

A New Algorithm for Unsupervised Induction of Concatenative Morphology

Harald Hammarström

Department of Computing Science, Chalmers University of Technology, SE-412 96 Göteborg, Sweden



1 Introduction

The problem considered here is the induction of natural language morphology from unlabeled corpus text. More precisely, we shall take on the problem idealised to concatenative morphology and let the desired outcome, morphological paradigms, be represented as sets of affixes. The problem can thus be formulated as: given an arbitrary (modulo large-enough) natural languages corpus, output a set of paradigms that correspond well to the intuitions of human morphologists about salient conjugational patterns of the language in question.

We will present a new algorithm that proceeds in two steps: segmentation and paradigm induction. The segmentation part computes possible segmentations for each word, and the paradigm induction takes the segmentation information and tries to group affixes that belong together. The final outcome is a ranked list of paradigms. The algorithm does not take grammar or lexical semantics into account and has no language specific knowledge. It differs from earlier approaches in almost all aspects.

2 Segmentation

The three-step segmentation procedure is illustrated here with respect to suffixes. Together with some examples, we will define the scores Z , Z^W and Z^W , which is the final product. The idea of the algorithm can be seen in the definition of Z , namely, to look at increases in suffix frequency relative to segmentation point.

First some notation:

- W = the set (not bag!) of words in the corpus
- $s \triangleleft w = s$ is a suffix of the word w
- $f(s) = |\{w \in W \mid s \triangleleft w\}|$ = the number of words with suffix s
- $s_i(w) =$ the suffix of w that begins at position $1 \leq i \leq |w|$ (i.e. index starts at 0)
- $Q(w) = \{s_i(w) \mid i < \text{len}(w)\}$ the set of non-empty suffixes of s
- $S = \bigcup_{w \in W} Q(w)$ = the set all suffixes in the corpus
- $\text{Stems}(x) = \{z \mid zx \in W\}$ = the set of stems on which the suffix s occurs (in W)
- $P \text{ xor } s = P \setminus \{s\}$ if $s \in P$, $P \cup \{s\}$ if $s \notin P$.

2.1 Relative Frequency Fluctuations

Define Z as:

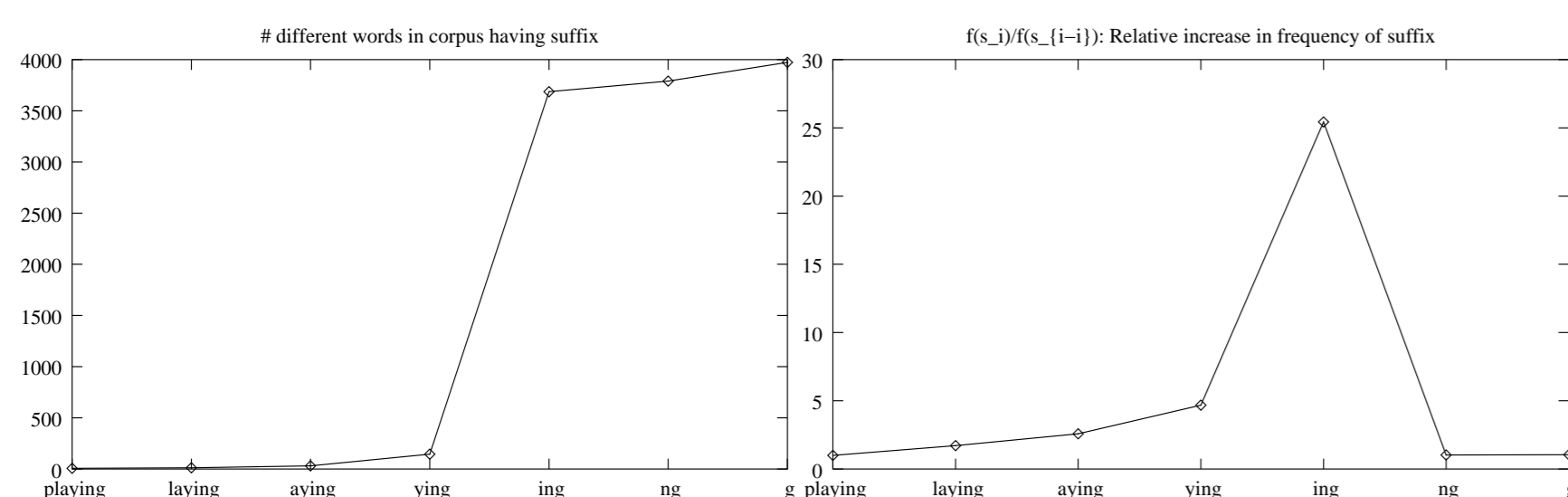
$$Z(s, w) = \begin{cases} 0 & \text{if not } s \triangleleft w \\ \frac{f(s)}{f(s_{i-1})} & \text{if } s = s_i(w) \text{ for some } i \end{cases} \quad (1)$$

Note that f , and hence Z , depends on W .

To understand what is going on, it is instructive to look at an example. For instance, using the Brown corpus [1] of English as our W , the values for “playing” are shown below:

s	playing	laying	aying	ying	ing	ng	g
$f(s)$	7	12	31	145	3687	3790	3973
$Z(s, \text{“playing”})$	1.00	1.70	2.60	4.70	25.40	1.00	1.40

The f -values mean that 7 (unique) words end in “playing” (“playing” itself, “displaying” etc), 12 words end in “laying” (those that end in “playing” and five more) and so on. The corresponding Z are exemplified relative to the word “playing”. The scores are also shown in graphical terms below.



The intuitive interpretation of the Z -value is intended to be: how good a segmentation is the suffix s of the word w .

2.2 Accumulate Suffix Scores

The natural continuation is to sum over the words in the corpus to get a suffix segmentation score that is not relative to some specific word. Define:

$$Z^W(s) = \sum_{w \in W} Z(s, w) \quad (2)$$

For the English Brown Corpus, this yields (30 top Z^W -s):

371161.0	s	60017.0	on	24125.0	o
238873.0	e	55564.0	er	23640.0	ts
148552.0	ed	53258.0	r	23409.0	le
139956.0	d	48669.0	l	22886.0	h
126897.0	y	47677.0	g	21321.0	ers
114298.0	ing	42072.0	ly	19891.0	ion
103841.0	t	35212.0	a	18497.0	en
87571.0	n	34110.0	ng	17701.0	m
84366.0	's	28341.0	al	15975.0	ted
72384.0	es	27057.0	an

The outcome ranked list looks like some kind of mixture between correct segmentation and frequent short character sequences.

2.3 Re-scale

As will be argued, we can in fact re-scale the mediocre Z^W to get a more reasonable segmentation. Re-scaling by $Z^W(s) = |s|^p \cdot Z^W(s)$ with $p = 2$ gives:

1028682.0	ing	168288.0	ly	109125.0	ate
594208.0	ed	159408.0	ations	108228.0	an
371145.0	s	143775.0	ted	97020.0	ies
337464.0	's	130960.0	able	94560.0	ts
326250.0	ation	116352.0	ated	81648.0	ically
289536.0	es	113364.0	al	81504.0	ment
238853.5	e	113280.0	ness	78669.0	led
222256.0	er	112264.0	ling	77900.0	ering
191889.0	ers	111132.0	ent	74976.0	er's
172800.0	ting	109725.0	ating

Although re-scaling by square length looks ad hoc, informal tests show that it is rather language independent. It also produces similar, yet better, results than some alternatives that explicitly adjust for probabilities of character n-grams. Moreover, some slight language-dependence of the re-scaling factor can be read off the ratio of suffixes/words¹. The table below shows such ratios computed on the bible (both Old and New Testament) for a few languages. Taking the re-scaling power as $p = \sqrt{\frac{|S|}{|W|}}$ yields a value close to 2 with the desired language-specific deviations.

Language	C	W	S	P	S / W	S /(S + P)
Adamawa fulfulde	597693	21644	65868	56793	0.036	0.574
Afrikaans	790547	15359	52264	48303	0.019	0.539
Albanian	271708	31152	98133	66969	0.040	0.682
Catalan	785233	34833	110097	87787	0.044	0.611
Cebuano	856959	27504	86191	92485	0.032	0.465
Danish	657202	25480	78687	63630	0.039	0.605
Dutch	757866	25768	86592	89884	0.034	0.481
English	784066	12997	39851	36667	0.017	0.542
Esperanto	684453	27196	88670	64537	0.040	0.658
Finnish	543919	54798	20725	164797	0.101	0.612
French	788740	24567	78471	68876	0.031	0.565
German	695990	20649	65960	56745	0.030	0.575
Greenlandic	391848	107891	657739	554195	0.275	0.585
Haitian creole	910093	7856	19831	21053	0.009	0.470
Hausa	693396	16592	50158	38956	0.024	0.624
Hindi	874447	20701	58654	53995	0.024	0.541
Hungarian	617345	63277	247151	173727	0.102	0.669
Icelandic	673010	35279	117688	94857	0.052	0.606
Italian	649257	48554	149872	114064	0.075	0.633
Kekchi	885711	22200	71753	82946	0.025	0.428
Latin	649257	48554	149872	114064	0.075	0.633
Lithuanian	593073	59009	193253	143853	0.099	0.643
Malayalam	511505	86245	398518	372120	0.169	0.534
Maori	967050	8412	23197	22721	0.009	0.510
Polish	805739	57966	198309	145183	0.072	0.651
Portuguese	759774	29403	95429	72078	0.039	0.637
Slovene	684076	43946	128702	86421	0.064	0.689
Spanish	701467	28816	92244	71071	0.041	0.628
Swahili	650541	46788	132163	187335	0.072	0.332
Swedish	784533	29095	93054	75207	0.037	0.605
Turkish	460372	56834	175819	143455	0.123	0.600
Zarma	816417	10704	30723	29901	0.013	0.514

The segmentation algorithm readily generalizes to prefixes as well as circumfixes (and even suffixes, prefixes and circumfixes all at the same time!). They type of affixation can be readily predicted from the ratio $|S|/(|S|+|P|)$, e.g. Swahili is predominantly prefixing. Infixes are harder because they don't have an end or a beginning of a word to be aligned to. Arbitrary infixes, such as infixes with “holes”, cannot be aligned either and are exponentially many in the length of each word.

3 Paradigm Induction

As mentioned, paradigm shall be taken to mean simply a set of suffixes. Of course, the paradigms we are interested in are not just any set of suffixes, but those which are linguistically motivated. At face value the problem looks very difficult:

- The number of theoretically possible paradigms is exponential (in the number of suffixes)
- Paradigms do not have to be disjoint (in real languages they are typically not)
- No matter what the size of the corpus few, if any, lemmas occur in all the forms of their paradigm.
- There is “noise”, i.e. stems exhibiting a genuine suffix also occur with “nonsense” suffixes by chance. For example bite, biting, bites occur, which is fine, but also the unrelated bitonic. To the computer it might look as if -onic can be added to other stems that also have -e, -ing etc. Or noise stems, such as e.g. sing because the corresponding sed that other stems taking -ing frequently exhibit, is illegal. Throughout, the algorithms is supposed to have no access to a lexicon or some other resource that would tell us what is a “real” stem and what is not.

We will discuss an approach based on the following heuristic:

Suffixes s_1, s_2 belong to paradigm P iff in a corpus of text s_1 and s_2 tend to occur on the same stems.

First we shall explain a metric for testing paradigms for linguistic “motivatedness”. Thereafter we shall turn to the question of how to come up with such hypotheses in the first place, given that enumerating all is prohibitive.

3.1 Testing Paradigm Hypotheses

First, define quotient lists: $H_x(y) : S \rightarrow [0, 1]$ as:

$$H_x(y) = \frac{|\{z \mid \exists z(z \in \text{Stems}(x) \wedge zy \in W)\}|}{|\text{Stems}(x)|} \quad (3)$$

Example:

y	$H_{\text{ing}}(y)$	y	$H_{\text{ed}}(y)$
ed	0.59	ing	0.42
a	0.41	n	0.33
s	0.25	e	0.21
e	0.24	s	0.20
es	0.19	es	0.17
er	0.12	er	0.08
ers	0.10	ion	0.07
ion	0.07	ers	0.05
y	0.05	y	0.04
ings	0.05	ions	0.03
ions	0.03	ation	0.03
in	0.03	able	0.02
ation	0.03	ings	0.02
's	0.03	's	0.02
ingly	0.03	or	0.02
or	0.02	in	0.01
...

The quotients mean e.g. that ed shows up on 59% of the stems on which ing occurs, the empty suffix (“”) on 41% of the ing -stems etc. Now, the idea is: For a paradigm P , sum the quotient lists for the members of P , and see how high up in the ranking they show up.

$$V_P(y) = \sum_{x \neq y \in P} H_x(y)$$

The $x \neq y$ is just there so that the y s that are also in P don't get an “extra” 1, since $H_x(x) = 1$ regardless of the data. The rank is just y sorted on highest $V_P(y)$. Example:

{ing,ed},s,er	{ing,ation,s,xt}		
"	1.51	"	1.83
ed	0.94	ed	1.10
ing	0.78	st	0.80
s	0.73	e	0.64
e	0.61	s	0.64
es	0.48	nd	0.60
ers	0.47	sted	0.60
er	0.24	sts	0.60
y	0.18	nder	0.60
's	0.17	nding	0.60
ion	0.15	nds	0.60
er's	0.13	nded	0.60
d	0.12	xts	0.60
ly	0.12	ar	0.42
ings	0.12	ll	0.40
in	0.10	aring	0.40
ered	0.10	nt	0.40
...
[2, 1, 0, 3, 7]		[32, 662, 661, 87352]	

In the case to the left the member of P , as predicted, show up high on the list (places [2, 1, 0, 3, 7]). In the case to the right, the members of P show up very far down the list because, even though individually they are very common suffixes, they don't tend to occur of the same stems.

Finally, we can define the paradigm quality metric $VI(P)$ as the sum of ranks of the members of P are, compared to the optimal sum (which depends on $|P|$ and is $0 + \dots + |P| - 1$):

$$VI(P) = \frac{|P|(|P| - 1)}{2 \sum_{x \in P} \text{rank}_P(x)} \quad (4)$$

3.2 Growing Paradigms

The idea here is to start with paradigm containing one suffix and try to include more suffixes as long as the VI -score increases. We also check if the VI -score is even better served by kicking one suffix out. Thus this is a typical greedy gradient search algorithm:

$$G(P) = \text{argmax}_{P \subseteq \{P \text{ xor } s \mid s \in S\}} VI(P) \quad (5)$$

$$G^*(P) = \begin{cases} P & \text{if } G(P) = P \\ G^*(G(P)) & \text{if } G(P) \neq P \end{cases} \quad (6)$$

Examples:

P	$VI(P)$	P	$VI(P)$
{ation}	0.00	{xt}	0.00
{ated, ation}	0.14	{xt, n}	0.04
{ate, ated, ation}	0.40	{xt, n, ns}	0.12
{ate, ated, ating, ation}	0.75	{n, ns}	0.55
{ate, ated, ating, ation, ations}	1.00

To the left the paradigm grows until it reaches to a maximum. To the right it grows from {xt} but later opts to kick out the original xt.

3.3 Final Definition of Output

We can now define the output paradigms as the set of paradigms converged to when trying to grow all possible single suffixes. They are then ranked by VI -score times the average “suffixness” score for their members.

$$A = \{G^*(\{s\}) \mid s \in S\}$$

$$R(P) = \frac{VI(P)}{|P|} \sum_{s \in P} Z^W(s) \quad (7)$$

4 Example Results

English					
n	n	n	n	n	n
1622496.3	1379391.7	894100.2	540701.1	469880.4	...
e	ed	able	ated	ies	...
ed	er	ably	ates	y	...
es	ers	ed	ating	ying	...
ing	ing	ing	ation		...
s	s	s	ations		...

Polish					
n	n	n	n	n	n
1217370.1	1205087.9	1078624.3	...		
a	na	a	...		
ac'	na,	a	...		
acie	ne	ach	...		
aja,	nego	ami	...		
al'	nej	e	...		
ali	nie	em	...		
al/a	niejsze	om	...		
al/o	niejszego	owe	...		
asz	niejszy	owej	...		
am	niejszych	owi	...		
ane	niejszym	owych	...		
anie	ny	u	...		
aniem	nych	y	...		
aniu	nym		...		
ania			...		
ano			...		
asz			...		

5 Discussion and Future Work

The results are promising but there still remains a lot of work:

- To prove properties of the algorithms
- Postprocess and filter output paradigms
- Robust approach to stacked suffixes
- Proper evaluation
- Investigate Minimum Description Length approaches to find a cut-off point in the ranked list of output paradigms (cf. e.g. [2])
- Investigate whether part-of-speech information can be beneficial and/or sufficient to separate derivational from inflectional morphology

References

- [1] Francis, N.W., Kucera, H.: Brown corpus. Department of Linguistics, Brown University, Providence, Rhode Island (1964) 1 million words.
- [2] Goldsmith, J.: Unsupervised learning of the morphology of natural language. Computational Linguistics 27 (2001) 153–198

¹It is slightly idealized to think of these ratios as measuring morphology since they may well reflect orthographic practices (such as writing compounds together in Swedish).