4.3.4 Bit cost of signature-based morphology: one particular way to do it (not the only way!)

List of stems:

$$\sum_{t \in Stems} \sum_{i=1}^{|t|+1} -log\, p(t_i|t_{i-1})$$

List of affixes:

$$\sum_{f \in Affixes} \sum_{i=1}^{|f|+1} -log\, p(f_i|f_{i-1})$$

Signatures:

$$\sum_{\sigma \in Signatures} \left(\sum_{stem\, t \in \sigma} -log\, p(t) + \sum_{suffix\, f \in \sigma} -log\, p(f) \right)$$

Figure 4.3.5 Word probability model: w is $word$, t $stem$, f $suffix$

$p(word) = pr(\sigma_W) * pr(t|\sigma_w) * p(f|\sigma)$,
where word w = stem t + suffix f; each stem belongs to a single signature.
.

Figure 4.3.6 More generally, an acyclic FSA. Natural identity between words and paths through the FSA: $w \approx path_w$. There are various natural, and not so natural, ways to assign these distributions.

PFSA $(\mathcal{V}, \mathcal{E}, \mathcal{L})$, with 4 distributions:
(a) $pr_1(\,)$over $\mathcal{E}$ s.t. $\sum_j pr_1(e_{i,j}) = 1$; (b) $pr_2()$ over $\mathcal{V}$;
(c) $pr_3()$ over $\mathcal{L}$ (labels, i.e., morphemes), and
(d) $pr_4()$ over Σ, i.e., the alphabet used for $\mathcal{L}$.
Then $p(w) = p(path_w) = \prod_{e \in path_w} pr_1(e)$.;
$|FSA| = |\mathcal{V}| + |\mathcal{E}| + |\mathcal{L}|$.
$|\mathcal{V}| = \sum_{v \in \mathcal{V}} |v|$, where $|v| = -logpr_2(v)$.
$|\mathcal{E}| = \sum_{e \in \mathcal{E}} |e|$, where $|e_{ij}| = |v_i| + |v_j| + |ptr(label_e)|$, and $|ptr(label_e)| = -logpr_3(label_e)$.
$|\mathcal{L}| = \sum_{l \in \mathcal{L}} |l|; |l| = -\sum_i logpr_4(l_i)$.

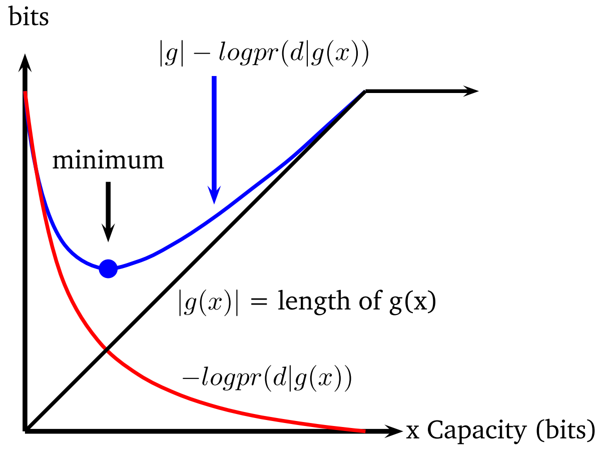
- The ideal solution would be one in which we could specify a general function LT ("*linguistic theory*")from pairs of grammar and data to the real numbers: G is the set of all grammars, and D the set of all data. $LT(G, D) \rightarrow Reals$ with the property that

  if $LT(g_1, d) < LT(g_2, d)$, then g_1 is a better grammar than g_2 for the data d (whatever "better" means to you—this is just a way of saying that it would be ideal if we could write an explicit function to the reals which expresses our grammatical theory's preferences); here, smaller is better, and we are looking for a minimum.

- **Probability** allows an elegant and natural solution. We may elect to choose the grammar which is the *most probable*, given the data (and the technical term here is *maximum likelihood*: roughly speaking, probabilities for theories are really *likelihoods*)

Figure 4.3.7 MDL optimization

Interpreting this graph: The x-axis and y-axis both quantities measured in *bits*. The x-axis marks how many bits we are allowed to use to write a grammar to describe the data: the more bits we are allowed, the better our description will be, until the point where we are over-fitting the data. Thus each point along the x-axis represents a possible grammar-length; but for any given length l, we care only about the grammar g that assigns the highest probability to the data, i.e., the $best$ grammar. The red line indicates how many bits of data are left unexplained by the grammar, a quantity which is equal to -1 * log probability of the data as assigned by the grammar. The blue line shows the sum of these two qunantities (which is the conditional *description length* of the data). The black line gives the length of the grammar.



Find g^* such that g* $= \arg\max_g pr(g|d) = \arg\max_g pr(d|g)pr(g)$

So to use this, we need to

1. specify that our grammars (which generate data) are *probabilistic*, i.e., every form that is output is assigned a probability, which sums to 1.0 over the infinite class of outputs; and part of our test is what the probability that it assigns to the actual data;

2. we need to specify what $pr(g)$ means. It needs to be a function that maps all possible grammars to reals between 0 and 1, and the (infinite) sum of these probabilities is 1.0. The most natural way to do this is to require the grammars to be expressed in binary format, and then take the probability of a particular grammar to be $2^{-1*length(g)}$.

If we do this, then we can replace the argmax with an argmin:

Find g^* such that g* $= \arg\min_g$ [length of g - log $probability_g$ of (d)]

This is the proposal of minimum description length (MDL) analysis.

- An MDL solution thus involves (a) a statement of what possible grammars are, how to compute their probabilities and the probabilities that each assigns to any set of data) and (b) a proposal for search: how to we find the best (or nearly the best) grammar g*, given a set of data?

  Bear in mind that we can imagine lots of solutions to problem (b), all associated with the same solution to (a).

- Turning this into a linguistic project

Some details first on the MDL model, followed by some time to talk about the search methods.

We can use the term *length* (of something) to mean the *number of bits = amount of information* needed to specify it. Except where indicated, the probability distribution(s) involved are from maximum likelihood models. The *length* of an FSA is the number of bits needed to specify it, and it equals the sum of these things:

1. List of morphemes: assigning the phonological cost of establishing a lean class of morphemes. Avoid redundancy; minimize multiple use identical strings. The probability distribution here is over phonemes (letters).

$$\sum_{t \in morphemes} \sum_{i=1}^{|t|+1} -log\, pr_{phono}(t_i | t_{i-1})$$

2. List of nodes v: the cost of morpheme classes

$$\sum_{v \in Vertices} -log\, pr(v)$$

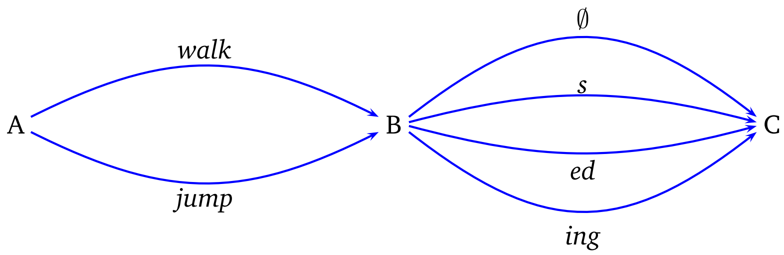
3. List of edges e: the cost of morphological structure: avoid morphological analysis except where it is helpful.

$$\sum_{e(v_1, v_2, m) \in \, Edges} -log\, pr(v_1) - log\, pr(v_2) - log\, pr(m)$$

(I leave off the specification of the probabilities on the FSA itself, which is also a cost that is specified in bits.)

In addition, a *word* generated by the morphology is the same as a *path* through the FSA. $Pr(w)$ = product of the choice probabilities of for w's path.

So: **for a given corpus, Linguistica seeks the FSA for which the description length of the corpus given the FSA is minimized,** which is something that can be done in an entirely language-independent and unsupervised fashion.



- English suffixes:

  NULL - s - ed - ing - es- er - 's - e - ly - y - al - ers - in - ic - tion - ation - en - ies - ion - able - ity - ness - ous - ate - ent - ment - t (*burnt*) - ism - man - est - ant - ence - ated - ical - ance - tive - ating - less - d (*agreed*) - ted - men - a (*Americana, formul-a/-ate*) - n (*blow/blown*) - ful - or - ive - on - ian - age - ial - o (*command-o, concert-o*) ...

4.4 What is the question?

We identify morphemes due to frequency of occurrence: yes, but all of their sub-strings have at least as high a frequency, so frequency is only a small part of the matter; and due to the non-informativeness of their end with respect to what follows.

But those are *heuristics*: the real answer lies in formulating an FSA (with post-editing) that is simple, and generates the data.

4.4.1 Gibbs sampling

Word w is analyzed into morphemes $\{m_i\}$, indicated $\mathcal{M}$.
$M_{ct}(w)$: number of morphemes analyzed in word w (4 for *board ing house s*); this is the size of $\mathcal{M}$.
The length of morpheme m in symbols is indicated by $|m|$. The number of occurrences of morpheme m in the whole lexicon is $[m]$.

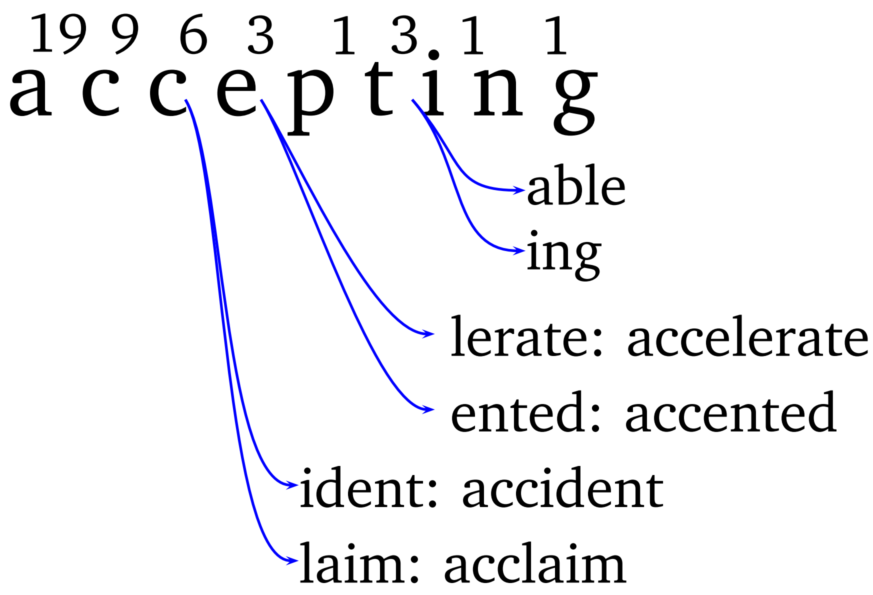
$$score = log(M_{ct}(w)) + \sum_{m \in \mathcal{M}} \frac{log(|m|!) + 5 \times |m|}{[m]} - log\, p(m)$$

morpheme	random	1 cycle	10 cycles	100 cycles
s	1639	1681	1253	1151
e	996	982	544	429
d	823	800	458	360
t	640	618	355	282
r	655	618	358	257
n	671	637	315	208
a	558	539	300	253
g	545	544	324	240
c	533	522	316	230
l	459	433	264	212
i	494	473	271	202
p	452	431	293	240
ing	235	461	1029	1059
’s	159	180	292	332
er	208	245	306	315
ed	431	532	640	631
-	45	–	102	363
es	241	289	277	262
re	174	211	242	287
ation	33	60	145	190
ness	26	134	154	154
able	27		140	174

random	1 cycle	10 cycles	100 cycles	200 cycles
board	board	board	board	board
board’s	board’s	board ’s	board’s	board ’s
boarded	boarded	board ed	board ed	board ed
bo ar der	bo ar der	board er	board er	board er
boarding	boarding	boar ding	boar ding	board ing
boardi nghouses	boardi nghouses	boar ding houses	board ing houses	board ing house s
bo ards	bo ards	board s	board s	board s
boast	boast	boast	boast	boast
boasted	boasted	boasted	boast ed	boast ed
bo as tfully	bo as tfully	boastfully	boast fully	boast fully
boasting	boasting	boasti ng	boast ing	boa sting
bo a stings	bo a stings	boastings	boast ings	boast ings
boasts	boasts	boasts	boast s	boast s
boat	boat	boat	boat	boat
boat-y ard	boat-y ard	boat-yard	boat-year	boat-yard

4.4.2 Putting phonology into the lexicon

Figure 4.4.1 Successor frequency



4.4.3 Putting segmentation structure in the lexicon: morphology 1

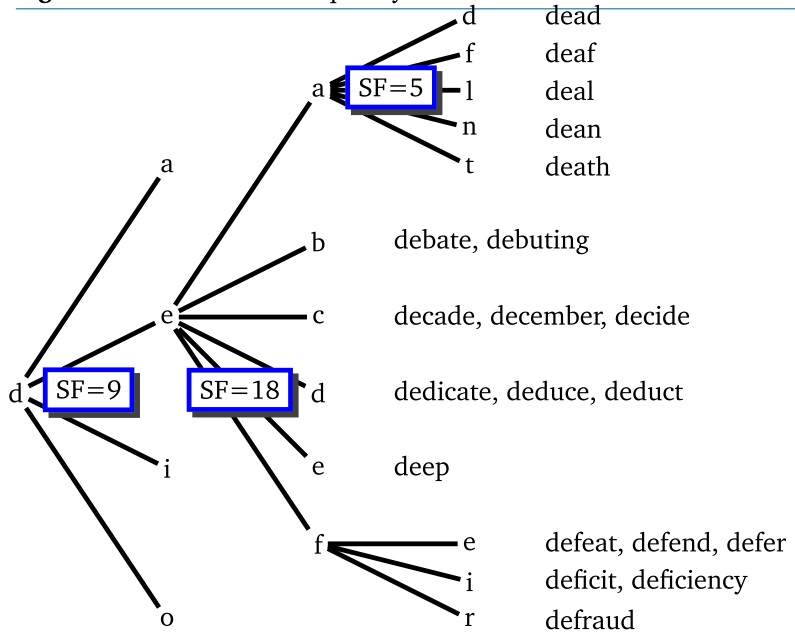
4.4.4 Successor Frequency

Zellig Harris 1955

4.5 What works better?

A better heuristic with about the same degree of simplicity is to look at word-final sequences of letters (if we are looking for suffixes), and evaluate them by multiplying their length times the number of times they occur. We will refer to this as the string's *robustness*. For a typical sample of written English of 14,000 words, we find the suffix ing occurring 961 times, and since its length is 3, that gives it a robustness score of 2,883. The second most robust word-final sequence in this corpus is s, which occurs 2,778 times, and thus has a robustness score of 2,778.

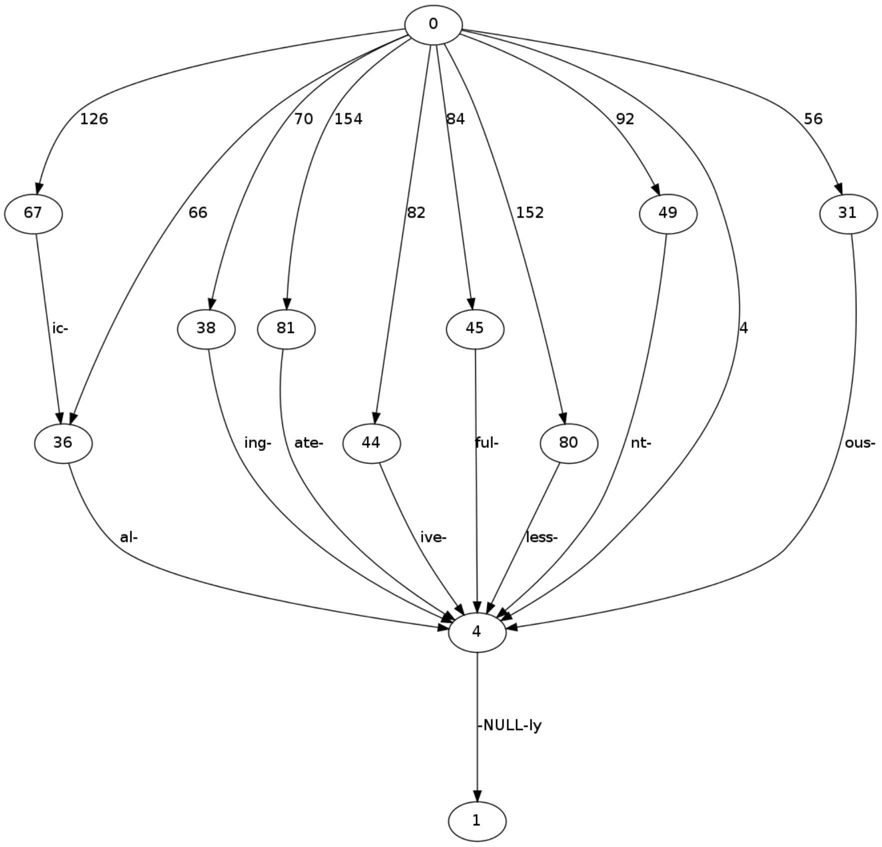
Figure 4.4.2 Successor frequency 2



4.6 adding layers of morphology

An initial morphology of the suffixes of English produces a very simple FSA. [example]

We ask each edge that is associated with a large set of stems to advance a set of candidates of stem-final suffixes, based on the count and the length of these candidate strings. For the stems that appear before *NULL-ly*, we obtain the following FSA:



Let us look at the morphemes associated with some of the edges. Edge 126, in the top left corner, contains the following labels (stems). The ones in blue are surely correct; the shorter ones, like *eth-* or *com-* are probably incorrect.

Edge number 126 To state: 67					
method	mag	log	ecolog	ideolog	psycholog
chronolog	graph	geograph	philosoph	eth	com
anatom	mechan	clin	cyn	typ	numer
categor	rhetor	histor	class	mathemat	tact
theoret	polit	uncrit	skept	vert	statist
analyt	paradox				

These are all analyzed as appearing before the suffix *-c*, and then *-al*, and then either followed by nothing or by *ly*.

Edge 66 is associated are stems that do not end in *-c*, but are followed by *-al*, and then either followed by nothing or by *ly*:

Edge number 66 To state: 36 Stem					
unequivoc	fisc	judici	unoffici	artifici	superfici
substanti	exponenti	quintessenti	potenti	sequenti	dism
phenomen	nomin	occasion	provision	congression	education
gravitation	fraction	addition	condition	uncondition	intention
convention	exception	proportion	unconstitution	etern	intern
cerebr	bilater	liter	sever	architectur	structur
accident	incident	coincident	increment	horizont	continu
usu	factu	contractu	perpetu	habitu	conceptu

How does this get produced? Here is an ordered list of the first 10 morphemes that are pulled out by this strategy:

Order:	From state:	Edge number	To state:	morpheme
1	20	37	2	er
2	21	39	2	tion
3	22	41	2	ing
4	23	43	5	e
5	24	44	6	e
6	25	46	2	ment
7	26	48	7	s
8	27	49	2	ist
9	28	51	24	at
10	29	53	2	ian

Let's look at the first morphemes that are specifically pulled out of the stems that precede NULL.s:

Order:	From state:	Edge number	To state:	morpheme
1	20	37	2	er
2	21	39	2	tion
3	22	41	2	ing
6	25	46	2	ment
8	27	49	2	ist
10	29	53	2	ian
11	30	55	2	tor
13	32	59	2	on
16	35	65	2	le
22	41	77	2	nce
23	42	79	2	nt
24	43	81	2	te
27	46	87	2	re
29	48	91	2	al
36	55	103	2	ne
37	56	105	2	et
39	58	109	2	ic
41	60	113	2	ship
42	61	115	2	out
44	63	119	2	de
45	64	121	2	ard
47	66	125	2	tive

The first set of stems has pulled off *-er* as a suffix on 540 words. In the following table, stems in blue are correct, and stems in green are arguably correct, though the vast majority of them are of the form *noun-verb-er*, where the noun is the object of the verb (as in *bartender*). Some cases are less regular: a *biographer* is not someone who biographs, but rather someone who writes biographies; but analyzing *biograph-er* seems perfectly reasonable.

Edge number 66 To state: 36 Stem					
scrubb	limb	climb	bomb	cucumb	plumb
trac	ulc	danc	announc	enforc	sauc
ringlead	cheerlead	load	grad	crusad	invad
shredd	feed	breed	raid	spid	provid
weld	homebuild	shipbuild	guild	fold	cardhold
stakehold	debthold	unithold	mold	bould	land
highland	island	salamand	command	bystand	defend
gend	spend	contend	bartend	bind	cind
remind	grind	transpond	decod	schrod	forward
camcord	intrud	auctione	conventione	overse	waf
coff	counteroff	lif	aquif	golf	surf
villag	teenag	pag	arbitrag	voyag	bridg
rodg	dagg	digg	jogg	mugg	folg
rang	strang	messeng	harbing	gunsling	ring
wing	charg	cheeseburg	hamburg	lug	bleach
schoolteach	ranch	launch	crunch	dispatch	watch
vouch	biograph	demograph	photograph	goph	philosoph
wash	dishwash	finish	extinguish	push	math
fanci	pacifi	amplifi	clothi	ski	chandeli
fli	highfli	colli	copi	photocopi	barri
couri	hoosi	dossi	fronti	courti	sneak
break	shak	lak	peacemak	pacemak	troublemak
dealmak	filmmak	carmak	moneymak	tak	caretak
hack	pack	meatpack	crack	firecrack	track
woodpeck	traffick	kick	slick	stick	knickerbock
block	rock	suck	seek	bik	hik
strik	talk	tank	think	drink	bunk
onlook	mark	casework	cowork	york	hawk
heal	gambl	assembl	recycl	peddl	toddl
swindl	feel	jewel	muffl	juggl	smuggl
mail	trail	fil	oil	sprinkl	install
resell	booksell	bestsell	tell	dwell	zell
kill	painkill	drill	thrill	roll	stroll
school	stapl	sampl	wrestl	hustl	settl
haul	rul	trawl	bowl	guzzl	dream
fram	ibm	disclaim	tim	programm	glimm
swimm	somm	drumm	newcom	monom	astronom
inform	perform	transform	polym	clean	afrikan
open	sweeten	fasten	listen	campaign	sign
bargain	complain	train	retain	entertain	din
berlin	airlin	jetlin	marin	bann	scann
beginn	spinn	sinn	forerunn	parishion	pension
practition	petition	question	common	soon	earn
northern	southern	eastern	western	midwestern	burn
vintn	kindergartn	down	landown	skyscrap	beep
peacekeep	housekeep	gatekeep	bookkeep	innkeep	shopkeep

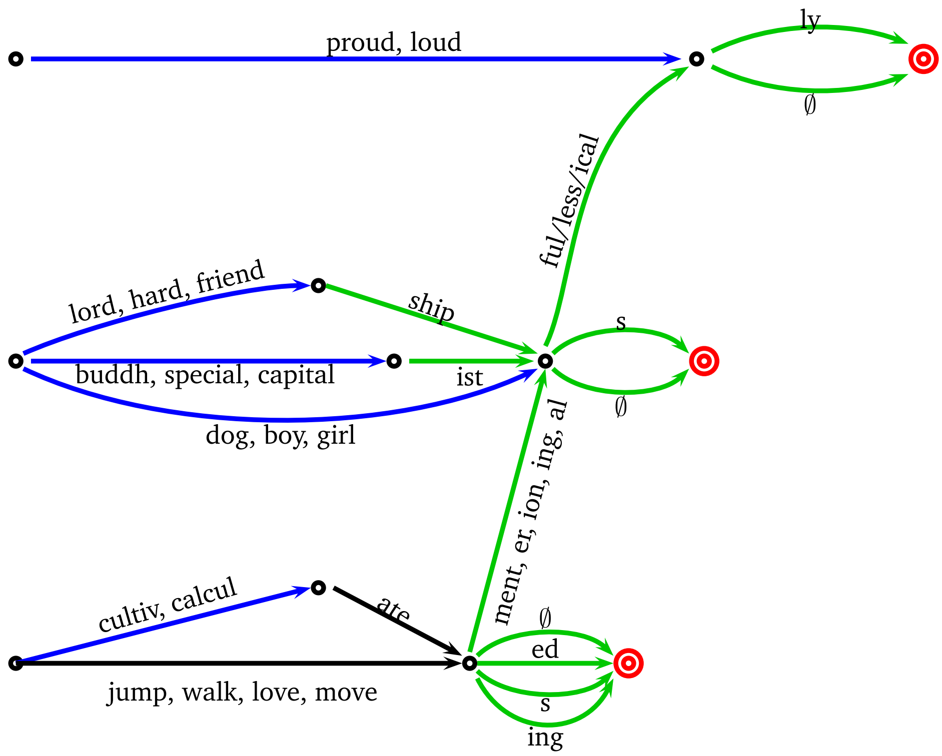
Edge number 66 To state: 36 (continued)					
minesweep	snip	junip	wip	help	camp
jump	interlop	troop	paratroop	rop	handicapp
rapp	wrapp	shipp	clipp	flipp	stripp
whopp	stopp	casp	jasp	bear	wear
murder	suffer	gather	cater	adulter	admir
labor	scor	explor	reinsur	lectur	adventur
las	rais	fundrais	apprais	exercis	merchandis
cruis	cleans	dispens	endors	pass	hairdress
accus	trous	heat	sweat	skat	float
floodwat	backwat	street	cathet	diet	telemarket
paramet	millimet	centimet	odomet	kilomet	thermomet
interpret	raft	draft	freight	fight	firefight
granddaught	stepdaught	wait	arbit	typewrit	songwrit
screenwrit	sportswrit	scriptwrit	copywrit	recruit	smelt
supercent	rent	dissent	point	headhunt	discount
scoot	shoot	adapt	chapt	helicopt	start
comfort	support	transport	frankfurt	forecast	postmast
roast	toast	disast	mobst	semest	forest
harvest	gangst	youngst	canist	pollst	hamst
rost	dumpst	bust	dust	adjust	platt
gett	sett	hitt	transmitt	critt	sitt
spott	cutt	gutt	putt	stutt	pollut
telecommut	minicomput	microcomput	supercomput	rescu	leagu
sav	lifesav	believ	reliev	nev	waiv
sliv	cabdriv	solv	revolv	holdov	changeov
hangov	rollov	mov	turnov	leftov	layov
observ	draw	review	interview	skew	widow
whistleblow	wildflow	sunflow	follow	mow	superpow
mix	box	ballplay	pay	ratepay	pray
moy	destroy	dry	fry	blaz	freez
stabiliz	fertiliz	tranquiliz	organiz	appetiz	bulldoz
analyz					

The second set of stems is this, based on a suffix *-tion*:

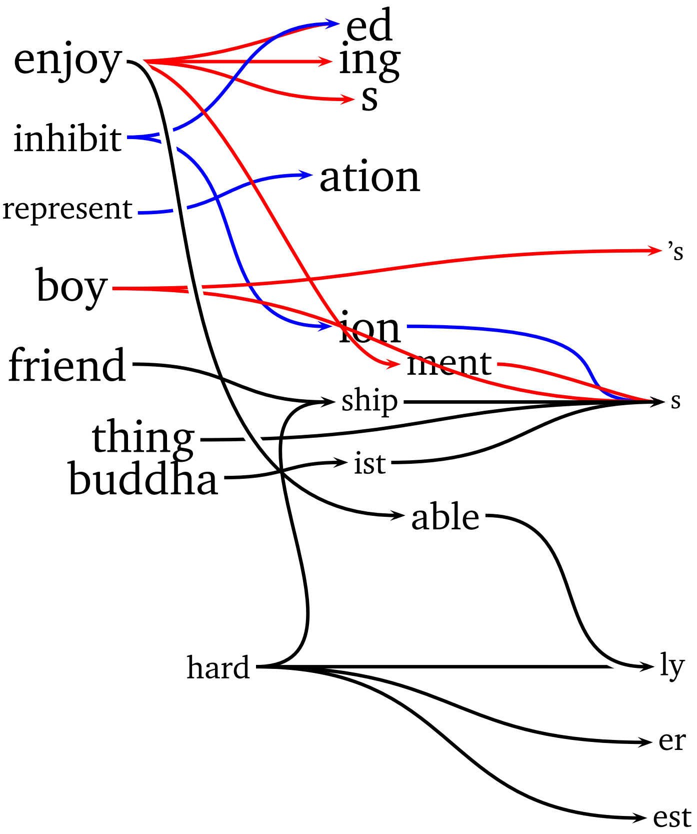
Edge number 66 To state: 36 Stem					
perturba	medica	indica	syndica	specifica	modifica
amplifica	magnifica	clarifica	classifica	identifica	certifica
implica	complica	applica	fabrica	loca	reloca
disloca	provoca	depreda	consolida	liquida	recommenda
delega	allega	obliga	interroga	denuncia	affilia
varia	appropria	negotia	renegotia	devia	abbrevia
revela	installa	cancella	viola	transla	specula
miscalcula	circula	regula	simula	formula	manipula
popula	congratula	proclama	exclama	affirma	confirma
transforma	explana	designa	resigna	combina	vaccina
origina	machina	inclina	examina	elimina	recrimina
denomina	termina	determina	rumina	assassina	destina
incarna	participa	preoccupa	declara	prepara	separa
vibra	delibera	reverbera	considera	exaggera	altera
aspira	expira	collabora	decora	perfora	explora
aberra	arbitra	concentra	registra	demonstra	illustra
configura	accusa	expecta	interpreta	cita	solicita
imita	limita	consulta	planta	presenta	misrepresenta
connota	quota	adapta	tempta	flirta	exhorta
manifesta	infesta	worksta	muta	reputa	amputa
valua	evalua	devalua	insinua	equa	fluctua
depriva	ova	renova	innova	observa	reserva
nationaliza	rationaliza	liberaliza	generaliza	capitaliza	hospitaliza
reorganiza	immuniza	characteriza	authoriza	dramatiza	privatiza
infrac	contrac	abstrac	distrac	attrac	defec
imperfec	rejec	injec	projec	selec	reflec
recollec	connec	interconnec	inspec	intersec	contradic
predic	afflic	depic	restric	evic	convic
injunc	concoc	abduc	deduc	reduc	reproduc
dele	comple	secre	inhibi	prohibi	exhibi
edi	rendi	precondi	defini	admoni	deposi
disposi	exposi	repeti	supersti	tui	deten
absten	atten	inven	lo	no	po
decep	misconcep	percep	mispercep	intercep	subscrip
prescrip	inscrip	redemp	exemp	assump	adop
interrup	disrup	asser	exer	por	distor
sugges	contribu	distribu	solu	resolu	substitu

Edge number 22 To state: 13 Stem
describ prescrib surfac outpac embrac balanc distanc experienc silenc sentenc influenc denounc persuad pervad cor

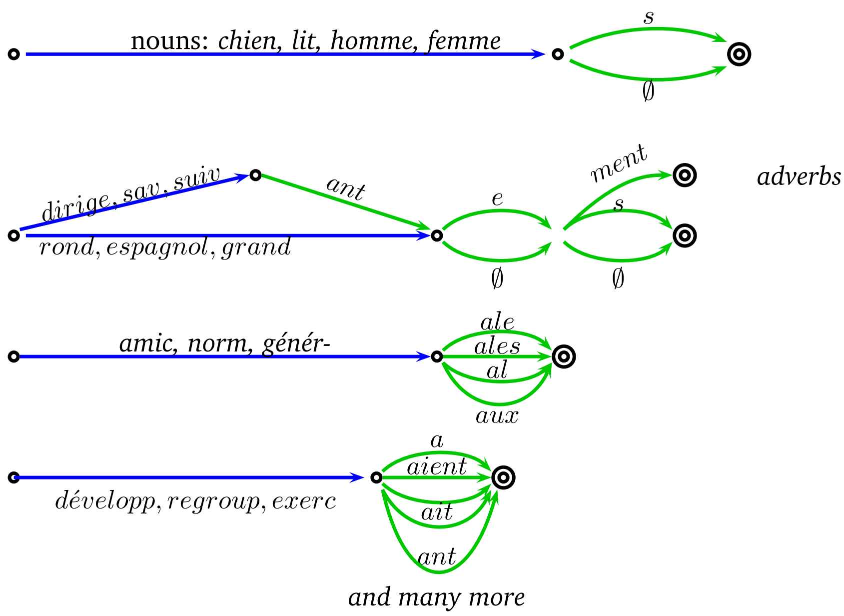
4.7 Immediate issues: getting the morphology right



English morphology: morphemes associated with nodes of an FSA



French



5

Signatures	Exemplar	Descr. Length (model)	Corpus Count	Stem Count	Source
NULL-s	accommodation	12996.7	13787	978	SF1
's-NULL	a*a*u	4237.23	8263	324	SF1
NULL-ly	according	3436.6	3391	259	SF1
NULL-ed-ing-s	account	886.936	2852	76	SF1
ed.ing	allott	1036.02	272	71	SF1
NULL.ed	abolish	1308.03	392	91	SF1
NULL.ed.s	accent	646.789	859	51	SF1
NULL.ing.s	boat	592.372	1060	46	SF1
NULL.ing	abound	1078.03	528	76	SF1
NULL.ed.ing	absorb	503.885	364	37	SF1
ing.s	awaken	172.814	29	11	SF1
ed.ing.s	fad	56.9268	13	3	SF1
's-NULL-s	afternoon	967.65	4258	83	SF1
e-ed-es-ing	accus	480.75	1345	40	Known stems to
e.ed.es	advanc	497.055	702	38	Check sigs
e.ed	acquiesc	825.969	311	58	Check sigs
e.ed.ing	anticipat	337.05	189	24	Known stems to
e.es.ing	battl	208.905	478	16	Known stems to
e.ing	abid	395.385	128	27	SF1
ed.es	aggravat	330.992	146	23	Check sigs
es.ing	celebrat	254.894	72	17	SF1
ed.es.ing	experienc	55.0602	35	3	From known sten
ies-y	abilit	899.932	642	66	SF1
NULL-al-s	addition	310.116	485	24	SF1
NULL.al	dramatic	87.2327	65	6	Check sigs
NULL-ly-s	absolute	320.709	468	25	SF1

1. Real versus accidental subcases: When should sub-signatures be subsumed by the "mother" signature? When are two signatures two samples from the same multinomial distribution? In some cases, this seems like a question with a clear meaning, as in case (a). Case (b) is less clear. Case (e) is interestingly different.

2. NULL-s *vs* NULL.ed.ing.s;

3. NULL-s *vs* NULL-s-'s

4. NULL-ed-ing-s *vs* NULL-ed-ing-ment-s

5. NULL-ed-er-ers-ing-s: how do we treat this?

6. NULL-ed-ing-s (vs) NULL-ing-s (e.g., *pull-pulling-pulls*); similar question arises for all so-called *strong* English verbs (this is a linguistically common situation).

7. The role of "post-editing": phonology and morphophonology. [6]

8. final *e*-deletion in English

9. C-doubling (*cut/cutting, hit/hitting; bite/bitten*)

10. *i/y* alternation: *beauty-beatiful; fly/flies;*

[5]**English**: NULL - s - ed - ing - es- er - 's - e - ly - y - al - ers - in - ic - tion - ation - en - ies - ion - able - ity - ness - ous - ate - ent - ment - t (*burnt*) - ism - man - est - ant - ence - ated - ical - ance - tive - ating - less - d (*agreed*) - ted - men - a (*Americana, formul-a/-ate*) - n (*blow/blown*) - ful - or - ive - on - ian - age - ial - o (*command-o, concert-o*) ...

[6]**French**: s - es - e- er - ent - ant - a - ée - é - és - ie - re - ement - tion - ique - ait - èrent - on - ées - te - ation - is - aient - al - ité - eur - aire - it - isme - en - age - ion - aux - ier - ale - iste - ien - t - eux - ance - ence - elle - iens - euse - ants - ienne - sion ...

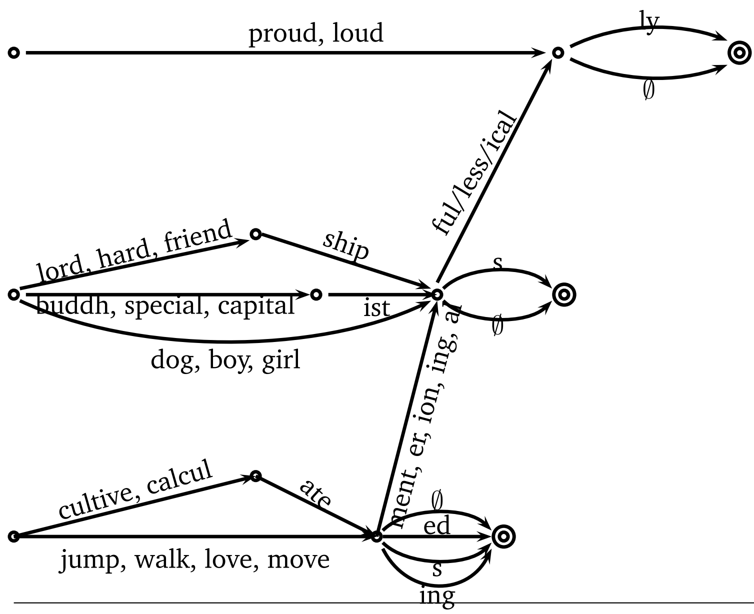
A calculation regarding a conjectured "phonological process" that falls half-way between heuristic and application of our DL-based objective function: Consider a process described as mapping $X \rightarrow Y/context$. [7] Rewrite the data as if that expressed an equivalence: we "divide" the data by that relation (for simplicity's sake, we ignore the context). [8] In this case, the result is a corpus from which all e's have been deleted. [9]What is the impact on the morphology that is induced from this new data? The lexical items are (of course) simpler (shorter). But the new morphology is *much* simpler than before, because *signatures* now collapse. *NULL.ed.ing.s* and *e.ed.es.ing* both map to *NULL.d.ing.s*. Each was of roughly the same order of magnitude; hence the bit cost of a pointer to the new signature is 1 bit less than that of the previous pointers, and that is a single bit of savings multiplied by thousands of times in the description length of the new corpus (quite independent of the missing es).

11. Succession of affixes: Stems of the signature NULL-s end in *ship, ist, ment, ing*. We can apply the analysis iteratively, re-analyzing all stems (and unanalyzed words), but this is not an adequate solution.

12. *NULL-ed-ing-s* vs. *t-ted-ts-ting* (Faulty MDL assumption?)

13. Clustering when no stem samples all its possible suffixes, but a family of them does: verbs in Romance languages.

Figure 4.7.1 What we would like to generate

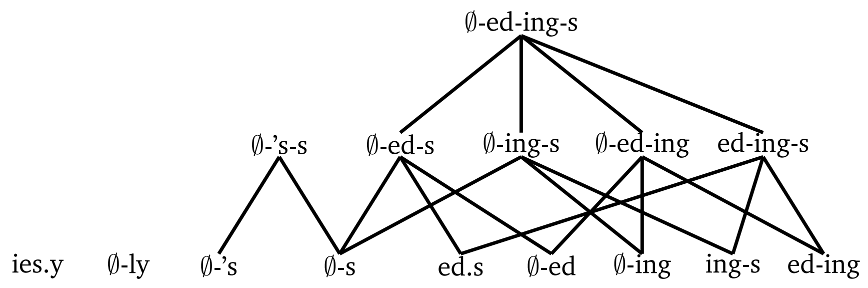


[7] $e \rightarrow \emptyset / -ed, -ing$

[8] $corpus \Rightarrow corpus/e \approx \emptyset$.

[9] *creeps* is now spelled *crps*, and *creeping* is *crping*.

Figure 4.7.2 Top signatures: First set



Signatures	Exemplar	Descr. Length (model)	Corpus Count	Stem Count
NULL-s	âge	42195.2	53520	2869
NULL-e-es-s	âgé	1338.17	5756	103
NULL.e	écaillé	2340.04	2038	151
NULL.e.es	éclatant	881.426	1740	62
NULL.e.s	élu	762.012	1474	54
NULL.es	ébranlé	1489.74	1010	97
e.es.s	asexué	200.907	339	13
e-ement-es	électriqu	1025	3516	77
ement.es	économiqu	784.345	802	53
e.ement	assèch	393.444	204	25
al-ale-ales-aux	aéropost	301.133	684	20
al.aux	élector	159.254	219	10
ale.aux	bilatér	59.4511	41	3
al.ale.aux	cruci	55.4465	11	2
ie-ique	allotrop	515.945	319	33
e-ent	élir	708.421	334	46
en-enne-ens	aéri	308.731	662	20
NULL-e-ement-es-s	étroit	160.381	1382	12
NULL.e.ement.es	clair	118.713	653	8
NULL.e.ement	aucun	38.1687	80	2
e-es-ique	anticyclon	265.309	542	18
e.ique	cinématograph	114.786	52	7
es.ique	artist	99.5247	105	6
ation-er	évapor	359.087	103	22
a-aient-ait-ant-e-ent-er-èrent-é-ée-ées-és	compos	115.702	701	3
a.ant.e.ent.er.èrent.é.ée.ées.és	entr	110.99	1084	4
a.e	belladon	403.511	134	25
er.é.ée.ées.és	enferm	121.61	68	6
a.e.ent.er.èrent.é.ée.ées.és	exerc	98.5132	130	3
a.aient.ait.ant.e.ent.er.é.ée.ées.és	privilégi	98.9718	343	2
a.ant	émerge	266.382	65	16
a.aient.ait.ant.e.ent.èrent.é.ée.ées.és	étudi	101.273	177	2
a.ait.ant	érige	135.998	98	8
a.er	abdiqu	239.706	49	14

4.8 Swahili

Typical case where morpheme frequency is more important than a count of the number of letters, in determining description length. The following is a correct change that this DL computation gets right:

$$ak + \{a, i\} + \{stems\} \rightarrow a + \{ka, ki\} + \{stems\}$$

Figure 4.7.3 3 Top signatures: inverted

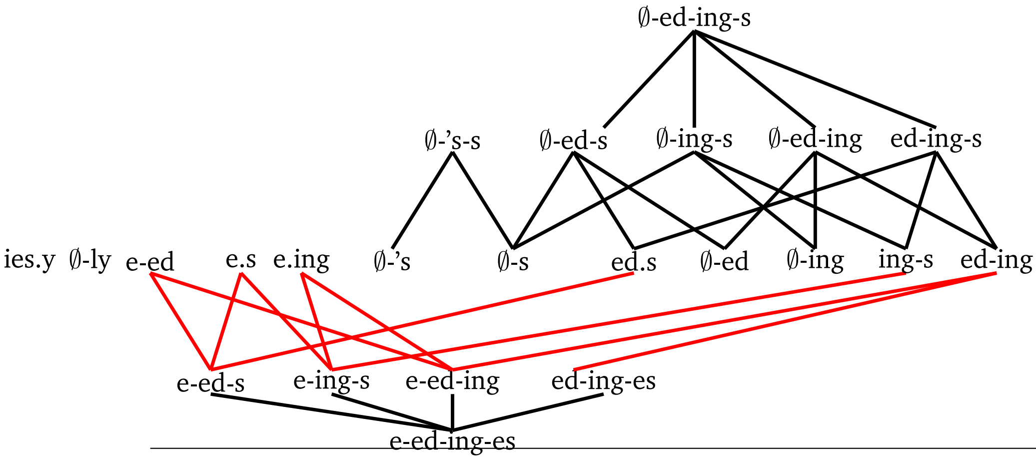


Figure 4.7.4 Stage 4

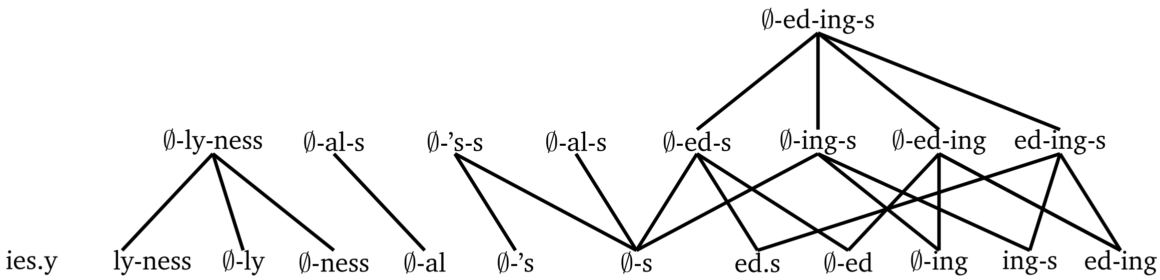
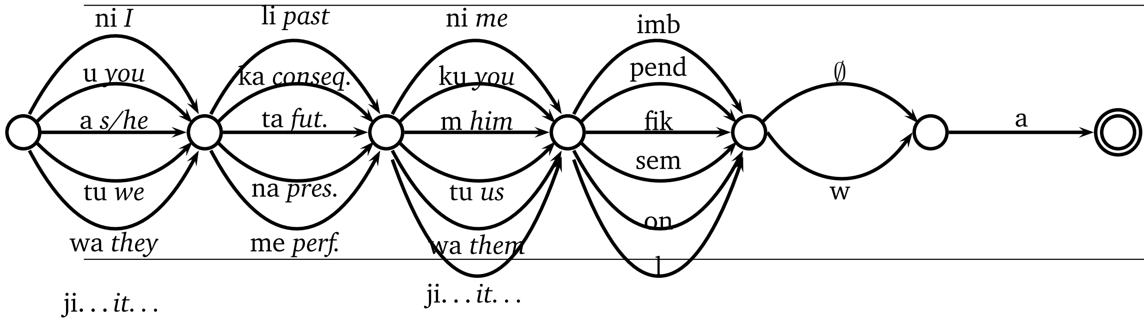


Figure 4.8.1 Simplified Swahili verbal morphology



because *ak* occurs nowhere else, but *ka* and *ki* are common. What is important is global, rather than local, parsimony.

4.8.1 String Edit Distance

4.8.2 Rich morphologies : morphology 2

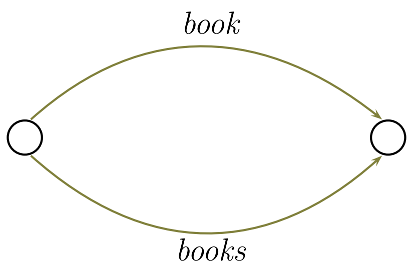
Linguistica

John Goldsmith

July 10, 2015

1 Cost in bits

1.1 A simple morphology



1.2 A simple signature

States

number	'pointer to me'
0	0
1	1
	2
sum	65 bits

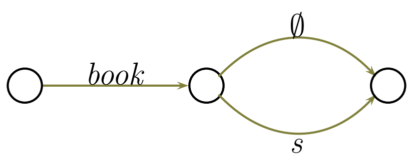
Edges

number	states	encoding of states	'pointer to me'
0	(0,1)	0 1	0
1	(0,1)	0 1	1
		4	2

Labels

edge ptr.	label
0	book#
1	books#
2	55

1.3

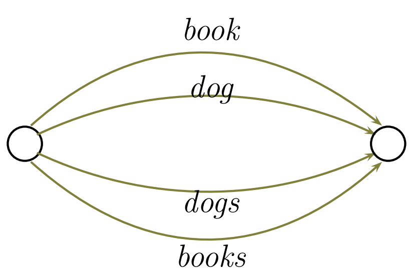


States	
number	'pointer to me'
0	0
1	10
2	11
	5
sum	66 bits

Edges			
number	states	encoding of states	'pointer to me'
0	(0,1)	0 10	0
1	(1,2)	10 11	10
2	(1,2)	10 11	11
		11	5

Labels	
edge ptr.	label
0	book#
10	#
11	s#
5	40

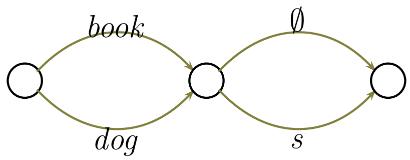
1.4 More complex signature



States	
number	'pointer to me'
0	0
1	1
	2
sum	126 bits

Edges			
number	states	encoding of states	'pointer to me'
0	(0,1)	0 1	00
1	(0,1)	0 1	01
2	(0,1)	0 1	10
3	(0,1)	0 1	11
		8	8

Labels	
edge ptr.	label
00	dog#
10	dogs#
10	book#
11	books#
8	100



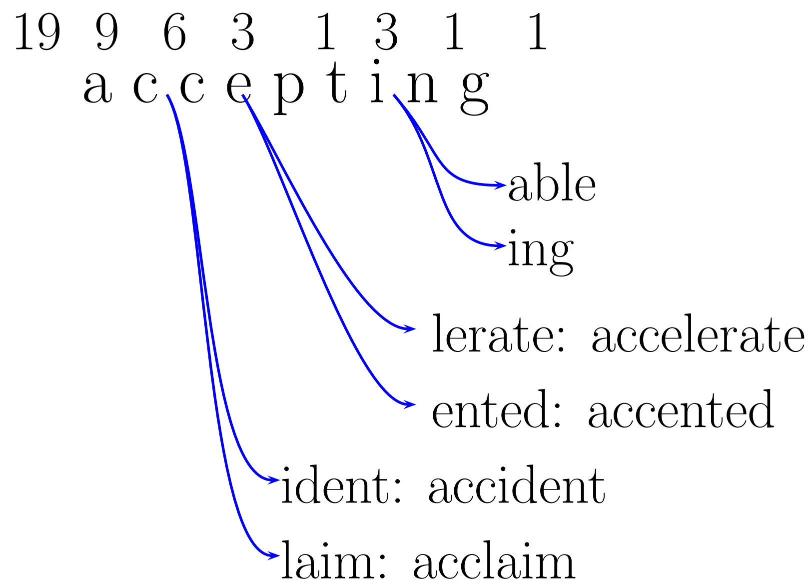
States		Edges				Labels	
number	'pointer to me'	number	states	encoding of states	'pointer to me'	edge ptr.	label
0	0	0	(0,1)	0 10	00	00	dog#
1	10	1	(0,1)	0 10	01	01	book#
2	11	2	(1,2)	10 11	10	10	#
		3	(1,2)	10 11	11	11	s#
	5			14	8	8	60
sum	95 bits						

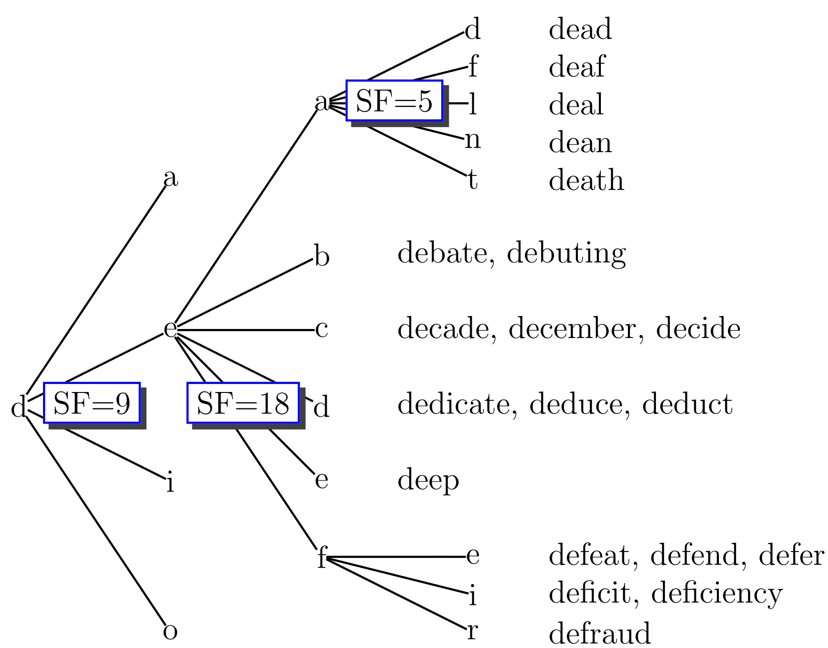
2 Learning morphology

One strategy is to begin with an initial heuristic, usually a conservative heuristic (high precision, low recall), and then use MDL to evaluate a lot of small, incremental changes. Linguistica 2001 used Harris's successor frequency as the first part of an initial heuristic.

2.1 Successor Frequency

Zellig Harris 1955





2.2 Initial heuristic:

1. Use (some version of) successor frequency to find some cuts in words.
2. If a word has more than one cut from previous step, ignore all but the last one.

3. If a word has a cut, call the piece on the left a *stem*, the piece on the right a *suffix*.
4. If the stem is too short or the suffix too long, remove the cut.
5. For each stem, collect all suffixes it appears with. Alphabetize those suffixes. If the stem appears as a free-standing word, add the suffix "NULL" to the set of suffixes.
6. Call each alphabetized set of suffixes a *signature*. Create a dictionary whose keys are signatures and whose values are lists of stems.
7. If a signature has fewer than Θ stems, remove that signature.

2.3 Local changes, evaluated by MDL calculation:

3 Cyclic reapplication

Word	Stem	inner layer	middle layer	outer layer
decline	declin		e	
declined	declin			ed
declines	declin			es
decolletage	decolletage			
decor	decor			
decorate	decor		at	e
decorating	decor		at	ing
decoration	decor	at	ion	s
decorative	decor		at	ive
decorator	decor		at	or
decorators	decor	at	or	s
decrease	decrease			
decree	decree			
decreeing	decree			ing
decried	decri			ed
decries	decri			es
dedicated	dedicat			ed

4 DL

1. States + Edges + Labels
2. Set of states S consists of a list of $|S|$ pointers, one to each state. This costs $|S| log |S|$. Each state has a *count* consisting of the number of words that passes through it; call the sum of those counts the *total morpheme count*. Then each state has a frequency equal to $\frac{\text{its count}}{\text{total morpheme count}}$. This forms a distribution over states. We assign an encoding to each state, whose length is equal to the plog of the state's frequency.

3. A set of edges: $e(i, j, m)$: a triple with pointers to the *from*-state, the *to*-state, and the label associated with that edge. Each edge costs you-know-what (right?) plus the length of the pointer to its label (a morpheme in the morpheme list; see below).

4. list of morphemes $\mathcal{M}$ (stems and affixes).

   Cost of the whole list is $log(length(\mathcal{M}))$

   + the phonological cost of each item on the list:

   $\sum_{m\in\mathcal{M}}\sum_{l\in m} plog\, pr(l)$.

   Associated with each morpheme is a frequency

   fr(m): $\frac{\text{number of words that contain it}}{\text{total number of morphemes used by all the words}}$,

   and a pointer to that morpheme costs $plog fr(m)$.

5. The theory of MDL leaves some questions unanswered: for example, should each stem in the stem-list have a pointer back to the signature in which it occurs? That is, how do we encode knowledge of how a particular stem *works*?

6. When we consider the relative cost of two morphologies, we will consider changes in each of these cost-components.

4.1 Typical early errors of proper signatures

1. **on & ve**:

affirmati	attenti	co-operati	destructi
imaginati	introspecti	positi	provocati
recepti	representati	15 more ...	

2. **l & tion**: differentia inaugura

3. **NULL & rs** ringside teenage

4. **ous & ty** tenaci vivaci

5. **e & y** admirabl audibl conceivabl considerabl equitabl formidabl honorabl impeccabl impossibl incomparabl incredibl indelibl irredeemabl justifiabl notabl predictabl preferabl reasonabl remarkabl terribl unavoidabl (4 more)

4.2 Detecting the first error: entropy of the ends of the stems

1. Measure how much variety there is among the last 1 (or 2,3,4) letters of the stems. If there's too much variety (= entropy), it's unlikely that the varying material ought to be in the suffixes. Rule of thumb: Entropy threshold : 1.5

   stem entropy for **on.ve**

Shift # letters: 1:	Entropy sufficiently small:	0
Shift # letters: 2:	Entropy sufficiently small:	0.987693 (why?)
Shift # letters: 3:	Entropy too large:	3.23619 (Threshold 1.5.)
Shift # letters: 4:	Entropy too large:	4.26269 (Threshold 1.5.)

2. suffix use by this signature:

affix	use count	Descr Length	Proportion of suffix info used by this signature
-on	26	7.685	0.885
-ve	23	7.862	1.000

Why do we consider the proportion of the suffix information used by this signature? The cost of an affix is motivated only by edges that employ it; and any signature should be expected to pay for its fair share of the bit-cost of a morpheme. If a morpheme is used by many signatures (i.e., edges), then it is less expensive for another signature to use it as well. "Le langage est un système où tout se tient."

Length of pointers to this signature:	180.833
Current signature's DL:	214.098

3. Entropy tells us to consider moving 1 or 2 letters to the right. Let's consider the case of moving 2 letters first.

4.2.1 Restructuring: First effort (which will fail to improve)

- First, consider moving **ti**, creating the following stems:

affirma	atten	co-opera	destruc
imagina	introspec	posi	provoca
recep	representa		

(We save some by shifting repeated *ti*s to the suffixes.)

and these suffixes: **tion** and **tive**:

Affix	Did affix already exist?	DL for this affix
tion	yes	7.138
tive	no	26.664

26.664 is a lot bigger, because this signature would have to pay for all of the new suffix.

Each stem contains a pointer to this signature; each such pointer costs 8.0639 bits.

Total bit cost of pointers to this sig: 80.639

Total for this signature: 114.441 bits

- Second, consider moving **si**, creating **sion** and **sive**

Affix	Did affix already exist?	DL for this affix
sion	no	26.664
sive	no	26.664

aggres	comprehen	conclu
deci	eva	exclu
expan	explo	indeci
percus	permis	persua
repres		

Pointers to this sig: 99.910
Total for this sig: 153.239

tion.tive	114.441
sion.sive	153.239
total of new analysis	267.68
old analysis	214.098

total for **tion.tive** and **sion.sive**: 267.680 compared to the original 214.098 That's a loser . . .

4.2.2 Second effort

Let's add one letter to the suffixes: i. This will save some phonological material on the stems; how about the suffixes?

1. New signature: ion.ive

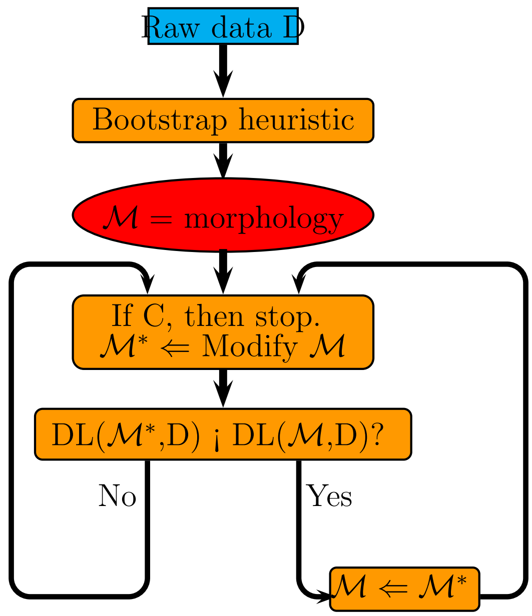
Affix	Did affix already exist?	Previous count	DL for this affix
ion	yes	85	18.211
ive	yes	5	26.664

2. Nice! New stems. . .

affirmat	aggress	attent	co-operat
comprehens	conclus	decis	destruct
evas	exclus	expans	explos
imaginat	indecis	introspect	percuss
permiss	persuas	posit	provocat
recept	representat	repress	

3. | | |
   |---|---|
   | ion.ive | 143.227 |
   | on.ve | 214.098 |

4. The new analysis wins (ion.ive) and the old analysis loses (on.ve).


Raw data D
Bootstrap heuristic
$\mathcal{M}$ = morphology
If C, then stop.
$\mathcal{M}^* \Leftarrow$ Modify $\mathcal{M}$
DL($\mathcal{M}^*$,D) ¡ DL($\mathcal{M}$,D)?
No
Yes
$\mathcal{M} \Leftarrow \mathcal{M}^*$