Expresiones regulares

Fernando Carranza
fernandocarranza86@gmail.com

Miércoles 20/03/2024

Estructura

En esta charla vamos a tratar de cumplir los siguientes objetivos:

1. Hacer un acercamiento al lugar de las expresiones regulares en el marco de la teoría de los lenguajes
2. Presentar las convenciones notacionales de las expresiones regulares
3. Revisar las funcionalidades de la librería de Python re, para la manipulación de textos mediante expresiones regulares

Teorema de Borges

Teorema de Borges

Un mono tipeando teclas al azar en una máquina de escribir tipeará las obras completas de Shakespeare si le damos suficiente tiempo

(Pablo Groisman. Te regalo un teorema. TantaAgua)

La Biblioteca Total

Todo estará en sus ciegos volúmenes. Todo: la historia minuciosa del porvenir, Los egipcios de Esquilo, el número preciso de veces que las aguas de Ganges han reflejado el vuelo de un halcón, el secreto y verdadero nombre de Roma, (...) Todo, pero por una línea razonable o una justa noticia habrá millones de insensatas cacofonías, de fárragos verbales y de incoherencias. Todo, pero las generaciones de los hombres pueden pasar sin que los anaqueles vertiginosos -los anaqueles que obliteran el día y en los que habita el caos- les hayan otorgado una página tolerable.

(Jorge Luis Borges. La Biblioteca total.)

Corpus

- En términos formales, el corpus que esecribió el mono es en esencia una larga cadena de símbolos.
- Ese mono es capaz de escribir cosas muy complejas, pero es incapaz de distinguir esas cosas de otras que son fárrragos verbales.

- Supongamos que queremos extraer ciertos datos de ese corpus. Nuestro deseo puede representarse en términos de un *problema*.

- Supongamos que queremos extraer ciertos datos de ese corpus. Nuestro deseo puede representarse en términos de un *problema*.
- A cada problema, un algoritmo.

Algunos problemas para nuestro corpus

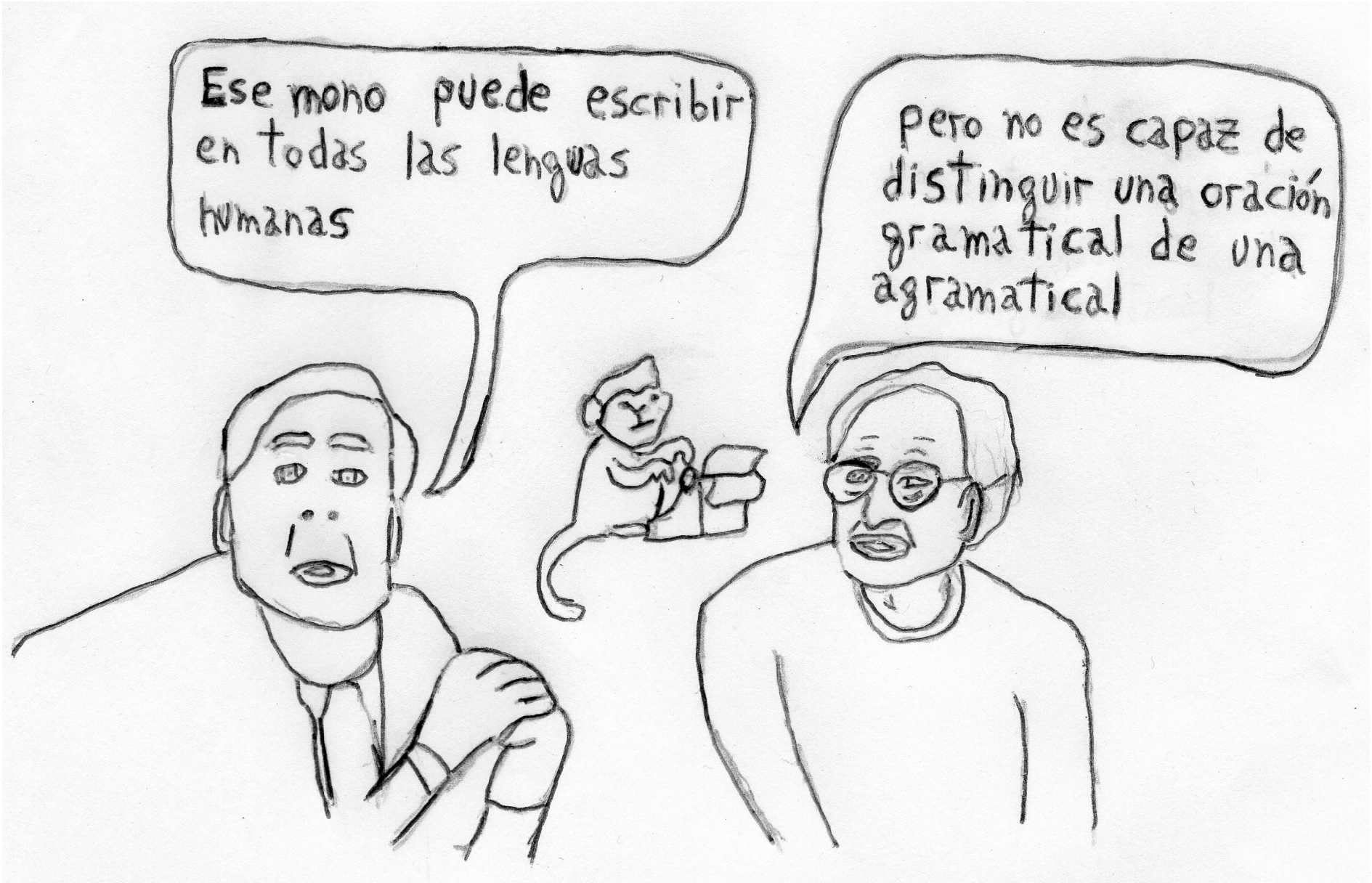
- ¿Cuáles son los guarismos escritos en números arábigos?
- ¿Cuáles son aquellas cadenas que expresan operaciones aritméticas?
- ¿Cuáles son aquellas operaciones aritméticas bien hechas y cuáles no?
- ¿Cuáles son las obras de Shakespeare?
- ¿Cuáles son los textos escritos en español?

Algunos problemas para nuestro corpus

- ¿Cuáles son los guarismos escritos en números arábigos?
- ¿Cuáles son aquellas cadenas que expresan operaciones aritméticas?
- ¿Cuáles son aquellas operaciones aritméticas bien hechas y cuáles no?
- ¿Cuáles son las obras de Shakespeare?
- ¿Cuáles son los textos escritos en español?

No todos los problemas acarrean la misma *complejidad*. La complejidad del algoritmo para solucionarlo dependerá de la complejidad del problema.

Un chiste nerd



Un poco de terminología

- Denominamos *alfabeto* al conjunto no vacío de símbolos discretos. En el caso de los textos digitales, estos serán caracteres pertenecientes a alguna fuente de codificación. El alfabeto se designa convencionalmente con la letra Σ

Un poco de terminología

- Denominamos *alfabeto* al conjunto no vacío de símbolos discretos. En el caso de los textos digitales, estos serán caracteres pertenecientes a alguna fuente de codificación. El alfabeto se designa convencionalmente con la letra Σ
- La concatenación de símbolos (iguales o diferentes) de un alfabeto Σ se conoce con el nombre de cadena.

Σ^*

Σ^* equivale al conjunto de todas las cadenas posibles que se pueden obtener a partir de un alfabeto Σ.

Lenguaje

- Denominamos lenguaje a un conjunto de cadenas que está incluido en Σ^*.

Definir un lenguaje por extensión consiste en nombrar uno a uno todos sus miembros. Esto, por supuesto, es un método de definición de conjuntos totalmente inviable para lenguajes formados por infinitos elementos. En su lugar, hay que recurrir a un tipo de definición intensional.

- Todo problema (e.g., ¿cuáles cadenas pertenecen al español?, ¿cuáles son las obras de Shakespeare?, etc.) se puede representar en términos de un lenguaje (e.g. el conjunto de cadenas del español, el conjunto de las obras de Shakespeare, etc.) y viceversa.
- Así como hay problemas de diversa complejidad, los respectivos lenguajes también se clasifican según su complejidad.

- Las expresiones regulares son un sistema de notación que permite caracterizar la familia de lenguajes más sencillos: los llamados lenguajes regulares.

- Las expresiones regulares son un sistema de notación que permite caracterizar la familia de lenguajes más sencillos: los llamados lenguajes regulares.
- Esa sencillez formal acarrea también un bajo costo de procesamiento, lo que vuelve a las expresiones regulares muy eficientes computacionalmente.

- Las expresiones regulares son un sistema de notación que permite caracterizar la familia de lenguajes más sencillos: los llamados lenguajes regulares.
- Esa sencillez formal acarrea también un bajo costo de procesamiento, lo que vuelve a las expresiones regulares muy eficientes computacionalmente.
- Las expresiones regulares no sirven, entonces, para caracterizar conjuntos de cadenas más complejos, como el conjunto de las oraciones gramaticales del español o el conjunto de operaciones aritméticas bien resueltas. Pero permiten identificar lenguajes más sencillos, como el conjunto de las cadenas de caracteres alfanuméricos separados por puntos, el conjunto de las cadenas que tienen el formato de direcciones de correo electrónico, etc.

- Las expresiones regulares son un sistema de notación que permite caracterizar la familia de lenguajes más sencillos: los llamados lenguajes regulares.
- Esa sencillez formal acarrea también un bajo costo de procesamiento, lo que vuelve a las expresiones regulares muy eficientes computacionalmente.
- Las expresiones regulares no sirven, entonces, para caracterizar conjuntos de cadenas más complejos, como el conjunto de las oraciones gramaticales del español o el conjunto de operaciones aritméticas bien resueltas. Pero permiten identificar lenguajes más sencillos, como el conjunto de las cadenas de caracteres alfanuméricos separados por puntos, el conjunto de las cadenas que tienen el formato de direcciones de correo electrónico, etc.
- Aplicado como problema, una expresión regular nos permite identificar cadenas de determinada naturaleza en el marco de un corpus, como el que elaboró el mono.

Primeramente, cualquier secuencia de caracteres del alfabeto $\Sigma = \text{ASCII}$, escapando aquellos caracteres que oficien de operadores, es una expresión regular r que matcheará con esa secuencia de caracteres.

Code	Char	Code	Char	Code	Char	Code	Char
0	NUL	32	SPACE	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

Figura: Tomado de `https://catonmat.net/my-favorite-regex`

En la bibliografía teórica se utiliza la función interpretación $[\![\alpha]\!]$ para evaluar el significado de la expresión regular α (ver Wintner 2010).
Por ejemplo,

- $[\![\text{abc}]\!] = \{\text{w}|$ w es una cadena formada por la concatenación de a, b y c$\} = \text{abc}$
- $[\![\text{Fer}]\!] = \{\text{w}|$ w es una cadena formada por la concatenación de F, e y r$\} = \text{Fer}$
- y así sucesivamente.

Paréntesis y estrella de Kleene

- ()
  Para delimitar bloques de cadenas
- *
  cero o más ocurrencias del caracter o bloque anterior
  Por ejemplo, $[\![a^*]\!] = \{w|$ w es una cadena formada por la concatenación de cero o más ocurrencias de a$\}$

$[\![ab^*]\!]$	
$[\![(ab)^*]\!]$	

+ y ?

- +
  una o más ocurrencias del caracter o bloque anterior
  Por ejemplo, $[\![a+]\!] = \{w|$ w es una cadena formada por la concatenación de una o más ocurrencias de a$\}$
- ?
  una o ninguna ocurrencia del caracter o bloque anterior
  Por ejemplo, $[\![ab?]\!] = \{a, ab\}$

$[\![ab+]\!]$	
$[\![(ab)+]\!]$	
$[\![(a?b)+]\!]$	

- .
  cualquier caracter excepto cambio de línea
  Por ejemplo, $[\![a.]\!] = \{aa, ab, ac, ad...\}$)
- {n}
  n cantidad de ocurrencias del caracter o bloque anterior
  Por ejemplo, $[\![a\{3\}]\!] = \{aaa\}$)
- {n,m}
  entre n y m cantidad de ocurrencias del caracter o bloque anterior
  (cuando se usa para matchear, matchea con el menor número posible)
  Por ejemplo, $[\![a\{3,6\}]\!] = \{aaa, aaaa, aaaaa, aaaaaa\}$)

$[\![Achi\{2\}s]\!]$	
$[\![(Bla)\{3\}]\!]$	
$[\![(Bla)\{2,4\}]\!]$	

Disyunción y rango

- [] = disyunción entre lo que aparezca adentro ($[\![$[abcd]$]\!]$ = {a,b,c,d})
- [x-y] = rango de caracteres que aparecen entre x e siguiendo el orden de ASCII ($[\![$[a-d]$]\!]$ = {a,b,c,d})
- [^x] = expresiones que excluyan x.

El rango mayor que se puede obtener es [-~]

Abreviaturas

- \s = espacio en blanco
- \S = cualquier caracter que no sea un espacio en blanco
- \d = cualquier dígito.
- \D = cualquier símbolo que no sea un dígito
- \w = cualquier caracter alfanumérico

Establecer las equivalencias en términos de rangos

Abreviatura	rango
\d	
\w	

Expresiones target	Expresión regular
Edad (en arábigos)	
Estructura de sílaba	
Perífrasis terminativas	

Páginas para probar expresiones regulares:

- `https://regex101.com/`
- `https://regexr.com/`

Bibliografía I

Hopcroft, J. E., Motwani, R., y Ullman, J. D. (2006). *Automata theory, languages, and computation*. Addison-Wesley, Boston, Massachusetts.

Jurafsky, D. y Martin, J. H. (2024). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Draft disponible.

Wintner, S. (2010). Formal language theory. En Clark, A., Fox, C., y Lappin, S., editores, *The Handbook of Computational Linguistics and Natural Language Processing*, pp. 11–42. Willey Blakwell, Massachusetts.