Principes élémentaires de recherche et d'analyse de données textuelles en investigation numérique

Emmanuel Giguet

Principes élémentaires de recherche et d’analyse de données textuelles en investigation numérique

Emmanuel Giguet
Chargé de recherche CNRS
Laboratoire GREYC

emmanuel.giguet@cnrs.fr

Recherche et analyse de contenus textuels

- Recherche de noms d'individu, de pseudos, de mots de passe, de lieux, d'adresses, d'adresses mail, de dates, d'organisations, de téléphones, de no de carte bleue, …
- Recherche thématique : harcèlement, prédation, prostitution, trafic en tout genre (faux documents, stups, armes…)

- Dans les documents stockés (fichiers bureautiques)
- Dans les sites web consultés
- Dans les échanges : mail, forum, conversation en ligne
- Dans tout fragment

Recherche et analyse de contenus textuels

- Mais de quel texte parle-t-on ?
  - Fichier de config (.ini)
  - Fichier de données (.json .xml)
  - Fichier de log (.log)
  - Fichier de langage de programmation (.c .java .py)
  - Fichier text/plain (.txt notes.txt todo.txt)
  - Metadonnées de photos (exif en .jpeg)
- Distinction langue/langage

Recherche et analyse de contenus textuels

- Principale difficulté reconnue :
  - quantité de données et temps d’analyse
- Les ordres de grandeur
  - 1 image JPEG de 3 Mo (3000Ko) interprétée en quelques ms.
  - 1 document Libreoffice 18 Ko interprétées en plusieurs secondes (+ temps de chargement de l'appli et d'ouverture du doc)
  - pour info : 4,5 pages = 11000 chars = environ 2500 caractères par page A4 taille 12
- La question centrale de l’interprétation
  - Il faut lire pour comprendre/interpréter : ça prend du temps
  - Photos => possibilité de faire des galeries de miniatures
  - Miniature de 250px interprétable pour une photo pas pour le texte
  - Type de document interprétable par la forme : CV, facture, article

Recherche et analyse de contenu textuel

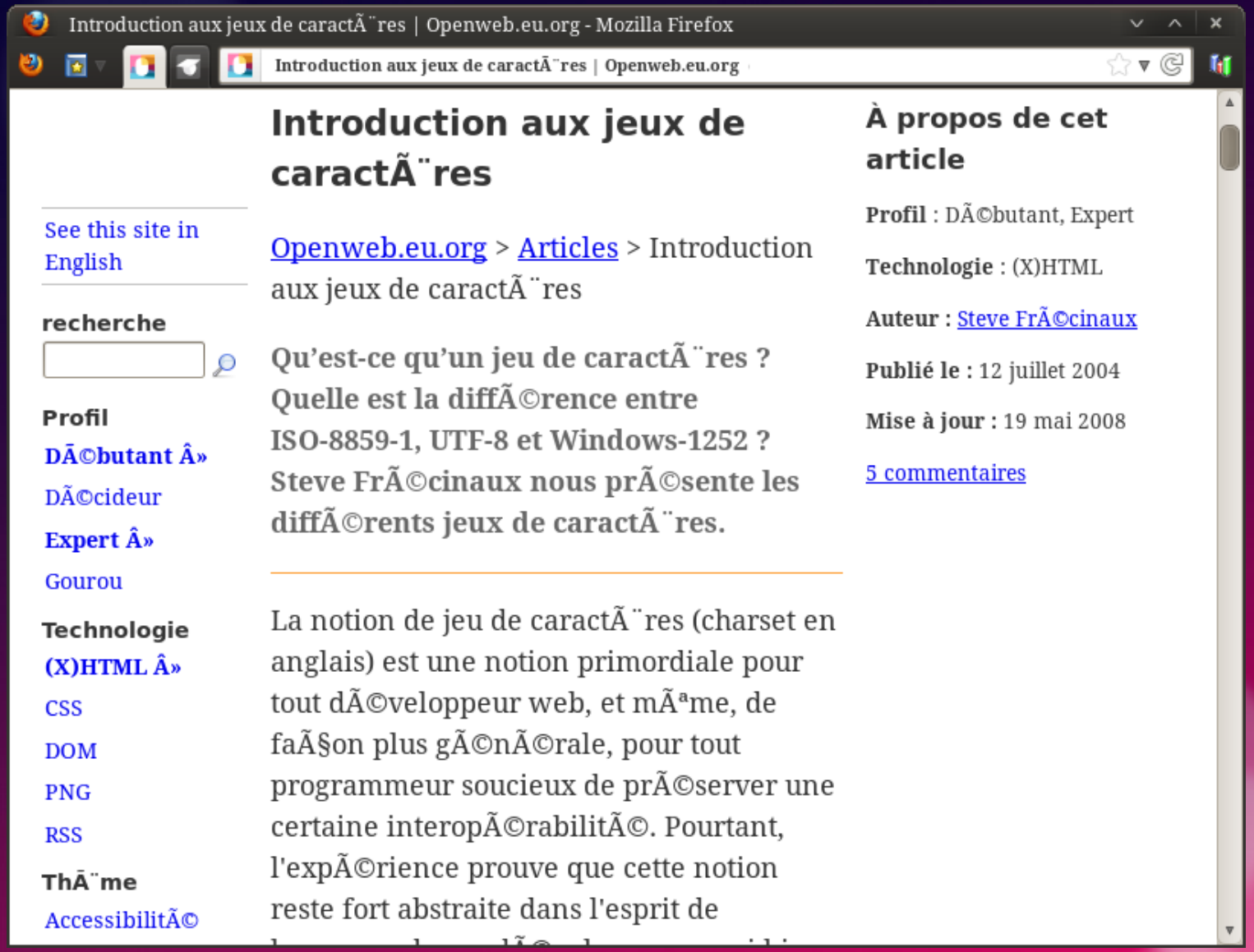
- Focus sur 2 problèmes fondamentaux
  - Le codage des caractères
  - Le codage des données

Recherche et analyse de contenu textuel

- Pourquoi 3 correspondances seulement ?!

```
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ ls -l ex-amelie.*
-rw-r--r-- 1 giguete utilisateurs_du_domaine  203181 déc.  21 11:10 ex-amelie.jpg
-rw-r--r-- 1 giguete utilisateurs_du_domaine 2569100 déc.  21 11:16 ex-amelie.odt
-rw-r--r-- 1 giguete utilisateurs_du_domaine  288671 déc.  21 11:14 ex-amelie.odt.pdf
-rw-r--r-- 1 giguete utilisateurs_du_domaine  288941 déc.  21 11:16 ex-amelie.pdf
-rw-r--r-- 1 giguete utilisateurs_du_domaine     520 déc.  21 11:44 ex-amelie.tex
-rw-r--r-- 1 giguete utilisateurs_du_domaine     574 déc.  21 11:43 ex-amelie.tex~
-rw-r--r-- 1 giguete utilisateurs_du_domaine       6 déc.  20 22:15 ex-amelie.utf8.cp1252.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine       6 déc.  20 22:15 ex-amelie.utf8.latin1.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine       7 déc.  20 21:52 ex-amelie.utf8-nfc.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      14 déc.  20 22:08 ex-amelie.utf8-nfc.utf16be.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      12 déc.  20 22:01 ex-amelie.utf8-nfc.utf16be.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      14 déc.  20 22:08 ex-amelie.utf8-nfc.utf16le.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      12 déc.  20 22:00 ex-amelie.utf8-nfc.utf16le.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine       8 déc.  20 21:52 ex-amelie.utf8-nfd.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      16 déc.  20 22:09 ex-amelie.utf8-nfd.utf16be.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      14 déc.  20 22:00 ex-amelie.utf8-nfd.utf16be.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      16 déc.  20 22:09 ex-amelie.utf8-nfd.utf16le.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine      14 déc.  20 22:00 ex-amelie.utf8-nfd.utf16le.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine       7 déc.  20 21:51 ex-amelie.utf8.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ grep Amélie ex-amelie*
grep Amélie ex-amelie*
Fichier binaire ex-amelie.jpg correspondant
ex-amelie.utf8-nfc.txt:Amélie
ex-amelie.utf8.txt:Amélie
```

Introduction au codage des caractères



Des influences lointaines

- Au tout début, des transmissions en 7 bits (non-8bit-clean)
  - Objectif : détecter l'intégrité des données transmises
  - Pallier les problèmes d'interférence sur le canal
  - Atténuation du signal, bruit (électromagnétisme), dégradation (diaphonie)
- Envoi de messages textuels
  - Transmission sur 8 bits mais 7 bits de données
  - ASCII (7bits = 127 caractères) : début 1960
- Détection des problèmes de transmission
  - Signalisation intrabande : In-Band Signaling
  - La parité / Le checksum / le CRC
  - Des meta-données
- Idem pour le stockage : cartes perforées/EBCDIC

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	00000000	00000001
1010001	3	10100011	10100010
1101001	4	11010010	11010011
1111111	7	11111111	11111110

Table ASCII

Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value	Hex	Value
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Table ASCII

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Le code ASCII

- 32 caractères de contrôle, non imprimables :
  - Les 32 premiers caractères (base 16) : (C0 control codes)
  - Pour les transmissions (SOH STX EOT EOT ACK NAK SYN...)
  - Pour la mise en forme (HT CR LF VT ...)
  - Pour marquer la structure = des séparateurs (FS GS RS US)
- 96 caractères imprimables :
  - Chiffres / Majuscules / Minuscules / Caractères spéciaux
  - Référence
    - au clavier américain
    - à l'écriture latine : Basic Latin
  - Des caractères non accentués
- Puis le bloc C1 : contrôles additionnels

En route vers l'internationalisation

- ASCII très vite insuffisant
  - Pas de caractères accentués
- Apparition du Latin 1 : iso-8859-1 (1986)
  - Europe occidentale : allemand, anglais, basque, catalan, danois, espagnol, italien, néerlandais, norvégien, portugais et suédois
- Apparition du Latin 2 : iso-8859-2
  - Europe centrale : bosnien, croate, polonais, tchèque, slovaque, slovène et hongrois
- Support partiel du français, passage à l'euro
  - Apparition du Latin-15 (1998)
  - Apparition du Latin-16
- Codage sur 8 bits compatible ASCII

	A4	A6	A8	B4	B8	BC	BD	BE
8859-1	¤	¦	¨	´	¸	¼	½	¾
8859-15	€	Š	š	Ž	ž	Œ	œ	Ÿ

En route vers l’internationalisation

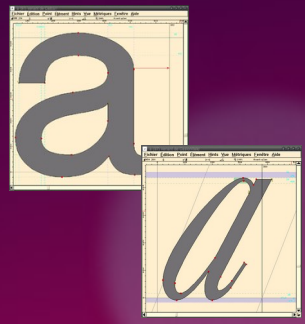
- Internationalisation par région du monde
  - Par système d’écriture / alphabet
- Une table par région : des tentatives normalisation plus ou moins fructueuses
  - iso-Latin-3 : Europe du Sud – turc, maltais
  - iso-Latin-4 : Europe du Nord – estonien, letton, lituanien, groenlandais, sami
  - iso-Latin-5 : Slave/Cyrillique – russe, serbe, ukrainien
  - iso-Latin-??
- Codage sur 8 bits compatible ASCII

En route vers l’internationalisation

- Standardisation : l’ISO à la barre !
  - Toujours des intérêts privés à contre-courant/à l’influence
  - Existence de jeux de caractères propriétaires concurrents
  - Contraintes techniques/matérielles, en phase avec les besoins du marché, utilisateurs captifs
- L’ISO propose ISO/CEI 646 (ASCII)
  - IBM propose EBCDIC
- L’ISO propose le ISO-8859-1
  - Microsoft propose Windows-1252 (CP1252)
  - Apple propose MacRoman
- L’ISO propose ISO/CEI 10646
  - Le consortium privé Unicode influence et participe

Terminologie relative au codage

- Jeu de caractères = charset
  - Table de correspondances entre des codepoints et des caractères abstraits
- Codepoint : une valeur numérique, le numéro d'ordre
  - En ASCII : 0x61 => a 0x62 => b
- Caractère abstrait = unité minimale de sens
  - Ex : la première lettre minuscule de l'alphabet latin
  - Indépendant de la police utilisée,
  - Indépendant des effets de style pour le rendu,
  - Dépendant de la casse : 2 codepoints



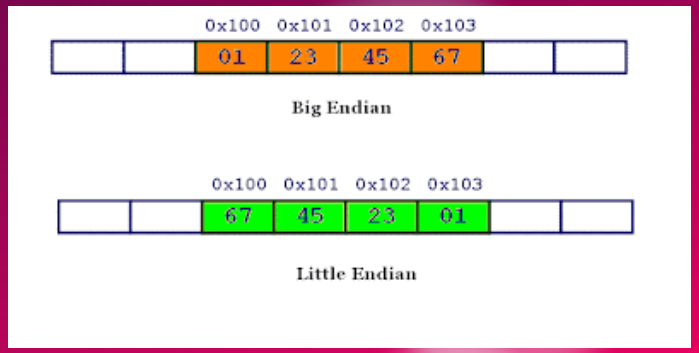
- Glyphe : ce que le lecteur perçoit, la représentation graphique du caractère
- Byte-sequence : la représentation binaire du codepoint, comment le codepoint est représenté par 1 ou plusieurs octets
- Le codepoint et la byte-sequence : Ce que l'ordinateur manipule, un entier, stocké sous la forme d'une séquence d'octets

Vers un jeu de caractère universel : UCS

- Limite de l'approche « régionale »
  - Cohabitation de plusieurs alphabets dans un même document ou message
  - Éviter le développement de nombreux jeux locaux en Asie
- Limite de l'approche 8bits
  - Trop de caractères à coder pour un codage sur 8bits
  - Passage du SBCS (Single-Byte Character Set) à MBCS (Multi-Byte Character Set)
- ISO-2022, une tentative de normalisation :
  - ISO-2022 et ses variantes : le chinois (ISO 2022-CN), le coréen (ISO 2022-KR) et le japonais (ISO 2022-JP)
  - ShiftJIS (japon) Big5 (taïwan) GB2312 (chine) EucKR (corée)
  - Possibilité de mixer les codes à l'aide de séquences d'échappement
  - Codage de longueur variable => compatibilité ASCII
- ISO/CEI 10646 et UCS :
  - Des codages informatiques aujourd'hui très répandus : UTF-8 UTF-16 UTF-32

ISO/CEI 10646 et ses transformations

- Transformation
  - sérialisation de la byte-sequence
- UTF-8 : Longueur variable (1 à 4 octets)
  - Compatible avec ASCII
  - Compact pour les langues occidentales
  - Pas économe pour les langues asiatiques
- UTF-16 et 32 : Longueur fixe (2 et 4 octets)
  - Plus économe pour les langues asiatiques
  - Non compatible avec ASCII
- Et l'endianness (= le boutisme) dans tout ça ??
  - Little endian UTF-16LE UTF-32LE
  - Big endian UTF-16BE UTF-32BE



UTF-8 : codage de longueur variable

Point de code	Codage UTF-8 en binaire	1er octet : valeurs possibles	Nb. de bits à coder
U+0000 à U+007F	0xxxxxxx (table ASCII)	00 à 7F	7
U+0080 à U+07FF	110xxxxx 10xxxxxx	C2 à DF	5+6=11
U+0D00 à U+FFFF	1110xxxx 10xxxxxx 10xxxxxx	E0 à EF	4+6+6=16
U+10000 à U+10FFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx	F0 à F4	2+6+6+6=20

Équivalence Unicode : la composition

***Amélie* en Unicode UTF16BE formes NFC et NFD**

Caractère représenté	A	m	é		l	i	e
UTF16-BE NFC	0041	006d	00e9		006c	0069	0065
UTF16-BE NFD	0041	006d	0065	0301	006c	0069	0065
Caractère représenté	A	m	e	◌́	l	i	e

Unicode BOM : Indicateur d’ordre d’octets

- Préfixe BOM = Byte Order Mark

Codage	Séquence d'octets (Représentation)
UTF-8	EF BB BF
UTF-16 Big Endian	FE FF
UTF-16 Little Endian	FF FE
UTF-32 Big Endian	00 00 FE FF
UTF-32 Little Endian	FF FE 00 00
SCSU	0E FE FF
UTF-7	2B 2F 76 et l'*un* des octets suivants : [38 | 39 | 2B | 2F]
UTF-EBCDIC	DD 73 66 73
BOCU-1	FB EE 28
UTF-1	F7 64 4C

- BOM vs Type mime : 2 types de signature
  - Codage du charset vs Codage des données

```
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.txt
00000000  41 6d c3 a9 6c 69 65                              |Am..lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ php to-nfd.php Amélie > ex-amelie.utf8-nfd.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ php to-nfc.php Amélie > ex-amelie.utf8-nfc.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfc.txt
00000000  41 6d c3 a9 6c 69 65                              |Am..lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfd.txt
00000000  41 6d 65 cc 81 6c 69 65                           |Ame..lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8-nfc.txt | iconv -f utf-8 -t utf-16le > ex-amelie.utf8-nfc.utf16le.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8-nfd.txt | iconv -f utf-8 -t utf-16le > ex-amelie.utf8-nfd.utf16le.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8-nfd.txt | iconv -f utf-8 -t utf-16be > ex-amelie.utf8-nfd.utf16be.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8-nfc.txt | iconv -f utf-8 -t utf-16be > ex-amelie.utf8-nfc.utf16be.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfc.utf16be.txt
00000000  00 41 00 6d 00 e9 00 6c  00 69 00 65              |.A.m...l.i.e|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfc.utf16le.txt
00000000  41 00 6d 00 e9 00 6c 00  69 00 65 00              |A.m...l.i.e.|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfd.utf16le.txt
00000000  41 00 6d 00 65 00 01 03  6c 00 69 00 65 00        |A.m.e...l.i.e.|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8-nfd.utf16be.txt
00000000  00 41 00 6d 00 65 03 01  00 6c 00 69 00 65        |.A.m.e...l.i.e|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ printf '\xEF\xBB\xBF'
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ printf '\xFE\xFF' > ex-utf16bebom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ printf '\xFF\xFE' > ex-utf16lebom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ ls -l ex-utf16*
-rw-r--r-- 1 giguete utilisateurs_du_domaine 2 déc.  20 22:07 ex-utf16bebom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 2 déc.  20 22:07 ex-utf16lebom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-utf16bebom.txt ex-amelie.utf8-nfc.utf16be.txt > ex-amelie.utf8-nfc.utf16be.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-utf16lebom.txt ex-amelie.utf8-nfc.utf16le.txt > ex-amelie.utf8-nfc.utf16le.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-utf16lebom.txt ex-amelie.utf8-nfd.utf16le.txt > ex-amelie.utf8-nfd.utf16le.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-utf16bebom.txt ex-amelie.utf8-nfd.utf16be.txt > ex-amelie.utf8-nfd.utf16be.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8.txt
Amélie(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8-nfd.txt
Ame´lie(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ iconv -f utf8 -t iso-8859-1 < ex-amelie.utf8.txt > ex-amelie.utf8.latin1.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ iconv -f utf8 -t windows-1252 < ex-amelie.utf8.txt > ex-amelie.utf8.cp1252.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.txt
00000000  41 6d c3 a9 6c 69 65                              |Am..lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.latin1.txt
00000000  41 6d e9 6c 69 65                                 |Am.lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.cp1252.txt
00000000  41 6d e9 6c 69 65                                 |Am.lie|
```

```
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.txt
00000000  41 6d c3 a9 6c 69 65                              |Am..lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.latin1.txt
00000000  41 6d e9 6c 69 65                                 |Am.lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ hexdump -C ex-amelie.utf8.cp1252.txt
00000000  41 6d e9 6c 69 65                                 |Am.lie|
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ cat ex-amelie.utf8.txt
Amélie(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ grep Amélie ex-amelie.utf8*
ex-amelie.utf8-nfc.txt:Amélie
ex-amelie.utf8.txt:Amélie
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ ls -l ex-amelie.utf8*
-rw-r--r-- 1 giguete utilisateurs_du_domaine  6 déc.  20 22:15 ex-amelie.utf8.cp1252.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine  6 déc.  20 22:15 ex-amelie.utf8.latin1.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine  7 déc.  20 21:52 ex-amelie.utf8-nfc.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 14 déc.  20 22:08 ex-amelie.utf8-nfc.utf16be.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 12 déc.  20 22:01 ex-amelie.utf8-nfc.utf16be.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 14 déc.  20 22:08 ex-amelie.utf8-nfc.utf16le.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 12 déc.  20 22:00 ex-amelie.utf8-nfc.utf16le.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine  8 déc.  20 21:52 ex-amelie.utf8-nfd.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 16 déc.  20 22:09 ex-amelie.utf8-nfd.utf16be.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 14 déc.  20 22:00 ex-amelie.utf8-nfd.utf16be.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 16 déc.  20 22:09 ex-amelie.utf8-nfd.utf16le.bom.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine 14 déc.  20 22:00 ex-amelie.utf8-nfd.utf16le.txt
-rw-r--r-- 1 giguete utilisateurs_du_domaine  7 déc.  20 21:51 ex-amelie.utf8.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings ex-amelie.utf8.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings ex-amelie.utf8.latin1.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e S ex-amelie.utf8.latin1.txt
strings -e S ex-amelie.utf8.latin1.txt
Am\351lie
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e S ex-amelie.utf8.txt
Amélie
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings ex-amelie.utf8-nfc.utf16be.txt
strings ex-amelie.utf8-nfc.utf16be.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e b ex-amelie.utf8-nfc.utf16be.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e b ex-amelie.utf8-nfc.utf16be.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e b ex-amelie.utf8-nfd.utf16be.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e b ex-amelie.utf8-nfd.utf16be.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e l ex-amelie.utf8-nfc.utf16le.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e l ex-amelie.utf8-nfd.utf16le.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e l ex-amelie.utf8-nfd.utf16le.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$ strings -e l ex-amelie.utf8-nfc.utf16le.bom.txt
(base) (focal-base r4703)giguete@C302L-G19P17:~/Documents$
```

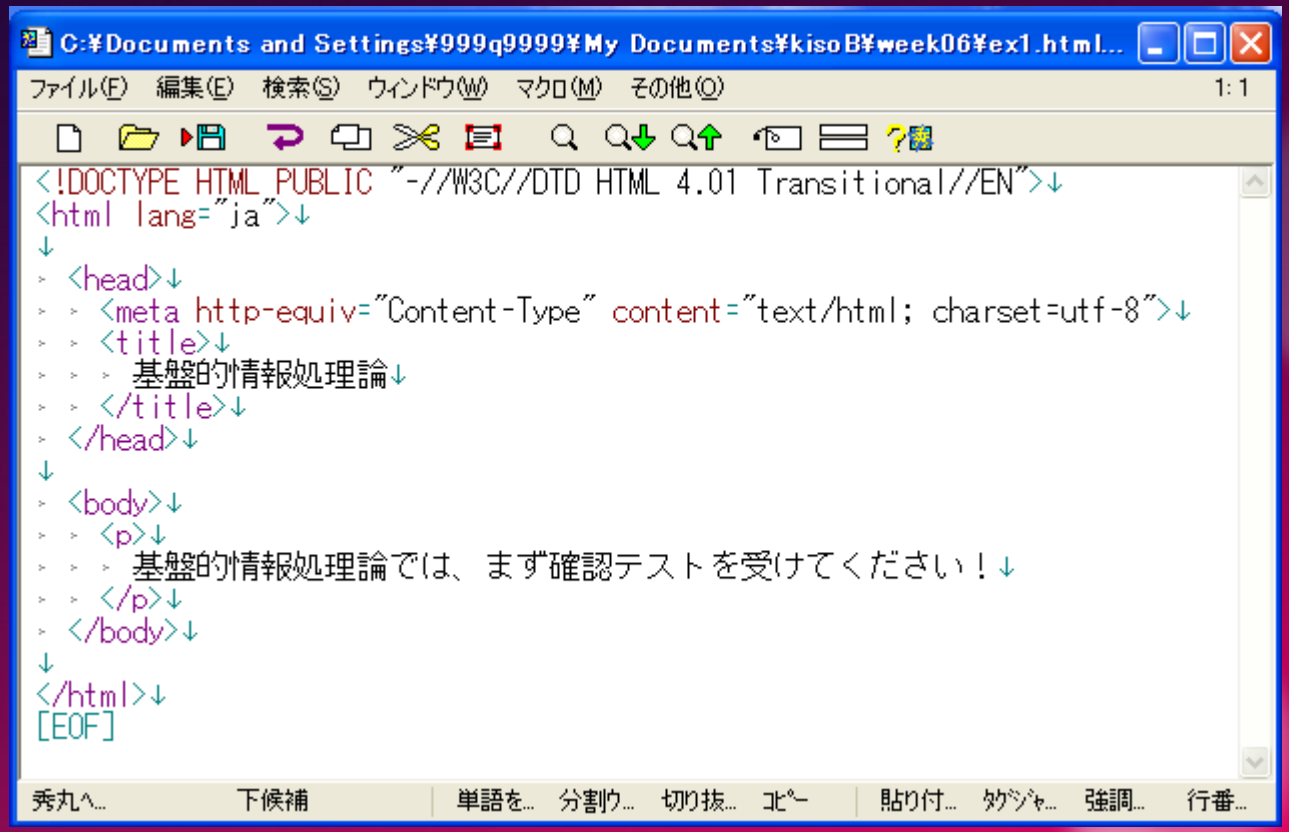
Metadonnées : indicateur de charset

- Déclaratif dans l’entête du fichier
- Déclaratif dans l’entête du protocole de transfert

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>UTF-8 Example</title>
  </head>
  <body>
    <p>Hablas español?</p>
  </body>
</html>
```

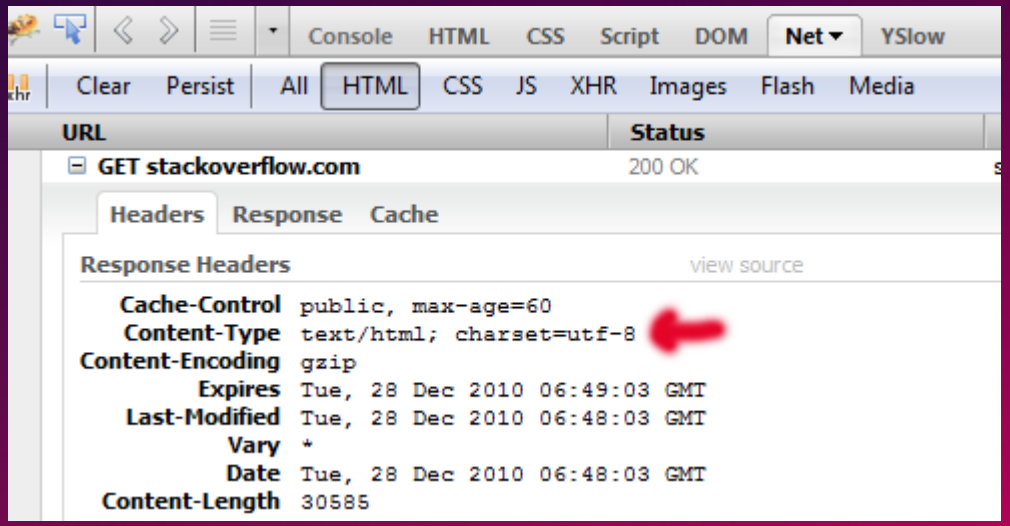
Metadonnées : indicateur de charset

- Déclaratif dans l’entête du fichier
- Déclaratif dans l’entête du protocole de transfert



Metadonnées : indicateur de charset

- Déclaratif dans l'entête du fichier
- Déclaratif dans l'entête du protocole de transfert



Quelques techniques incontournables

- Utiliser un détecteur de jeu de caractères
  - Charset detector : attention à la fiabilité !
  - Ex : uchardet
  - Heuristique / statistique / lié à l'identification de la langue
- Utiliser un convertisseur de jeu de caractères
  - Charset converter : convertir vers un jeu englobant !
  - Ex : iconv
- Utiliser un extracteur de chaînes avec gestion des codages (itération sur les charsets si nécessaire)
  - Ex : strings –encoding=<b|e|S>
  - Pour l'analyse de fragments : coupler avec dd hexdump xxd

Quelques erreurs classiques

- Matching apostrophe :
  - Inter charset : cp1252 / latin1
  - Intra charset : Quotation marks en UCS
- Matching insensible à la casse (-i)
- Matching insensible aux accents
- Matching des ligatures (lettres entrelacées)

$AE \rightarrow Æ$	$ij \rightarrow ij$
$ae \rightarrow æ$	$st \rightarrow st$
$OE \rightarrow Œ$	$ft \rightarrow ft$
$oe \rightarrow œ$	$et \rightarrow \&$
$ff \rightarrow ff$	$fs \rightarrow ß$
$fi \rightarrow fi$	$ffi \rightarrow ffi$

Quelques erreurs classiques

- Une chaîne de traitement non UTF-8
- Vérifier que la chaîne de traitement est UTF-8
  - Fichier à analyser à convertir vers UTF-8
  - Fichier de script (python, php, …) codé en UTF-8
  - Le terminal de commandes configuré en UTF-8

Les codage des données

- Dans un environnement 8bits non fiable, il convient de garantir l'intégrité des données
  - Ex : l'envoi et la réception de mails et MIME
  - Transmettre des données 8 bits sur 7 bits
  - Garantir l'intégrité / la réversibilité
  - 3 principaux systèmes de codage de données
    - Quoted-printable
    - Base64
    - Uuencode
  - Application sur la séquence d'octets

Les codage des données

- Contexte : environnement 8bits non fiable
- Objectif : garantir l'intégrité des données
- Application : l'envoi et la réception de mails avec MIME
  - Transmettre des données 8 bits sur 7 bits
  - Garantir l'intégrité / la réversibilité
  - 3 principaux systèmes de codage de données
    - Quoted-printable : pour les données textuelles
    - Base64 : plutôt pour les données binaires
    - Uuencode (avant MIME)
  - Couche de codage au dessus du codage des caractères

Les codage des données et SMTP/MIME

- Quoted-Printable (ou QP) :
  - Lisible : Printable Char ASCII préservé sauf = (caractère d'échappement)
  - Codage du bloc C0, du =, et au dessus de 126 : =<Hex>
  - 76 caractères max par ligne
- Base64 (ou B64) :
  - Texte « occidental » pas lisible mais pas fait pour
  - Codage sur 6 bits : accroissement de 1/3 de la taille
  - 76 caractères max par ligne
  - Padding avec des =
- Codage des caractères à expliciter (ou deviner)

```
----------------------------------------------------
From    : Bostjan Laba via ClarionHub <notifications@clarionhub.com>
Subject :  [ClarionHub] Activating image control as ActiveImage at runtime
----------------------------------------------------
From    : Quora Digest <english-personalized-digest@quora.com>
Subject :  =?utf-8?q?Why_doesn=27t_Microsoft_take_action_on_the_millions_of_pirated_c
op?=
 =?utf-8?q?ies_of_Windows=3F?=
----------------------------------------------------
From    : "Redbubble" <heythere@m.redbubble.com>
Subject :  =?utf-8?B?MjUlIG9mZiBhbGwgY2xvdGhlcyDwn5GVIEdldCBpbnNwaXJlZCBh?=
           =?utf-8?B?bmQgZ2V0IGRyZXNzZWQu?=
----------------------------------------------------
From    : "Codeur.com" <site@digest.codeur.com>
Subject :  =?UTF-8?Q?17_nouveaux_projets_publi=C3=A9s_sur_Codeur.com?=
----------------------------------------------------
```

Les codage des données

- uudecode :
  - Moins utilisé de nos jours mais à connaître
  - Même principe que B64
  - Jeu de caractère pour coder moins simple que B64
- Urlencode :
  - Codage de données textuelles dans les URLS
  - Même principe que QP : % est le caractère d'échappement
  - https://fr.wikipedia.org/wiki/%C3%89conomie
  - A connaître pour traiter les recherches effectuées et les urls consultées
  - Appliqué sur une séquence d'octets en UTF-8

Les codage des données et HTML

- HTML et le codage des entités
  - Possibilité de choisir un codage sur 8 bits,
    UTF-8 -16, ou iso-latin-1 ou window-1252, ...
- Possibilité de sauvegarder la séquence d’octets
- Possibilité d’utiliser des entités
  - 3 formats : &#x<hex>; &#<dec>; &<name>;
  - Hex et dec représente le codepoint :
    pas la séquence d’octets !
- Impératif de décoder (vers UTF-8) pour analyser
  - Difficulté : charset déclaré erroné, mix de charsets

Les codage des données et TeX/LaTeX

- Nouvelle illustration avec TeX/LaTeX:
  - le codage des données sur 8 bits est omniprésent
  - Nécessité de décoder (vers UTF-8) pour analyser !
- Codage des accents : é = \'e ou \'{e}
- Indication des césures : bou\-gie

- \'e ou \'{e} pour é ;
- \`e ou \`{e} pour è ;
- \^\i ou \^{\i} pour î ;
- \`A ou \`{A} pour À ;
- \"e ou \"{e} pour ë ;

Longtemps, je me suis couché de bonne heure. Parfois, à peine ma bougie éteinte, mes yeux se fermaient si vite que je n'avais pas le temps de me dire : « Je m'endors. »

Longtemps, je me suis couché de bonne heure. Parfois, à peine ma bougie éteinte, mes yeux se fermaient si vite que je n'avais pas le temps de me dire : « Je m'endors. »

Formats ouverts et formats propriétaires

- Pour les fichiers textes :
  - Des formats ouverts
  - Volonté de préserver l'interopérabilité et la lisibilité
  - Volonté d'assurer l'intégrité si 8 bits
  - Nécessite un décodage simple des données et une conversion de jeu de caractères : html latex rtf
- Pour les documents :
  - Des formats propriétaires .doc (Word) avant OpenXML et OpenDocument
  - Pas de standardisation pendant longtemps, possibilité de mise en page complexe
  - Nécessite des extracteurs de textes : strings insuffisant ou inutilisable
  - Extracteurs pas toujours disponibles pour les formats anciens/abandonnés

Formats ouverts : OpenXXXXX

- Pour les documents :
  - 2 grands formats : OpenDocument et OpenXML
  - Répond aux besoins d'interopérabilité et de convergence pour la bureautique
- Un document = 1 container binaire = fichier ZIP
  - Contenu textuel pas accessible sans décompresser
  - Facilement exploitable après décompression
  - Principaux contenus textuels OpenDocument : content.xml et meta.xml

Les documents PDF

- Portable Document Format, par Adobe
- 3 types pour les documents « textuels »
  - PDF Image : contient l'image d'une page, souvent un scan de page
  - PDF Texte : contient du contenu positionné du texte, des formes vectorielles, des images
  - PDF Texte+Image : l'image d'une page et le résultat de l'ocrisation
- Format ouvert :
  - Spécifications accessibles
  - Outils d'extraction disponibles

Les documents PDF

- 1 seul fichier pour un document : pas un container
- L’unité historique : la page
- Pas de représentation de la mise en forme
  - Colonne, entête, pied de page, titraison = résultat d’une activité interprétative du lecteur
- Unité « mot » parfois absente ou à reconstruire :
  - Calcul des mots par proximité des caractères
- Ordre de lecture non représenté
- Logiciels d’extraction pas totalement fiables
  - Texte sérialisé dans le mauvais ordre
  - Caractères deviennent des mots si espacement trop grand
  - Dissimulation de texte possible derrière des images ou formes
  - Problème de fonts corrompues ; codage problématique

Les documents PDF

- Représentation du texte en PDF
  - Dictionnaire d'objets sérialisés, codés, souvent compressés pour économie de place
- Texte pas directement accessible sans décodage
  - Codage par cascade de filtres

ASCIIHexDecode	no	Decodes data encoded in an ASCII hexadecimal representation, reproducing the original binary data.
ASCII85Decode	no	Decodes data encoded in an ASCII base-85 representation, reproducing the original binary data.
LZWDecode	yes	Decompresses data encoded using the LZW (Lempel-Ziv-Welch) adaptive compression method, reproducing the original text or binary data.
FlateDecode	yes	*(PDF 1.2)* Decompresses data encoded using the zlib/deflate compression method, reproducing the original text or binary data.

Les documents PDF

- Représentation du texte en PDF
  - Objets sérialisés et codés :
    Metadata + Content Stream
  - Texte pas directement accessible sans décodage et décompression
  - Codage et compression du flux à l'aide de Filtres
  - Cascade de filtres à appliquer pour décoder

```
1 0 obj
  << /Length 534
     /Filter [/ASCII85Decode /LZWDecode]
  >>
```

Représentation des contenus textuels en PDF

- Contenu textuel codé/compressé
- Cascade de deux filtres à appliquer :
  - LZWDecode puis ASCII85Decode

```
1 0 obj
   << /Length 534
      /Filter [/ASCII85Decode /LZWDecode]
   >>
stream
J..)6T`?p&<!J9%_[umg"B7/Z7KNXbN'S+,*Q/&"OLT'F
LIDK#!n`$"<Atdi`\Vn%b%)&'cA*VnK\CJY(sF>c!Jnl@
RM]WM;jjH6Gnc75idkL5]+cPZKEBPWdR>FF(kj1_R%W_d
&/jS!;iuad7h?[L-F$+]]0A3Ck*$I0KZ?;<)CJtqi65Xb
Vc3\n5ua:Q/=0$W<#N3U;H,MQKqfg1?:lUpR;6oN[C2E4
ZNr8Udn.'p+?#X+1>0Kuk$bCDF/(3fL5]Oq)^kJZ!C2H1
'TO]Rl?Q:&'<5&iP!$Rq;BXRecDN[IJB`,)o8XJOSJ9sD
S]hQ;Rj@!ND)bD_q&C\g:inYC%)&u#:u,M6Bm%IY!Kb1+
":aAa'S`ViJglLb8<W9k6Yl\\0McJQkDeLWdPN?9A'jX*
al>iG1p&i;eVoK&juJHs9%;Xomop"5KatWRT"JQ#qYuL,
JD?M$0QP)lKn06l1apKDC@\qJ4B!!(5m+j.7F790m(Vj8
8l8Q:_CZ(Gm1%X\N1&u!FKHMB~>
endstream
endobj
```

Représentation des contenus textuels en PDF

- Contenu textuel après décodage

```
1 0 obj
   << /Length 568 >>
stream
2 J
BT
/F1 12 Tf
0 Tc
0 Tw
72.5 712 TD
[(Unfiltered streams can be read easily) 65 (, )] TJ
0 -14 TD
[(b) 20 (ut generally tak) 10 (e more space than \311)] TJ
T* (compressed streams.) Tj
0 -28 TD
[(Se) 25 (v) 15 (eral encoding methods are a) 20 (v) 25 (ailable in PDF) 80 (.)] TJ
0 -14 TD
(Some are used for compression and others simply) Tj
T* [(to represent binary data in an ) 55 (ASCII format.)] TJ
T* (Some of the compression filters are \
suitable ) Tj
T* (for both data and images, while others are \
suitable only ) Tj
T* (for continuous-tone images.) Tj
ET
endstream
endobj
```

Les codages et l’investigation numérique

- L’identification du ou des jeux de caractères et l’identification du codage des données ont un impact fondamental dans l’investigation

  => Engendre du silence = des oublis
- La capacité à traiter des données textuelles corrompues ou non parfaites est essentielles

  => codage déclaré invalide, mix de codage
- Interprétabilité du résultat au cœur du process

  => bien décodé si interprétable

Les codages et l’investigation numérique

- 3 types d’outils à disposition
- Les commandes de bas niveaux :
  - Dans un terminal, en ligne de commande :

    dd hexdump xxd uchardet iconv strings grep
- Les extracteurs dédiés :
  - Pdftoxml
- OCRisation : Tesseract / Kraken / FineReader
- Les boîtes à outils :
  - Apache Tika : détection et extraction de métadonnées et du contenu textuel de nombreux formats
  - Attention : pas vraiment un outil forensique à la base

La recherche d’information

- Recherche par mots-clés ou expressions régulières
- Recherche d’entités nommées : Nom d’individu, pseudo, lieu, adresse, adresse mail, date, organisation, téléphone, no de carte bleue, …
- Apprentissage par réseaux de neurones (Bi-LSTM : utilise les contextes gauche et droit) :
  - capture mieux la variabilité / plus coûteux en espace / explicabilité ?

La recherche d’information

- La détection de thématique : harcèlement, menace, prédation, racisme, injurieux…
- La prise en compte du niveau de langue
  - Exemple tiré du Détecteur de harcèlement de B.Maurice (2019)

    « un jour jvai le croiser a chatelet jvai lui niquer sa grand mere surtout pcq un moment il sfoutait dla gueule des marocains »
- La remise en contexte du texte : horodatage, contenus liés, événements liés (caméra, micro, partage, réception…)
- L’analyse par rebond