Natural Language Processing (NLP)

2 — Caractères et alphabets

Georges-André Silber

2023/2024

École des mines de Paris


ER 1190FL
Bigfoot™
SNOW PUSHER
with Ergonomic Oversize D-Grip Cushion Handle
REVENDEUR DE DROGUE DE NEIGE
avec grande poignée rembourrée et ergonomique D-Grip
25" blade/lame
• COMFORTABLE oversized D-Grip handle and cushion comfort grip
• DURABLE, lightweight poly resin blade with wear strip for long blade life
• LIGHTWEIGHT metal handle for easy handling of snow
• Poignée fermée CONFORTABLE surdimensionnée et surface d'appui confort
• Lame DURABLE légère en résine poly avec segment d'usure pour une longue durée de vie
• Manche métallique LÉGER pour ramasser facilement la neige
MADE IN THE USA
EMSCO GROUP
0 72358 01190 8

Données sources du NLP : caractères et alphabets

Pourquoi s'intéresser aux caractères ?

- Donnée de base du NLP : caractère ∈ alphabet
- Qualité des données primordiale pour le NLP
- En 2023, le Mojibake existe toujours
- Diversité des caractères dans les langues humaines
- Traitements plus compliqués quand on ne traite pas de l'anglais

Représentation des caractères

- Sur un ordinateur, un caractère est un entier positif.
- Cet entier positif représente un caractère dans une table donnée.
- Si on ne connaît pas la table, on ne peut pas connaître le caractère.
- 1 caractère : 5 bits, 7 bits, 1, 2, 3 ou 4 octets (Unicode).

Émile Baudot (1845–1903) : code Baudot

- Code sur 5 bits (1878)
- Téléscripteur, Telex
- Baud (Bd)
  - unité de modulation
  - nb de symboles par *s*
- *Stateful*

Code Baudot (US TTY)

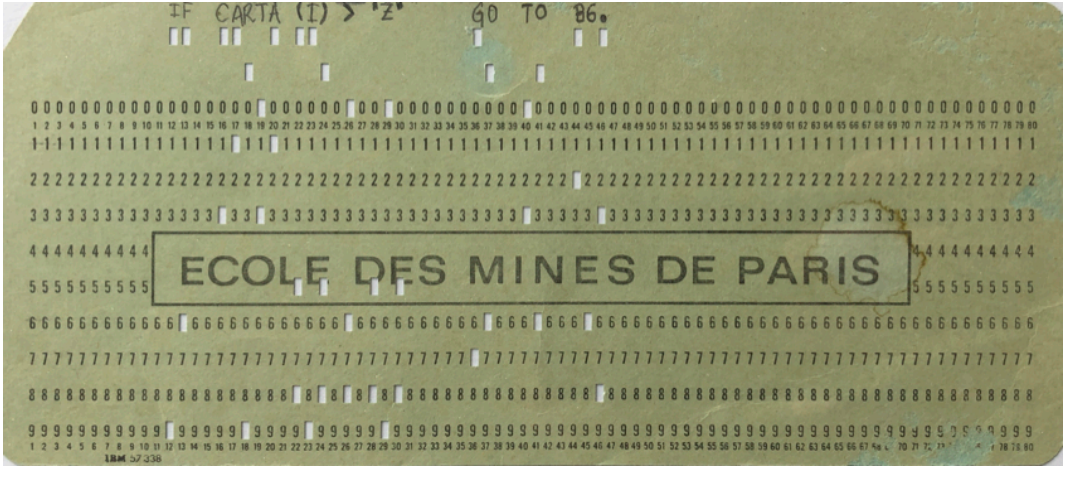
00	01	02	03	04	05	06	07
NUL	E 3	LF	A -	SP	S '	I 8	U 7
08	09	0A	0B	0C	0D	0E	0F
CR	D ENQ	R 4	J BEL	N ,	F !	C :	K (
10	11	12	13	14	15	16	17
T 5	Z +	L)	W 2	H £	Y 6	P 0	Q 1
18	19	1A	1B	1C	1D	1E	1F
O 9	B ?	G &	FIGS	M .	X /	V ;	LTRS

Letters	Figures	Control Chars.

Lab : Code Baudot

```
$ python encode_hello.py
***.**
* *.
    . *
*  .*
*  .*
** .
  *.
*  .**
[...]
```

EBCDIC



Extended Binary Coded Decimal Interchange Code (8 bits)

Code EBCDIC

```
$ python fortran.py
I -> 0xc9
F -> 0xc6
  -> 0x40
C -> 0xc3
A -> 0xc1
R -> 0xd9
T -> 0xe3
A -> 0xc1
[...]
```

US-ASCII (IANA)

USASCII code chart

b_7 b_6 b_5 → / Bits b_4 b_3 b_2 b_1 ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Column → / Row ↓	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

American Standard Code for Information Interchange (7 bits)

Code ASCII

```
#include <stdio.h>
int main(int ac, char *av[])
{
        printf("hell\x08o\n");
}
```

Caractères étendus : ISO 8859

- Ajouts aux 96 caractères affichables de l'US-ASCII
- Encodage sur 8 bits
- 128 nouveaux caractères (utilisation du bit restant d'un octet)
- 16 parties différentes (de 8859-1 à 8859-16)
- Pas utilisable pour certaines langues d'Asie (CJCV)

Code ISO-8859-1

```
(nlp) cervinia:latin1 gasilber$ cat latin1.py
s="Hé ôh"
with open("doc.txt", "w", encoding="iso-8859-1") as f:
    f.write(s)
(nlp) cervinia:latin1 gasilber$ python latin1.py
(nlp) cervinia:latin1 gasilber$ hexdump -C doc.txt
00000000  48 e9 20 f4 68                                    |H. .
00000005
(nlp) cervinia:latin1 gasilber$ cat doc.txt
H? ?h
```

Caractères pour le monde entier : Unicode

- Développé depuis 1991 (première version).
- Standard Unicode 13 (mars 2020) : 144 697 caractères.
- Relié à la norme ISO/CEI 10646 (un sous-ensemble du standard Unicode).
- Un caractère ISO/CEI 10646 : couple nom unique / numéro unique (point de code).
- 245 000 points de codes assignés dans un espace pouvant contenir 1 114 112 codes différents (21 bits).
- 17 zones de 65 536 points de code : plans de code.
- Chaque plan de code est divisé en 4096 colonnes de code de 16 points de code.
- Codes : de `0` à `0x10FFFF` (1 114 112 - 1).
- De `0x0` à `0xFF` : ISO/CEI 8859-1.

Unicode

```
(nlp) cervinia:unicode gasilber$ python wat.py
ê == ê -> False
(nlp) cervinia:unicode gasilber$ python unicode.py
--> Aéêê 5
0 0041 Lu LATIN CAPITAL LETTER A
1 00e9 Ll LATIN SMALL LETTER E WITH ACUTE
2 00ea Ll LATIN SMALL LETTER E WITH CIRCUMFLEX
3 0065 Ll LATIN SMALL LETTER E
4 0302 Mn COMBINING CIRCUMFLEX ACCENT
(nlp) cervinia:unicode gasilber$ python spaces.py
0x2000 '\u2000' EN QUAD Zs ' '
0x2001 '\u2001' EM QUAD Zs ' '
0x2002 '\u2002' EN SPACE Zs ' '
0x2003 '\u2003' EM SPACE Zs ' '
```

UTF-8

- Universal Character Set Transformation Format sur 8 bits.
- Encodage de ISO/CEI 10646 sur 8 bits, compatible avec l'US-ASCII (`0x0` à `0x7F`).
- Stockage d'un point de code sur 1 à 4 octets consécutifs.
- Le décodage des *strings* devient *stateful*.
- Souvent compatible avec le code existant, insensible à l'*endianness*.

Définition du nombre d'octets utilisés dans le codage (attention ce tableau de principe contient des séquences invalides)

Caractères codés	Représentation binaire UTF-8	Premier octet valide (hexadécimal)	Signification
U+0000 à U+007F	0bbb·bbbb	00 à 7F	1 octet, codant jusqu'à 7 bits
U+0080 à U+07FF	110b·bbbb 10bb·bbbb	C2 à DF	2 octets, codant jusqu'à 11 bits
U+0800 à U+FFFF	1110·bbbb 10bb·bbbb 10bb·bbbb	E0 à EF	3 octets, codant jusqu'à 16 bits
U+10000 à U+10FFFF	1111·0bbb 10bb·bbbb 10bb·bbbb 10bb·bbbb	F0 à F3	4 octets, codant jusqu'à 21 bits
	1111·0100 1000·bbbb 10bb·bbbb 10bb·bbbb	F4	

UTF-8

```
(nlp) cervinia:utf8 gasilber$ python count.py
...
0 feff Cf ZERO WIDTH NO-BREAK SPACE
1 0048 Lu LATIN CAPITAL LETTER H
2 00e9 Ll LATIN SMALL LETTER E WITH ACUTE
(nlp) cervinia:utf8 gasilber$ wc -c Classeur1.csv
      XX Classeur1.csv
```

UTF-16 et UTF-32

- UTF-16 : codage sur 2 ou 4 octets (16 ou 32 bits).
- UTF-32 : codage fixe sur 4 octets (32 bits).
- Codages sensibles à l'*endianness*.
- Utilisation d'un BOM, `0xFFFE` (*little endian*) ou 0xFEFF (*big endian*).
- Dans certains fichiers UTF-8 (par ex. CSV généré par Excel) utilisation d'un BOM pour indiquer que le fichier est UTF-8.
- Il existe également un format UTF-5 (unicode sur téléscripteur).
- Bush hid the facts

Formats de fichiers

- Format texte
- Comma-Separated Values (CSV)
- JavaScript Object Notation (JSON)
- YAML Ain't Markup Language
- TOML Tom Obvious, Minimal Language
- eXtensible Markup Language (XML)
- Portable Document Format (PDF)
  - Langage de description de page (Turing complet)
  - $b^2 - 4ac \rightarrow$ `b b mul 4 a mul c mul sub`