Class 2a: Word learning

2

2.1 Language induction: Word chunking

A good deal of work beginning in the late 1960s. Two widely-cited MIT dissertations in the mid 1990s on this, by Michael Brent and Carl de Marcken.

3749 sentences, 400,000 characters:
TheFultonCountyGrandJurysaidFridayaninvestigationofAtl anta'srecentprimaryelectionproducednoevidencetha
yirregularitiestookplace.f Thejuryfurthersaidinterm-endpresentmentsthattheCityE
xecutiveCommittee,whichhadover-allchargeoftheelecti on,deservesthepraiseandthanksoftheCityofAtlantaforthe
annerinwhichtheelectionwasconducted . . .

The Fulton County Grand Ju ry s aid Friday an investi gation of At l anta 's recent prim ary e lection produc ed no e videnc e that any ir regul ar it i es took place . Thejury further s aid in term - end present ment s thatthe City Ex ecutive Commit t e e ,which had over - all charg e ofthe e lection , d e serv e s the pra is e and than k softhe City of At l anta forthe man ner in whichthe e lection was conduc ted.

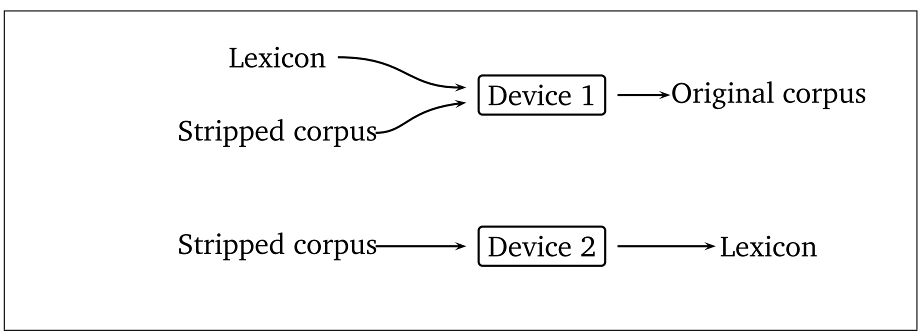
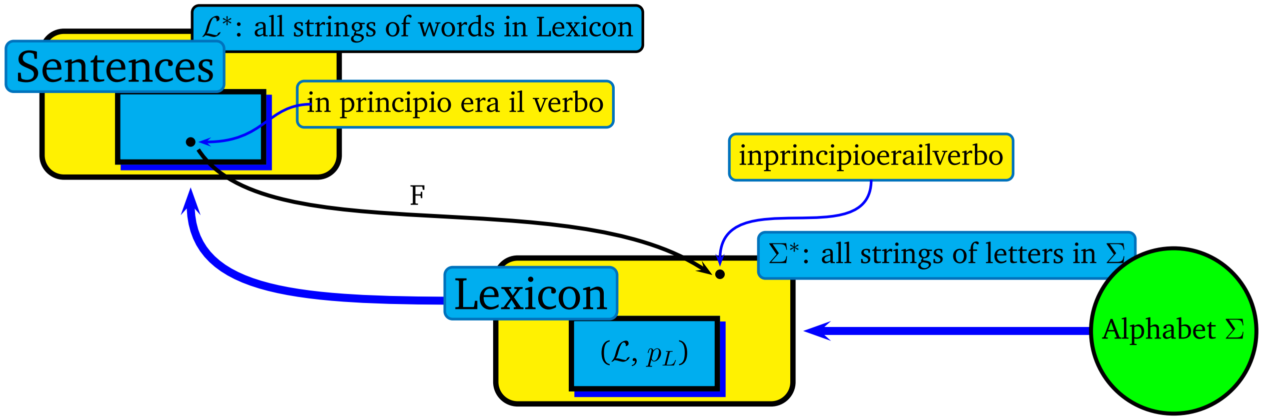


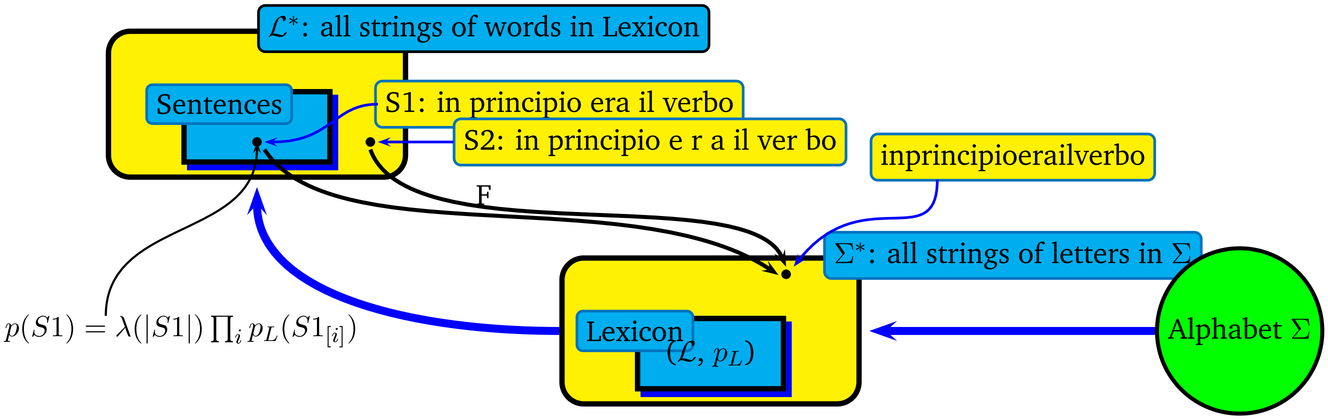
Fig. 2.1: The two problems of word segmentation



Select the lexicon $\mathcal{L}$ which minimizes the description length of the corpus $\mathcal{C}$. A lexicon $\mathcal{L}$ is a distribution $pr_{\mathcal{L}}$ over a subset of Σ^*. $\mathcal{L}$'s length is the length in bits in some specified format (the format matters!) and encoding. Any such distribution assigns a minimal encoding (up to trivial variants) to the corpus, and this encoding requires precisely $-logp(\mathcal{C})$ bits. The description length of a corpus given lexicon $\mathcal{L}$ is defined as $|\mathcal{L}| - logpr_{\mathcal{L}}\mathcal{C}$: select the lexicon that minimizes this quantity (as best you can). $|\mathcal{L}|$ comes into the picture because if we assume $\mathcal{L}$ is expressed in a binary-encoded format in which no morphology is a prefix of another, this encoding induces a natural probability distribution, with $p(l)$ proportional to $2^{|l|}$

A lexicon L is a pair of objects (L, p_L):

- a set $L \in A^*$, and
- a probability distribution p_L that is defined on A^* for which L is the support of p_L. We call L the words.
- We insist that A $\in$ L: all individual letters are words;
- We define a language as a subset of L^*; its members are sentences.
- Each sentence can be uniquely associated with an utterance (an element in A^*) by a mapping F:



Lexicon 1: a,b,c,. . . ,z

Lexicon 2: a,b,c,. . . ,t, th, . . . z

How do these two models of English compare? Why (and how) is Lexicon 2 better?

$[t]$	count of t
$[h]$	count of h
$[th]$	count of th
Z	total number of words (tokens)
	$= \sum_{m \in lexicon} [m]$

Let's compare the probability of the corpus under each of those assumptions regarding the correct lexicon. Let's break out the log probability of corpus $= \sum_{m\, in\, lexicon} [m] log \frac{[m]}{Z}$ into its component terms:

(i) all letters are separate words	(ii) th treated as a word
$[t]_1 log \frac{[t]_1}{Z_1}$ $[h]_1 log \frac{[h]_1}{Z_1}$ $\sum_{m \neq t,h} [m]_1 log \frac{[m]_1}{Z_1}$	$[t]_2 log \frac{[t]_2}{Z_2}$ $[h]_2 log \frac{[h]_2}{Z_2}$ $\sum_{m \neq t,h} [m]_1 log \frac{[m]_1}{Z_2}$ $[th]_2 log \frac{[th]_2}{Z_2}$
$[t]_1$ $[h]_1$ Z_1	$[t]_2 = [t]_1 - [th]$ $[h]_2 = [h]_1 - [th]$ $Z_2 = Z_1 - [th]$

Word discovery A good deal of work beginning in the late 1960s. Two widely-cited MIT dissertations in the mid 1990s on this, by Michael Brent and Carl de Marcken. We will explore this in detail, because the most important result that emerges from this work is that where the method fails, it fails for an extremely interesting reason: it fails because it does not know enough linguistics. This does not invalidate the overall conception; it means that the methods for extracting structure and system must be smarter than cookie-cutters, and that is excellent news!

3749 sentences, 400,000 characters:
TheFultonCountyGrandJurysaidFridayaninvestigationofAtl anta'srecentprimaryelectionpr
yirregularitiestookplace.f Thejuryfurthersaidinterm-endpresentmentsthattheCityE
xecutiveCommittee,whichhadover-allchargeoftheelecti on,deservesthepraiseandthanksoft
annerinwhichtheelectionwasconducted ...

The Fulton County Grand Ju ry s aid Friday an investi gation of At l anta 's recent prim ary e lection produc ed no e videnc e that any ir regul ar it i es took place . Thejury further s aid in term - end present ment s thatthe City Ex ecutive Commit t e e ,which had over - all charg e ofthe e lection , d e serv e s the pra is e and than k softhe City of At l anta forthe man ner in whichthe e lection was conduc ted.

Select the lexicon $\mathcal{L}$ which minimizes the description length of the corpus $\mathcal{C}$. A lexicon $\mathcal{L}$ is a distribution $pr_{\mathcal{L}}$ over a subset of Σ^*. $\mathcal{L}$'s length is the length in bits in some specified format (the format matters!) and encoding. Any such distribution assigns a minimal encoding (up to trivial variants) to the corpus, and this encoding requires precisely $-logpr(\mathcal{C})$ bits. The description length of a corpus given lexicon $\mathcal{L}$ is defined as $|\mathcal{L}| - logpr_{\mathcal{L}}\mathcal{C}$: select the lexicon that minimizes this quantity (as best you can). $|\mathcal{L}|$ comes into the picture because if we assume $\mathcal{L}$ is expressed in a binary-encoded format in which no morphology is a prefix of another, this encoding induces a natural probability distribution, with $pr(l)$ proportional to $2^{|l|}$

piece	count	status
th	127,717	
he	119,592	
in	86,893	
er	81,899	
an	72,154	
re	67,753	
on	61,275	
es	59,943	
en	55,763	
at	54,216	
ed	52,893	
nt	52,761	
st	52,307	
nd	50,504	
ti	50,253	
to	48,233	
or	47,391	
te	44,280	
ea	41,913	
is	41,159	
ar	40,402	
of	40,296	
ha	39,922	
it	39,304	
ng	39,018	

Iteration number 2

Corpus cost: 43,593,516.07501816
Dictionary cost: 670.9952683596506
Break based Word Precision 0.2617 recall 0.9837
Token based Word Precision 0.0317 recall 0.1134
Type based Word Precision 0.7048 recall 0.0011

piece	count	status
the	51,775	
ou	35,767	
al	34,321	
and	29,107	
ing	27,883	
as	24,936	
ll	24,681	
ro	22,267	
om	21,073	
ic	20,855	
ec	20,185	
el	19,262	
le	18,278	
ly	17,604	
il	16,559	
ac	16,232	
se	16,115	
em	16,076	
co	15,381	
li	14,940	
wa	14,706	
ch	14,632	
ur	14,241	
be	14,224	
ion	13,762	

Corpus cost: 34,131,012.08884644
Dictionary cost: 842.2498702922143

Break based Word Precision 0.2917 recall 0.9642
Token based Word Precision 0.0624 recall 0.1965
Type based Word Precision 0.6538 recall 0.0012

Iteration number 3

piece	count	status
for	12,923	
ent	12,373	
id	12,290	
ow	11,441	
wh	11,121	
wi	10,302	
am	10,268	
that	10,003	
ad	9,995	
ver	9,969	
gh	9,840	
ld	9,582	
no	9,357	
was	9,295	
ation	9,188	
im	9,011	
ir	8,788	
ig	8,539	
ts	8,425	
ith	8,384	
ers	8,356	
ol	8,324	
ter	8,195	
ther	8,158	
ri	8,100	

Corpus cost: 30,164,461.41543184
Dictionary cost: 1,040.771864391648

Break based Word Precision 0.3125 recall 0.9626
Token based Word Precision 0.0770 recall 0.2260
Type based Word Precision 0.6000 recall 0.0014

Iteration number 4

piece	count	status
ve	8,192	
ab	8,034	
The	7,997	
with	7,681	
ce	7,577	
ay	7,506	
ag	7,467	
ofthe	7,456	
his	7,021	
us	6,810	
et	6,709	
pro	6,572	
ut	6,476	
ap	6,441	
,and	6,313	
su	6,260	
od	6,024	
un	6,006	
ep	5,973	
tion	5,972	
op	5,967	
ul	5,918	
po	5,798	
bu	5,766	
ain	5,712	

```
absen                ce
absen                ce                                       s
```

absen	t			
absen	t	ee		
absen	t	ee	ism	
absen	t	ee	s	
absen	t	ia		
abso	l	ut	e	
abso	l	ut	e	ly
abso	l	ut	e	s
abso	l	ut	i	on
abso	l	ut	i	s
abso	l	ut	i	s
abso	l	ut	i	ve
abso	l	ved		
abso	r	aka		
abso	r	b		
abso	r	b	able	
abso	r	b	e	d
abso	r	b	e	n
abso	r	b	e	n
abso	r	b	e	r
abso	r	b	e	r
abso	r	b	ing	
abso	r	b	s	
abso	r	pti	on	
abso	r	pti	ve	
abst	ain			
abst	ain	ed		
abst	ain	ing		
abst	e	miousness		
abst	e	ntion		
abst	inence			
abst	ract			
abst	ract	ed		
abst	ract	i	ng	
abst	ract	i	on	
abst	ract	i	on	s

abst	ract	ly	
abst	ract	s	
absurd			
absurd	i	s	m
absurd	i	s	t
absurd	i	s	t
absurd	i	t	ies
absurd	i	t	y
absurd	ly		

2.2 Sequitur: a non-probabilistic approach

2.3 MDL style approaches to word learning

2.3.1 What works well

2.3.2 What does not work well

Two serious problems: MDL is used primarily as a stopping criterion, and it does not do a good job of that. Even more importantly, the learning confuses word learning and phrase learning from the start; and slices off suffixes putting them together with following high frequency words. MDL is incapable of handling this problem as long as we stay with nothing but words.

3 Learning morphology

3.1 Class 2b: Zellig Harris

3.1.1 Harris 1955

3.1.2 Harris 196x

3.1.3 Hafer and Weiss

Hafer and Weiss 1974: Word segmentation by letter successor varieties

Information Storage and Retrieval 10 371-385

They point out the question of: which is the stem?

Four techniques:

1. SF threshold
2. Peak and plateau (or just peaks?")h make a cut at point k when SF(k) is >= SF(k-1) and also SF(k) >= SF(k+1).
3. Is the stem a free standing word?
4. Entropy of successor letter set

Best: 11 and 15.

1. SF threshold: worked so badly that they did not pursue it.
2. Both SF and PF reach "cutoff" (threshold). They don't tell us what the threshold used was! Other evidence suggests it was 5 and 17 for SF and PF respectively. Precision: 0.894, recall 0.511

3. Threshold exceeded by the sum of SF and PF. Precision 0.848, recall 0.565. They don't give the threshold, again!
4. Make breaks only after a " completed word" . Precision 0.904, recall 0.318.
5. The mirror image of 4: Useless.
6. Make breaks after a completed word, OR PF reaches threshold. Precision 0.778 recall 0.711.
7. SF at "peak and plateau" Precision: 0.486 recall 0.734. This works very badly at the beginning of words.
8. Both SF and PF are at "peak and plateau": Precision 0.787, recall 0.569.
9. Sum of SF and PF are at "peak and plateau" Recall: 0.828 precision: 0.441. This makes 3 times as many cuts as method 8, and 80
10. Make breaks after a complete word, also where PF is at "peak or plateau": works for FIND-ING, COMPUT-ER. Precision 0.484, Recall 0.937.
11. Hybrid of method 2 and 6: Make a cut when either of the following conditions is met:
    a) a. Left to right: completed word PF >= 5; OR
    b) b. SF >= 2 and PF >= 17

       Precision 0.91 recall 0.610

    Entropy-based techniques:
12. Left to right: completed word, PF-entropy > -3. Precision 0.72, recall 0.728.
13. Sum of entropies greater than threshold = 4, and also make break after complete word (or before complete word). Precision 0.609 recall 0.59.
14. Entropy version of 11: Make a cut when:
    a) Left to right completed word and predecessor entropy >= 0.8, OR
    b) Right to left completed word and successor entropy >= 1.0. Precision 0.874, recall 0.526.

15. Relaxation of 14: basically just a fudge, not interesting, I think. Cut as in 14, OR: if SF = 1 at point k, and EITHER SuccEntropy or PreEntropy >= 0.8 at k+1, cut at k+1.

3.2 Finding signatures

3.3 Learning morphology: Linguistica

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

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*) ...

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 ...

3.4 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.

3.5 Immediate issues: getting the morphology right

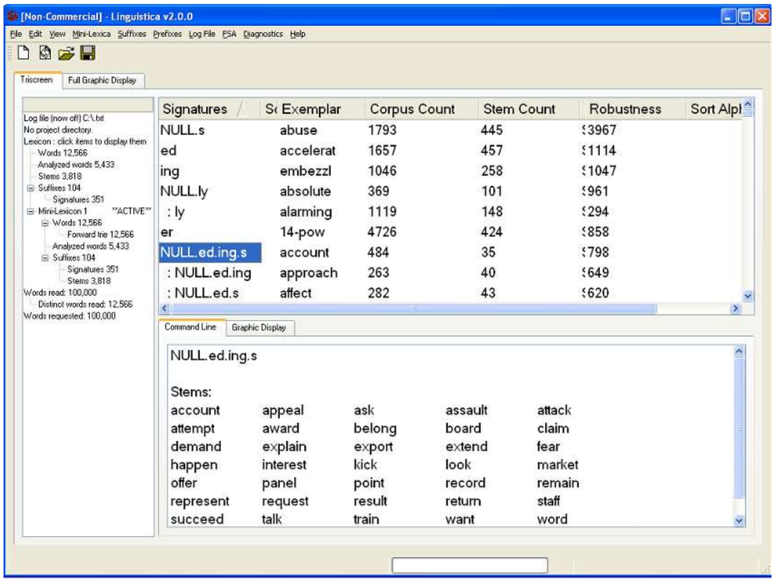
English: NULL - s - ed - ing - es- er - 's - e - ly - y - al - ers - in - ic - tion - ation - en - ies - ion - able - ity - ness - ous - ate - ent - ment - t (*burnt*) - ism - man - est - ant - ence - ated - ical - ance - tive - ating - less - d (*agreed*) - ted - men - a (*Americana, formul-a/-ate*)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

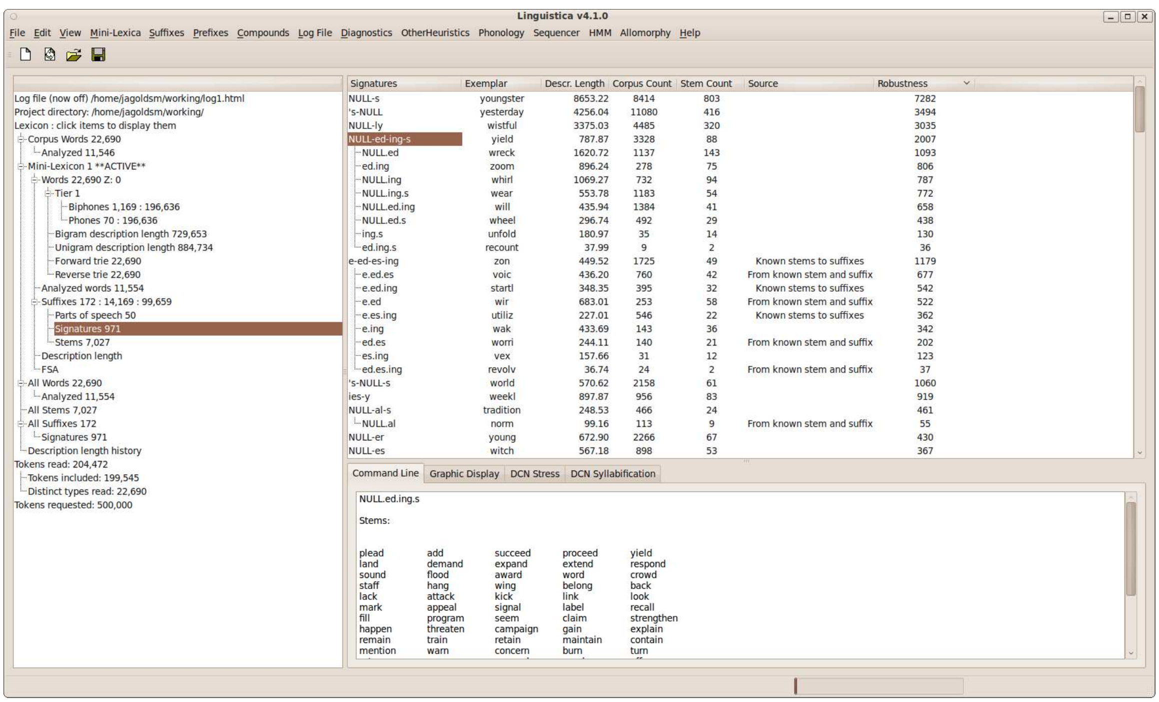
The key insight

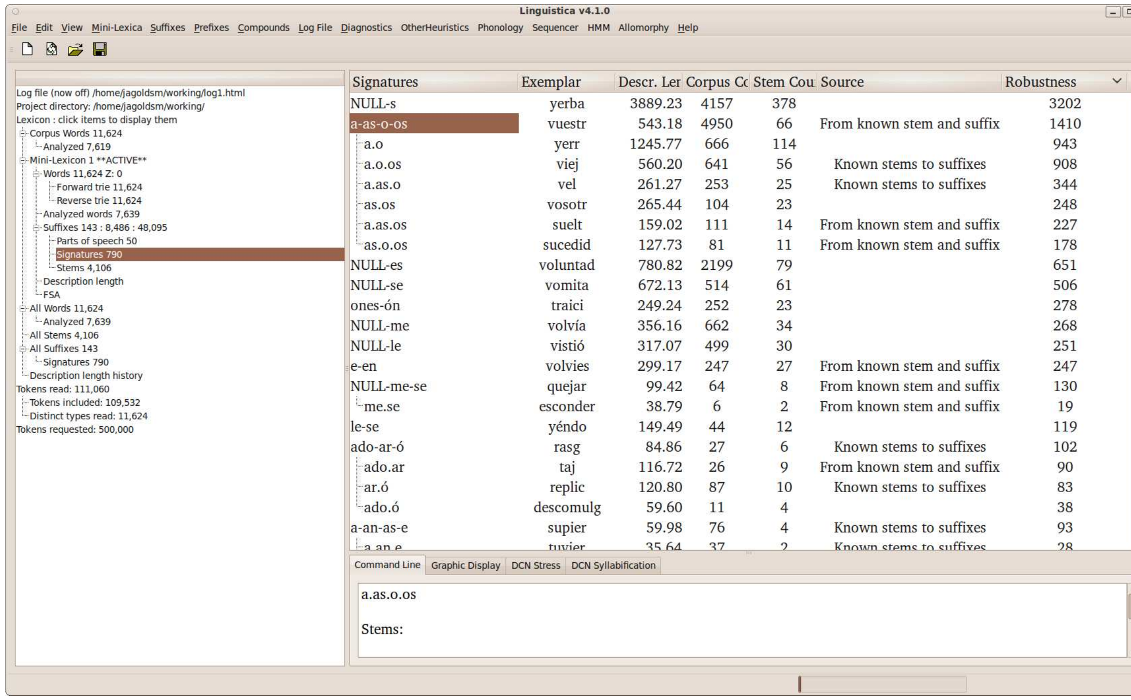
The overall complexity of the grammar, not how we get there.

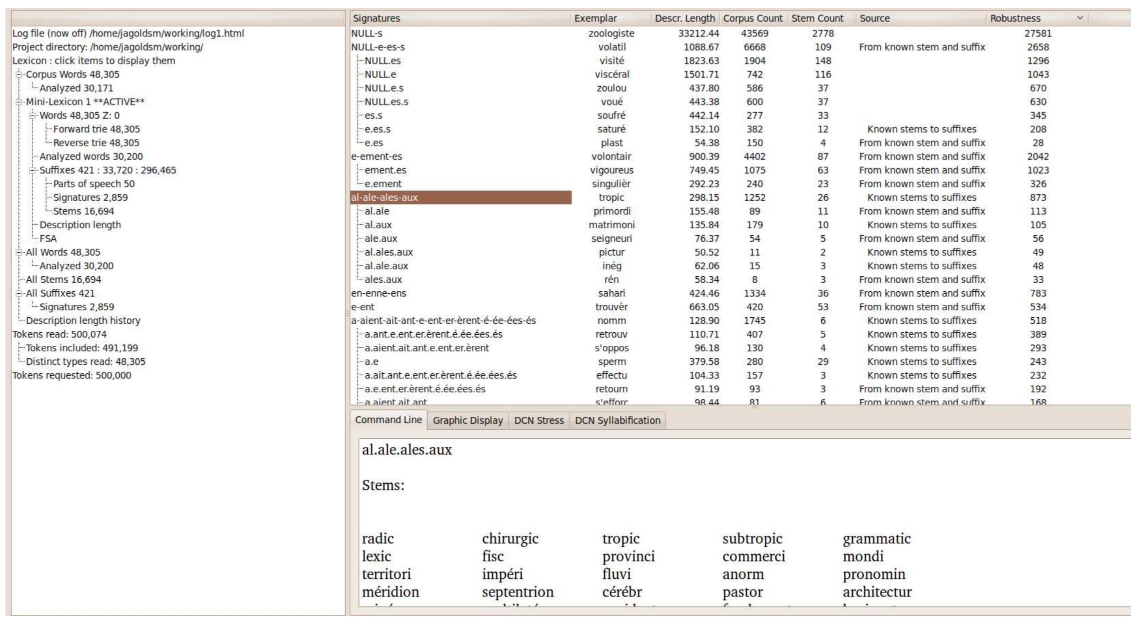
The key question: if we recognize that the learner needs something to be able to learn, what sorts of things can we give her that will in any way help solve the problem? What kinds of tools will actuall be useful? The purpose of the enterprise that we are engaged in is to answer that question.

3.5.1 Lxa 3 and 4 model

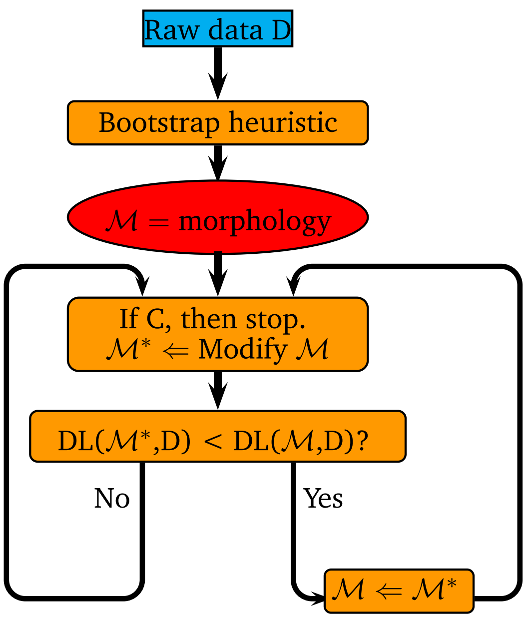





Linguistica v4.1.0
File Edit View Mini-Lexica Suffixes Prefixes Compounds Log File Diagnostics OtherHeuristics Phonology Sequencer HMM Allomorphy Help
Log file (now off) /home/jagoldsm/working/log1.html
Project directory: /home/jagoldsm/working/
Lexicon : click items to display them
Corpus Words 11,624
Analyzed 7,619
Mini-Lexicon 1 **ACTIVE**
Words 11,624 Z: 0
Forward trie 11,624
Reverse trie 11,624
Analyzed words 7,639
Suffixes 143 : 8,486 : 48,095
Parts of speech 50
Signatures 790
Stems 4,106
Description length
FSA
All Words 11,624
Analyzed 7,639
All Stems 4,106
All Suffixes 143
Signatures 790
Description length history
Tokens read: 111,060
Tokens included: 109,532
Distinct types read: 11,624
Tokens requested: 500,000
Signatures Exemplar Descr. Ler Corpus Cc Stem Cou Source Robustness
NULL-s yerba 3889.23 4157 378 3202
a-as-o-os vuestr 543.18 4950 66 From known stem and suffix 1410
a.o yerr 1245.77 666 114 943
a.o.os viej 560.20 641 56 Known stems to suffixes 908
a.as.o vel 261.27 253 25 Known stems to suffixes 344
as.os vosotr 265.44 104 23 248
a.as.os suelt 159.02 111 14 From known stem and suffix 227
as.o.os sucedid 127.73 81 11 From known stem and suffix 178
NULL-es voluntad 780.82 2199 79 651
NULL-se vomita 672.13 514 61 506
ones-ón traici 249.24 252 23 278
NULL-me volvía 356.16 662 34 268
NULL-le vistió 317.07 499 30 251
e-en volvies 299.17 247 27 From known stem and suffix 247
NULL-me-se quejar 99.42 64 8 From known stem and suffix 130
me.se esconder 38.79 6 2 From known stem and suffix 19
le-se yéndo 149.49 44 12 119
ado-ar-ó rasg 84.86 27 6 Known stems to suffixes 102
ado.ar taj 116.72 26 9 From known stem and suffix 90
ar.ó replic 120.80 87 10 Known stems to suffixes 83
ado.ó descomulg 59.60 11 4 38
a-an-as-e supier 59.98 76 4 Known stems to suffixes 93
Command Line Graphic Display DCN Stress DCN Syllabification
a.as.o.os
Stems:


Log file (now off) /home/jagoldsm/working/log1.html
Project directory: /home/jagoldsm/working/
Lexicon : click items to display them
Corpus Words 48,305
Analyzed 30,171
Mini-Lexicon 1 **ACTIVE**
Words 48,305 Z: 0
Forward trie 48,305
Reverse trie 48,305
Analyzed words 30,200
Suffixes 421 : 33,720 : 296,465
Parts of speech 50
Signatures 2,859
Stems 16,694
Description length
FSA
All Words 48,305
Analyzed 30,200
All Stems 16,694
All Suffixes 421
Signatures 2,859
Description length history
Tokens read: 500,074
Tokens included: 491,199
Distinct types read: 48,305
Tokens requested: 500,000
Signatures Exemplar Descr. Length Corpus Count Stem Count Source Robustness
NULL-s zoologiste 33212.44 43569 2778 27581
NULL-e-es-s volatil 1088.67 6668 109 From known stem and suffix 2658
NULL.es visité 1823.63 1904 148 1296
NULL.e viscéral 1501.71 742 116 1043
NULL.e.s zoulou 437.80 586 37 670
NULL.es.s voué 443.38 600 37 630
es.s soufré 442.14 277 33 345
e.es.s saturé 152.10 382 12 Known stems to suffixes 208
e.es plast 54.38 150 4 From known stem and suffix 28
e-ement-es volontair 900.39 4402 87 From known stem and suffix 2042
ement.es vigoureus 749.45 1075 63 From known stem and suffix 1023
e.ement singulièr 292.23 240 23 From known stem and suffix 326
al-ale-ales-aux tropic 298.15 1252 26 Known stems to suffixes 873
al.ale primordi 155.48 89 11 From known stem and suffix 113
al.aux matrimoni 135.84 179 10 Known stems to suffixes 105
ale.aux seigneuri 76.37 54 5 From known stem and suffix 56
al.ales.aux pictur 50.52 11 2 Known stems to suffixes 49
al.ale.aux inég 62.06 15 3 Known stems to suffixes 48
ales.aux rén 58.34 8 3 From known stem and suffix 33
en-enne-ens sahari 424.46 1334 36 From known stem and suffix 783
e-ent trouvèr 663.05 420 53 From known stem and suffix 534
a-aient-ait-ant-e-ent-er-èrent-é-ée-ées-és nomm 128.90 1745 6 Known stems to suffixes 518
a.ant.e.ent.er.èrent.é.ée.ées.és retrouv 110.71 407 5 Known stems to suffixes 389
a.aient.ait.ant.e.ent.er.èrent s'oppos 96.18 130 4 Known stems to suffixes 293
a.e sperm 379.58 280 29 Known stems to suffixes 243
a.ait.ant.e.ent.er.èrent.é.ée.ées.és effectu 104.33 157 3 Known stems to suffixes 232
a.e.ent.er.èrent.é.ée.ées.és retourn 91.19 93 3 From known stem and suffix 192
Command Line Graphic Display DCN Stress DCN Syllabification
al.ale.ales.aux
Stems:
radic chirurgic tropic subtropic grammatic
lexic fisc provinci commerci mondi
territori impéri fluvi anorm pronomin
méridion septentrion cérébr pastor architectur

Words	Stem	Mini-Lexicon 4	Mini-Lexicon 3	Mini-Lexicon 2	Mini-Lexicon 1
fully	ful				ly
fulton	fult				on
fumes	fum				es
function	func				tion
functional	func			tion	al
functionary	func			tion	ary
functions	func			tion	s
fundamental	fundament				al
fundamentalism	fundament		al	ism	
fundamentally	fundament			al	ly
fundamentals	fundament			al	s
fund-raiser	fund-rais				er
fund-raisers	fund-rais			er	s
fund-raising	fund-rais				ing



Boot-strapping heuristic for signatures, followed by a sequence of incremental heuristics, each applying until the MDL criterion is achieved

The qantity that we are trying to identify is letter-based recurrence: the product of the length times the number of occurrences. This is at the heart of de Marcken, and much of MDL (if the MDL model is chunk-based).

Low Hanging Fruit First:

Data: this text
Result: A morphology
m: a modification method in $\mathcal{M}ods$, which is universal;
M ← Bootstrap(data);
for *m* ∈ $\mathcal{M}ods$ **do**
    while *m improves the morphology* **do**
        M ← modified M;
    end
end

Algorithm 1: Linguistica 3-4: more specific

Data: this text
Result: A morphology
m: a modification method in $\mathcal{M}ods$, which is universal; they modify signatures;
M ← Bootstrap(data);
for *m* ∈ $\mathcal{M}ods$ **do**
    for *signature* σ ∈ *Signatures* **do**
        σ' ← m(M, σ, data);
        M' ← replace(M,σ, σ');
        if *DL(M', data)* < *DL(M,data)* **then**
            M ← M';
        end
    end
end

Algorithm 2: Linguistica 3-4: still more specific

Data: this text
Result: A morphology
m: a modification method in $\mathcal{M}ods$, which is a universal *list*; they modify signatures;
M $\leftarrow$ Bootstrap(data);
for $i \in (1 \ldots length(\mathcal{M}))$ **do**
    m = $\mathcal{M}ods_i$;
    for *signature* $\sigma \in$ *Signatures* **do**
        σ' $\leftarrow$ m(M, σ, data);
        M'= replace(M,σ, σ');
        if *DL(M', data)* < *DL(M,data)* **then**
            M $\leftarrow$ M';
        end
    end
end

Looking for affixes, there is a lot of noise (spurious structure) if we look at short words. So: we look only a longer words first, where we can get some reliable conclusions (meaning high precision, low recall).

It is an extremely bad error to look for solutions that solve the problem right from the beginning.

The solution only comes into focus as we proceed.

problems:

3.6 Class 3: On beyond Lxa 4: allomorphy, FSAs and paradigms