Désuffixation – Algorithme de Porter

Thierry Lecroq

Université de Rouen
FRANCE

But : avoir la même forme de base pour des mots de la même famille

Conçu pour l'anglais, adaptable à d'autres langues (français, ...)

Plan

Consonne ou voyelle

- v représente une voyelle (y est considéré comme une voyelle s'il est précédé par une consonne)
- c représente une consonne
- V représente une suite de voyelles
- C représente une suite de consonnes

De la mesure

Un mot en anglais peut être de l'une des 4 formes suivantes :

- $CVCV \cdots C$
- $CVCV \cdots V$
- $VCVC \cdots C$
- $VCVC \cdots V$

ce qui peut se représenter par

$$[C]VCVC \cdots [V]$$

ou

$$[C](VC)^m[V]$$

où m est appelée la mesure d'un mot.

Mesurons la mesure

- $m = 0$: `tree`, `by`
- $m = 1$: `trouble`, `oats`, `trees`, `ivy`
- $m = 2$: `troubles`, `private`, `oaten`, `orrery`

Règles

Les règles de désuffixation sont exprimées sous la forme
(condition) $S_1 \rightarrow S_2$
ce qui signifie que si un mot se termine par S_1 et que le préfixe satisfait la condition alors le suffixe S_1 est remplacé par S_2

UNIVERSITÉ DE ROUEN

Condition

- $*$e : le préfixe se termine par la lettre e
- $*v*$: le préfixe contient une voyelle
- $*d$: le préfixe se termine par une consonne doublée
- $*o$: le préfixe se termine par cvc où le second c n'est ni w, ni x, ni y

Il est possible d'utiliser des opérateurs booléens : et, ou, non

À chaque étape seule la règle capturant le plus long S_1 s'applique

Plan

Étape 1a

sses	→	ss	caresses	→	caress
ies	→	i	ponies	→	poni
			ties	→	ti
ss	→	ss	caress	→	caress
s	→		cats	→	cat

Étape 1b

$(m > 0)$eed	$\rightarrow$	ee	feed	$\rightarrow$	feed
			agreed	$\rightarrow$	agree
$(*v*)$ed	$\rightarrow$		plastered	$\rightarrow$	plaster
			bled	$\rightarrow$	bled
$(*v*)$ing	$\rightarrow$		motoring	$\rightarrow$	motor
			sing	$\rightarrow$	sing
at	$\rightarrow$	ate	conflat(ed)	$\rightarrow$	conflate
bl	$\rightarrow$	ble	troubl(ed)	$\rightarrow$	trouble
iz	$\rightarrow$	ize	siz(ed)	$\rightarrow$	size

$(*d$ et non $(*$l ou $*$s ou $*$z$)) \rightarrow$ lettre non doublée

hopp(ing) $\rightarrow$ hop

tann(ed) $\rightarrow$ tan

fall(ing) $\rightarrow$ fall

hiss(ing) $\rightarrow$ hiss

fizz(ed) $\rightarrow$ fizz

$(m = 1)$ et $*o$	$\rightarrow$	e	fail(ing)	$\rightarrow$	fail
			(filing)	$\rightarrow$	file

Étape 1c

$(*v*)$y $\longrightarrow$ i happy $\longrightarrow$ happi

sky $\longrightarrow$ sky

Étape 2

$(m > 0)$ational	$\rightarrow$	ate	relational	$\rightarrow$	relate
$(m > 0)$tional	$\rightarrow$	TION	conditional	$\rightarrow$	condition
			rational	$\rightarrow$	rational
$(m > 0)$enci	$\rightarrow$	ence	valenci	$\rightarrow$	valence
$(m > 0)$anci	$\rightarrow$	ance	hesitanci	$\rightarrow$	hesitance
$(m > 0)$izer	$\rightarrow$	ize	digitizer	$\rightarrow$	digitize
$(m > 0)$abli	$\rightarrow$	able	conformabli	$\rightarrow$	conformable
$(m > 0)$alli	$\rightarrow$	al	radicalli	$\rightarrow$	radical
$(m > 0)$entli	$\rightarrow$	ent	differentli	$\rightarrow$	different
$(m > 0)$eli	$\rightarrow$	e	vileli	$\rightarrow$	vile
$(m > 0)$ousli	$\rightarrow$	ous	analogousli	$\rightarrow$	analogous
$(m > 0)$ization	$\rightarrow$	ize	vietnamization	$\rightarrow$	vietnamize
$(m > 0)$ation	$\rightarrow$	ate	predication	$\rightarrow$	predicate
$(m > 0)$ator	$\rightarrow$	ate	operator	$\rightarrow$	operate
$(m > 0)$alism	$\rightarrow$	al	feudalism	$\rightarrow$	feudal
$(m > 0)$iveness	$\rightarrow$	ive	decisiveness	$\rightarrow$	decisive
$(m > 0)$fulness	$\rightarrow$	ful	hopefulness	$\rightarrow$	hopeful
$(m > 0)$ousness	$\rightarrow$	ous	callousness	$\rightarrow$	callous
$(m > 0)$aliti	$\rightarrow$	al	formaliti	$\rightarrow$	formal
$(m > 0)$iviti	$\rightarrow$	ive	sensitiviti	$\rightarrow$	sensitive
$(m > 0)$biliti	$\rightarrow$	ble	sensibiliti	$\rightarrow$	sensible

Étape 3

$(m > 0)$icate	$\rightarrow$	ic	triplicate	$\rightarrow$	triplic
$(m > 0)$ative	$\rightarrow$		formative	$\rightarrow$	form
$(m > 0)$alize	$\rightarrow$	al	formalize	$\rightarrow$	formal
$(m > 0)$iciti	$\rightarrow$	ic	electriciti	$\rightarrow$	electric
$(m > 0)$ical	$\rightarrow$	ic	electrical	$\rightarrow$	electric
$(m > 0)$ful	$\rightarrow$		hopeful	$\rightarrow$	hope
$(m > 0)$ness	$\rightarrow$		goodness	$\rightarrow$	good

Étape 4

$(m > 1)$al	$\rightarrow$	revival	$\rightarrow$	reviv
$(m > 1)$ance	$\rightarrow$	allowance	$\rightarrow$	allow
$(m > 1)$ence	$\rightarrow$	inference	$\rightarrow$	infer
$(m > 1)$er	$\rightarrow$	airliner	$\rightarrow$	airlin
$(m > 1)$ic	$\rightarrow$	gyroscopic	$\rightarrow$	gyroscop
$(m > 1)$able	$\rightarrow$	adjustable	$\rightarrow$	adjust
$(m > 1)$ible	$\rightarrow$	defensible	$\rightarrow$	defens
$(m > 1)$ant	$\rightarrow$	irritant	$\rightarrow$	irrit
$(m > 1)$ement	$\rightarrow$	replacement	$\rightarrow$	replac
$(m > 1)$ment	$\rightarrow$	adjustment	$\rightarrow$	adjust
$(m > 1)$ent	$\rightarrow$	dependent	$\rightarrow$	depend
$(m > 1$ et $(*$s ou $*$ t$))$ion	$\rightarrow$	adoption	$\rightarrow$	adopt
$(m > 1)$ou	$\rightarrow$	homologou	$\rightarrow$	homolog
$(m > 1)$ism	$\rightarrow$	communism	$\rightarrow$	commun
$(m > 1)$ate	$\rightarrow$	activate	$\rightarrow$	activ
$(m > 1)$iti	$\rightarrow$	angulariti	$\rightarrow$	angular
$(m > 1)$ous	$\rightarrow$	homologous	$\rightarrow$	homolog
$(m > 1)$ive	$\rightarrow$	effective	$\rightarrow$	effect
$(m > 1)$ize	$\rightarrow$	bowdlerize	$\rightarrow$	bowdler

Étape 5a

$(m > 1)$e	$\longrightarrow$	probate	$\longrightarrow$	probat
		rate	$\longrightarrow$	rate
$(m > 1$ et non $* o)$e	$\longrightarrow$	cease	$\longrightarrow$	cease

Étape 5b

$(m > 1$ et $*d$ et $*\texttt{l}) \longrightarrow$ lettre non doublée

`controll` $\longrightarrow$`control`

`roll`$\longrightarrow$`roll`

generalizations
→generalization étape 1
→generalize étape 2
→general étape 3
→gener étape 4

oscillators
→oscillator étape 1
→oscillate étape 2
→oscill étape 4
→oscil étape 5

sur 10 000 mots

étape 1	3597
étape 2	766
étape 3	327
étape 4	2424
étape 5	1373
non réduits	3650

6370 formes réduites

UNIVERSITÉ DE ROUEN

Référence

M. F. Porter
An Algorithm for Suffix Stripping
Program, **14**(3), 130–137, 1980