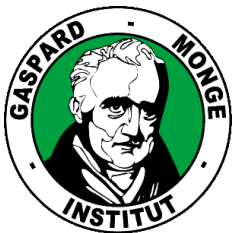


Master 1 Informatique – Université Marne-la-Vallée (IGM)
09/03/2012 – Cours 1
Ingénierie Linguistique

Introduction

Langages rationnels



Philippe Gambette

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Aspects administratifs

Déroulement du cours

- CM (6 x 2h = 12h)

Philippe Gambette (philippe.gambette@univ-mlv.fr)

Cours 1 à 4, cours 6, 1 TP pour un groupe

- TP (6 x 2h = 12h)
2 groupes

Anthony Sigogne

Cours 5 et TP

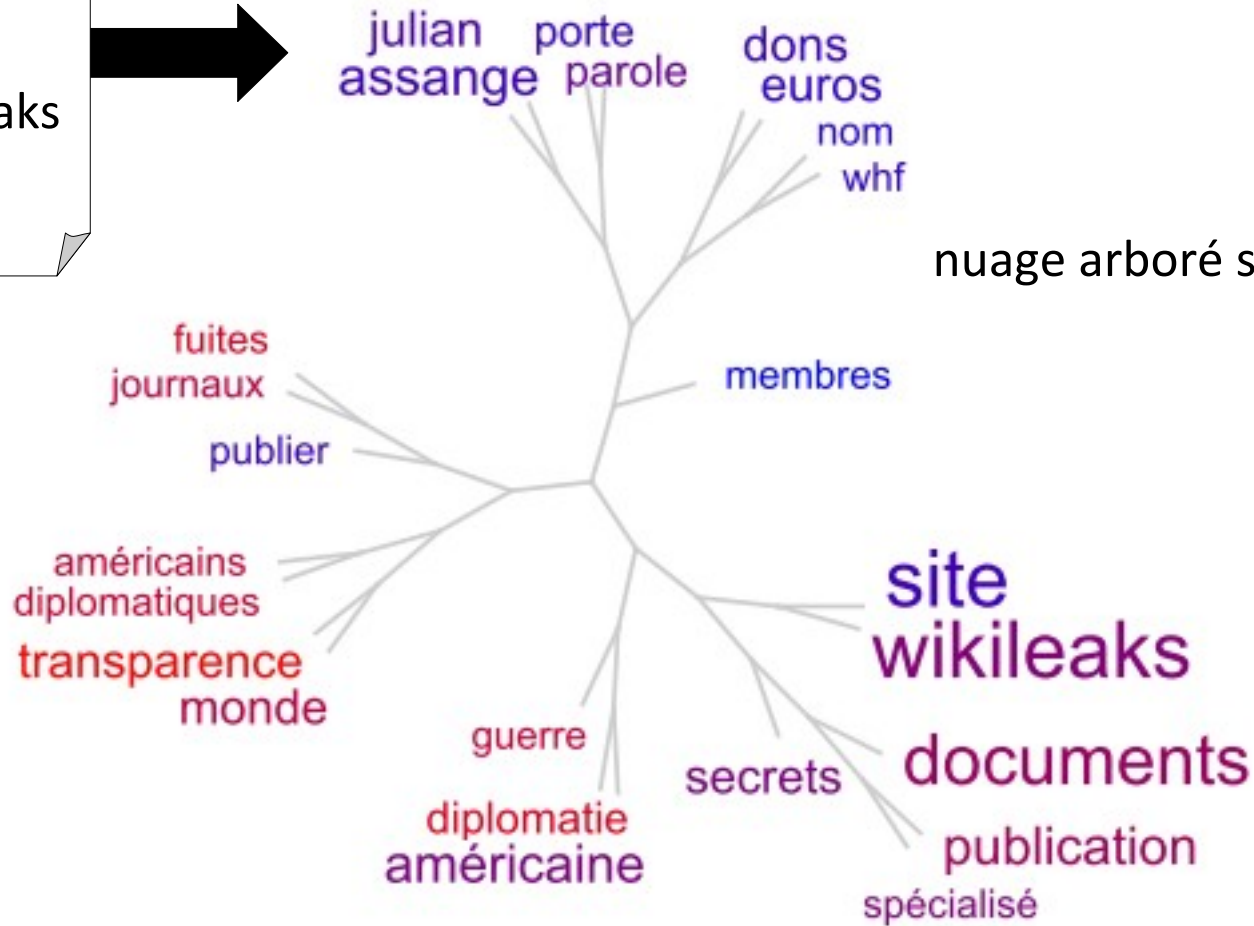
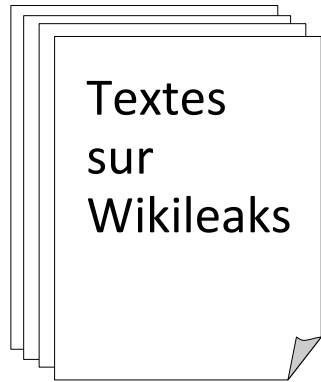
Évaluation

- Examen final
- Projet de programmation
- TP à rendre (éventuellement)

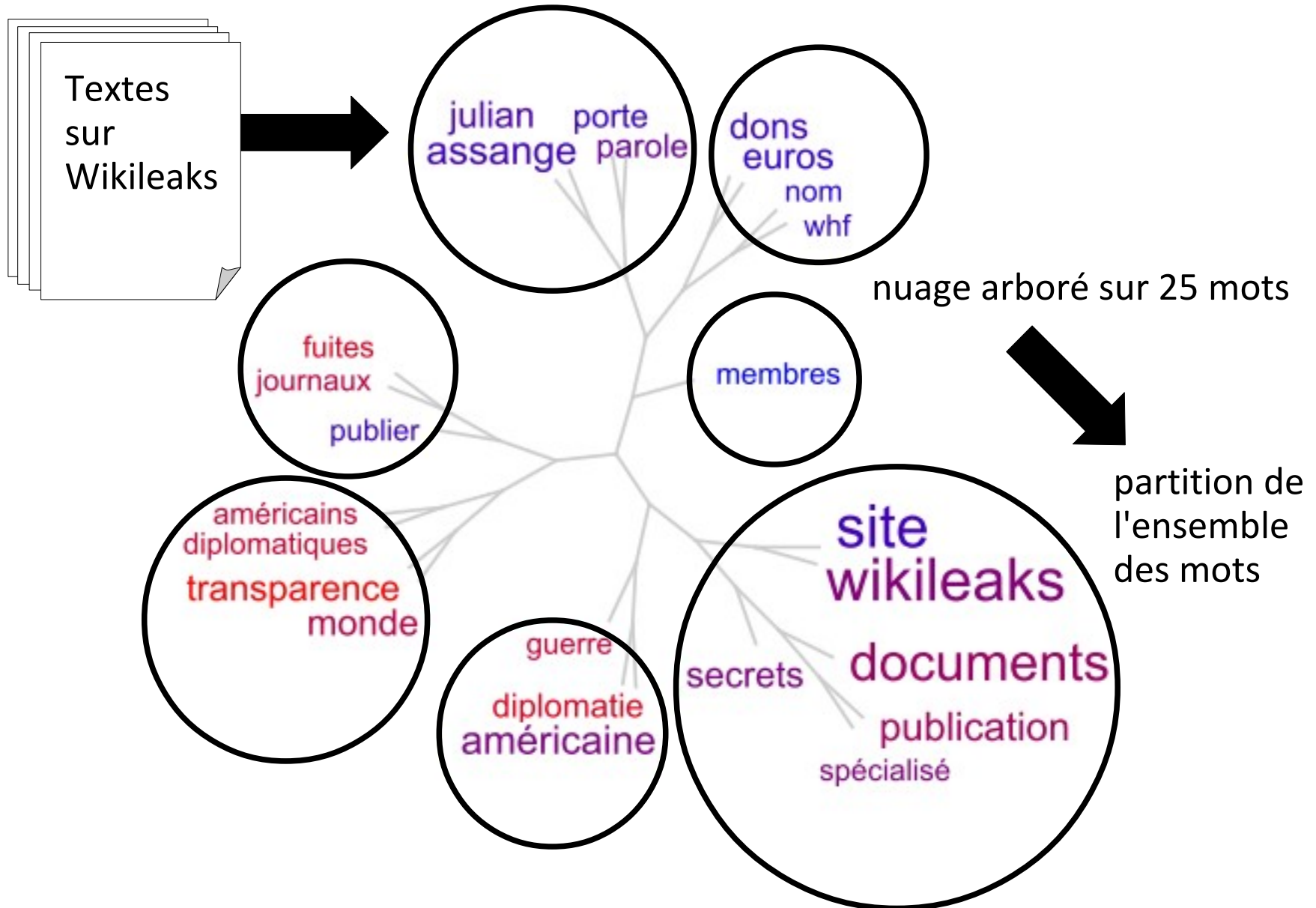
Supports de cours, sujets de TP

- <http://tinyurl.com/Infoling2012>

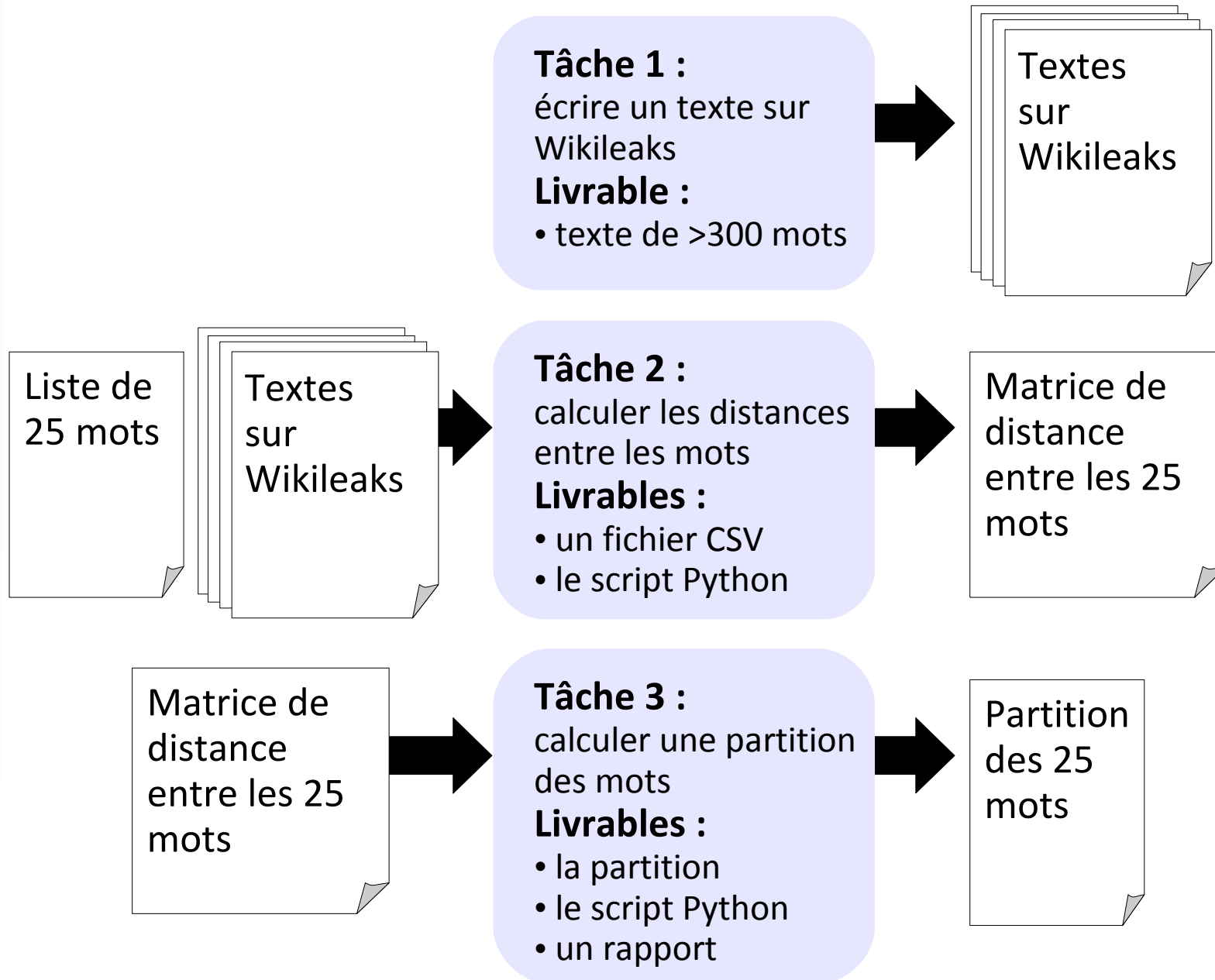
Projet de programmation



Projet de programmation



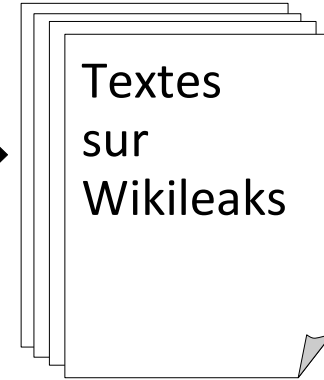
Projet de programmation



Projet de programmation

*Par groupes de 3
(2 groupes de 2)*

Tâche 1 :
écrire un texte sur
Wikileaks
Livrable :
• texte de >300 mots



Liste de
25 mots

Textes
sur
Wikileaks

Tâche 2 :
calculer les distances
entre les mots
Livrables :
• un fichier CSV
• le script Python

Matrice de
distance
entre les 25
mots

Matrice de
distance
entre les 25
mots

Tâche 3 :
calculer une partition
des mots
Livrables :
• la partition
• le script Python
• un rapport

Partition
des 25
mots

**Soutenance
courte
(10 min)**

Sources du cours

- Cours de Matthieu Constant, *Ingénierie Informatique 1*

<http://igm.univ-mlv.fr/ens/Master/M1/2010-2011/IngenierieLinguistique1/cours.php>

- Cours de Jean Véronis, *Informatique et Linguistique 1*

<http://sites.univ-provence.fr/~veronis/cours/INFZ18/veronis-INFZ18.pdf>

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Textes

Un texte est une séquence de caractères

- lettres : abABéàûï
- chiffres : 1842
- séparateurs : espace, tabulation, retour à la ligne
- symboles de ponctuation :., ?
- autres symboles <(>)

Différents encodages

- ASCII, ISO-8859, Latin1
- unicode UTF-8, UTF-16 Little Endian, UTF-16 Big Endian

Analyse linguistique de textes

Différents niveaux d'analyse

1. Segmentation
2. Analyse lexicale
3. Étiquetage morphosyntaxique
4. Analyse syntaxico-sémantique
5. Analyse sémantico-pragmatique

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier. M. Bunton précise que c'est son plus bas niveau depuis 1998.

Segmentation en phrases

Principe

Une **phrase** est délimitée par un symbole de fin de phrase

(ex. symbole de ponctuation, retour à la ligne)

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier. M. Bunton précise que c'est son plus bas niveau depuis 1998.

Segmentation en phrases

Principe

Une **phrase** est délimitée par un symbole de fin de phrase

(ex. symbole de ponctuation, retour à la ligne)

Attention : c'est pas si simple !

M. Bunton précise que c'est son plus bas niveau depuis 1998.

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier. M. Bunton précise que c'est son plus bas niveau depuis 1998.

Segmentation en tokens

Tokenisation

- Découpage d'un texte en tokens
- Un **token** = un mot (séquence de lettres), un nombre, un symbole de ponctuation, ...

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier.

→ [Le | cours | de | l' | or | a | baissé | de | 10 | euros | lundi | dernier]

Analyse morphosyntaxique

Analyse lexicale

- Assigner à chaque token, l'ensemble de ses catégories grammaticales possibles
- Catégories grammaticales : nom (N), verbe (V), adjectif (A), adverbe (Adv), déterminant (D), préposition (P), conjonction de coordination (CC), pronom (Pro), ...

Exemple

Le	cours	de	l'	or	a	baissé	de	10	euros	lundi	dernier
D	N	D	D	N	V	V	D	Num	N	N	N
Pro	V	P	Pro	CC	N		P				A

Analyse morphosyntaxique

Analyse lexicale

- Assigner à chaque token, l'ensemble de ses catégories grammaticales possibles
- Catégories grammaticales : nom (N), verbe (V), adjectif (A), adverbe (Adv), déterminant (D), préposition (P), conjonction de coordination (CC), pronom (Pro), ...

Etiquetage grammatical

- Assigner à chaque token sa catégorie grammaticale dans le contexte de la phrase

Exemple

Le	cours	de	l'	or	a	baissé	de	10	euros	lundi	dernier
D	N	P	D	N	V	V	P	Num	N	N	A
le	cours	de	le	or	avoir	baisser	de	10	euro	lundi	dernier

Analyse morphosyntaxique

Analyse lexicale

- Assigner à chaque token, l'ensemble de ses catégories grammaticales possibles
- Catégories grammaticales : nom (N), verbe (V), adjectif (A), adverbe (Adv), déterminant (D), préposition (P), conjonction de coordination (CC), pronom (Pro), ...

Etiquetage grammatical, lemmatisation

- Assigner à chaque token sa catégorie grammaticale dans le contexte de la phrase, en déduire le **lemme** du token

Exemple

Le	cours	de	l'	or	a	baissé	de	10	euros	lundi	dernier
D	N	P	D	N	V	V	P	Num	N	N	A
le	cours	de	le	or	avoir	baisser	de	10	euro	lundi	dernier

formes fléchies

lemmes

Analyse syntaxico-sémantique

Analyse syntaxique de surface

- Identification des constituants syntaxiques simples (ou chunks)
- Types de chunks : groupes nominaux (XN), groupes prépositionnels (XP), complexes verbaux (XV), groupes adverbiaux (XADV), ...

Exemple

Le cours	de l' or	a baissé	de 10 euros	lundi dernier
XN	XP	XV	XP	XADV+date
cours	or	baisser	euro	lundi_dernier

Analyse syntaxico-sémantique

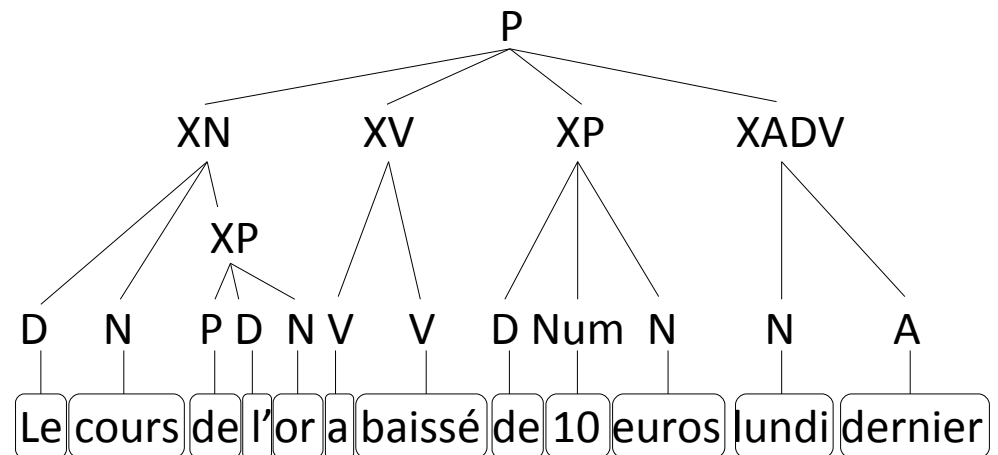
Analyse syntaxique en profondeur

- Construction de l'arbre syntaxique de la phrase

→ reconnaissance des constituants syntaxiques et leurs attachements

Exemple

(P
(XN Le cours (XP de l'or XP) XN)
(XV a baissé XV)
(XP de 10 euros XP)
(XADV lundi dernier XADV)
P)



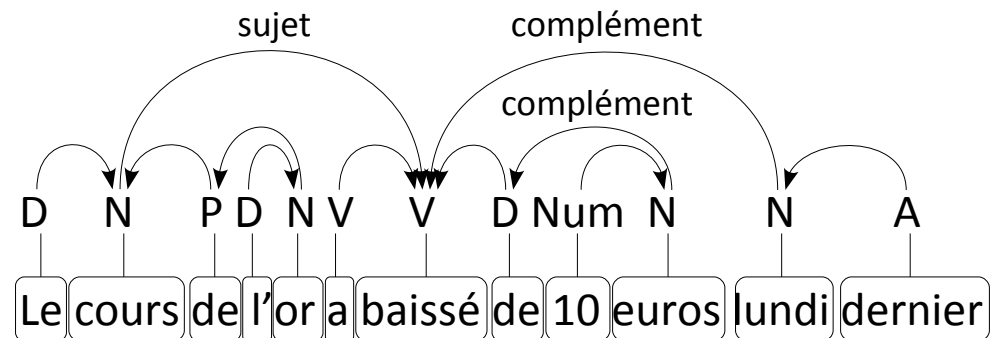
Analyse syntaxico-sémantique

Analyse syntaxique en profondeur

- Construction de l'analyse syntaxique en dépendance de la phrase
→ reconnaissance des dépendances des constituants syntaxiques

Exemple

(P
(XN Le cours (XP de l'or XP) XN)
(XV a baissé XV)
(XP de 10 euros XP)
(XADV lundi dernier XADV)
P)



Analyse syntaxico-sémantique

Analyse sémantique

- Analyse du sens

→ représentation prédicat-argument des phrases

Exemple

BAISSER(COURS,"10 euros")

COURS("or")

SE_DEROULER(BAISSER,"lundi dernier")

Analyse syntaxico-sémantique

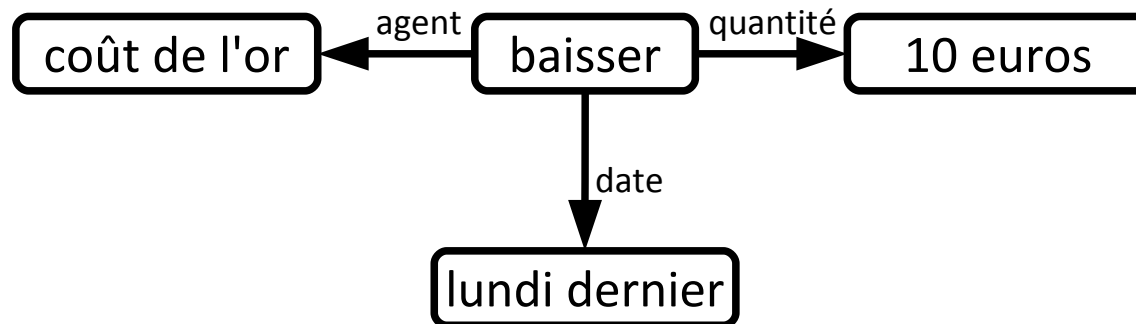
Analyse sémantique

- Analyse du sens

→ représentation prédicat-argument des phrases

→ représentation en réseau sémantique des phrases

Exemple



Analyse avancée

Différentes analyses

- Comprendre les phrases dans leur contexte (ex. résolution d'anaphores)
- Comprendre le sens caché (Tu as l'heure ?)

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier. M. Bunton précise que c'est son plus bas niveau depuis 1998.

Analyse avancée

Différentes analyses

- Comprendre les phrases dans leur contexte (ex. résolution d'anaphores)
- Comprendre le sens caché (Tu as l'heure ?)

Exemple

Le cours de l'or a baissé de 10 euros lundi dernier. M. Bunton précise que c'est **son** plus bas niveau depuis 1998.

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- **Plan du cours**
- Langages rationnels
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Plan du cours

1. Introduction - Langages rationnels et ingénierie linguistique
2. Espaces vectoriels et recherche d'informations
3. Classification de documents
4. n-grammes, modèles de Markov et étiquetage grammatical
5. *Analyse syntaxique par grammaires*
6. Introduction à la traduction automatique, alignement de textes parallèles

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- **Langages rationnels**
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Langages rationnels et ingénierie linguistique

Langages rationnels

- Expressions rationnelles
- Automates finis

Application au Traitement Automatique des Langues (TAL)

- En général, pour les tâches présyntaxiques
- Exemples : tokenisation, analyse lexicale, chunks

Langages rationnels et ingénierie linguistique

Éléments de base

- Un alphabet Σ de "lettres"
- Deux opérateurs : union ($|$) et concaténation ($.$)

Langage rationnel : ensemble de mots générés à partir de l'alphabet et des opérateurs

Définition récursive : grammaires linéaires droites

Exemple :

$$\Sigma = \{a, b, \dots, z\}$$

$$L = b.o.n.(\varepsilon | n.e).(\varepsilon | s) = \{\text{bon, bons, bonne, bonnes}\}$$

ε : le symbole vide

Langages rationnels et ingénierie linguistique

Éléments de base

- Un alphabet Σ de "lettres"
- Deux opérateurs : union ($|$) et concaténation ($.$)

Langage rationnel : ensemble de mots générés à partir de l'alphabet et des opérateurs

Définition récursive : **grammaires linéaires droites**

Exemple :

$\Sigma = \{a,b,\dots,z\}$

$L = b.o.n.(\epsilon | n.e).(\epsilon | s) = \{\text{bon,bons,bonne,bonnes}\}$

ϵ : le symbole vide

$A \rightarrow bB$

$B \rightarrow oC$

$C \rightarrow nD$

$C \rightarrow n$

$D \rightarrow nE$

$D \rightarrow F$

$E \rightarrow eF$

$E \rightarrow e$

$F \rightarrow s$

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- **Expressions rationnelles**
- Automates et transducteurs
- Utilisation en TAL

Expressions régulières

<http://xkcd.com/208>
<http://xkcd.free.fr/?id=208>

A CHAQUE FOIS QUE JE DÉCOUVRE UNE NOUVELLE TECHNIQUE, J'IMAGINE DES SITUATIONS COMPLEXES OU ÇA ME PERMETTRAIT D'ÊTRE UN HÉROS.

OH NON ! LE TUEUR A DÛ LA SUIVRE SUR SON LIEU DE VACANCES !

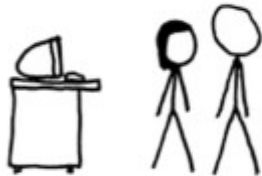


POUR LES RETROUVER, IL FAUDRAIT RECHERCHER CE QUI RESSEMBLE À UNE ADRESSE PARMIS SES 200 Mo DE MAILS !



C'EST SANS ESPOIR !

Laissez-moi faire.



JE CONNAIS LES EXPRESSIONS RÉGULIÈRES.



xkcd.com,
traduction P. Gambette

Oups, j'ai oublié d'échapper un caractère espace. Fuiiiiiiiiiii-tap-tap-tap-iiiiiii

Expressions régulières

Définition informelle

Motifs qui permettent de reconnaître des séquences appartenant à un langage rationnel

Peuvent se définir récursivement comme les langages rationnels

Exemple :

$$\Sigma = \{a, b, \dots, z\}$$
$$\text{Regexp} = \text{ajout}(e \mid e(s \mid z \mid nt) \mid \text{ons})$$

Expressions régulières en Python

Caractères spéciaux

- `[...]` : spécification de classe de caractères
- `.` : la classe de caractère prédéfinie des caractères graphiques visibles ou blancs ou de contrôle (sauf saut de ligne)
- `*` : quantificateur pour zéro, une ou plusieurs occurrences de ce qui précède
- `?` : quantificateur pour au plus une occurrence de ce qui précède
- `|` : alternative, soit ce qui précède soit ce qui suit
- `()` : délimiteurs de groupe (avec capture)

Désécialisation des caractères spéciaux avec `\`

- `\w` : un caractère lettre ou chiffre
- `\W` : un caractère ni lettre, ni chiffre, le complément de `\w`
- `\t` : tabulation horizontale
- `\n` : saut de ligne
- `\s` : un élément de l'ensemble `[\t \n \r \f]`
- `\S` : tout élément non compris dans l'ensemble `[\t \n \r \f]`

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- Expressions rationnelles
- **Automates et transducteurs**
- Utilisation en TAL

Automates finis

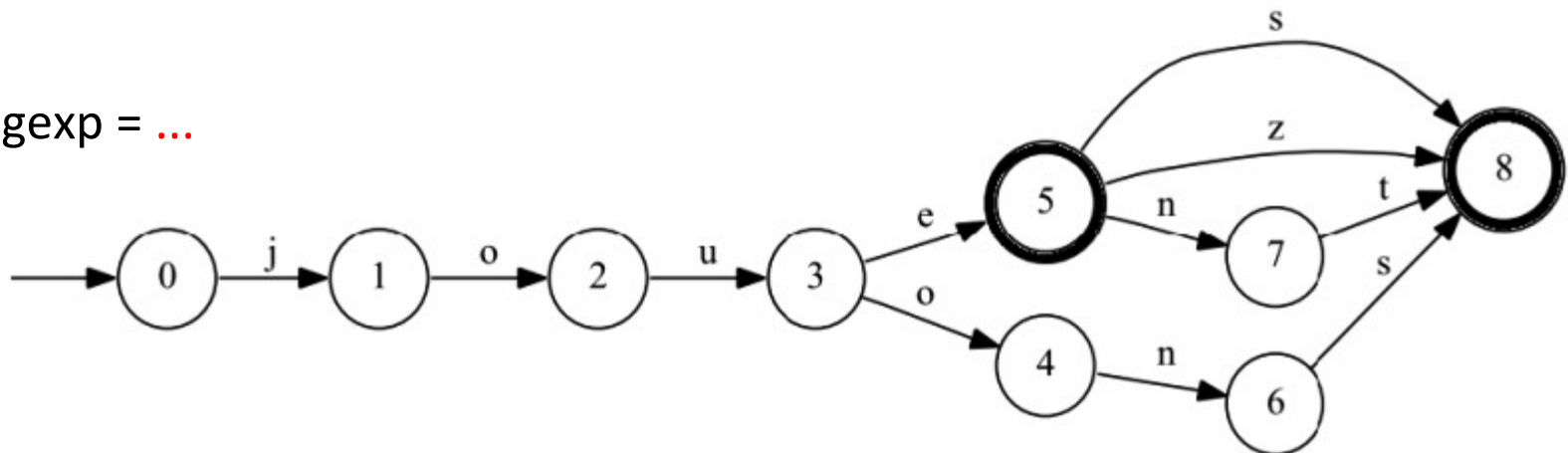
Définition informelle

- Un alphabet
- Un ensemble d'états
- Transitions allant d'un état à un autre, étiquetées par une lettre de l'alphabet
- États "spéciaux" : initiaux et finaux

Reconnaissance

- ensemble de mots reconnus par un automate : langage rationnel
- mot reconnu par un automate s'il existe un chemin de l'automate allant d'un état initial à un état final, étiqueté par les lettres du mot (dans l'ordre)

Regexp = ...



Automates finis

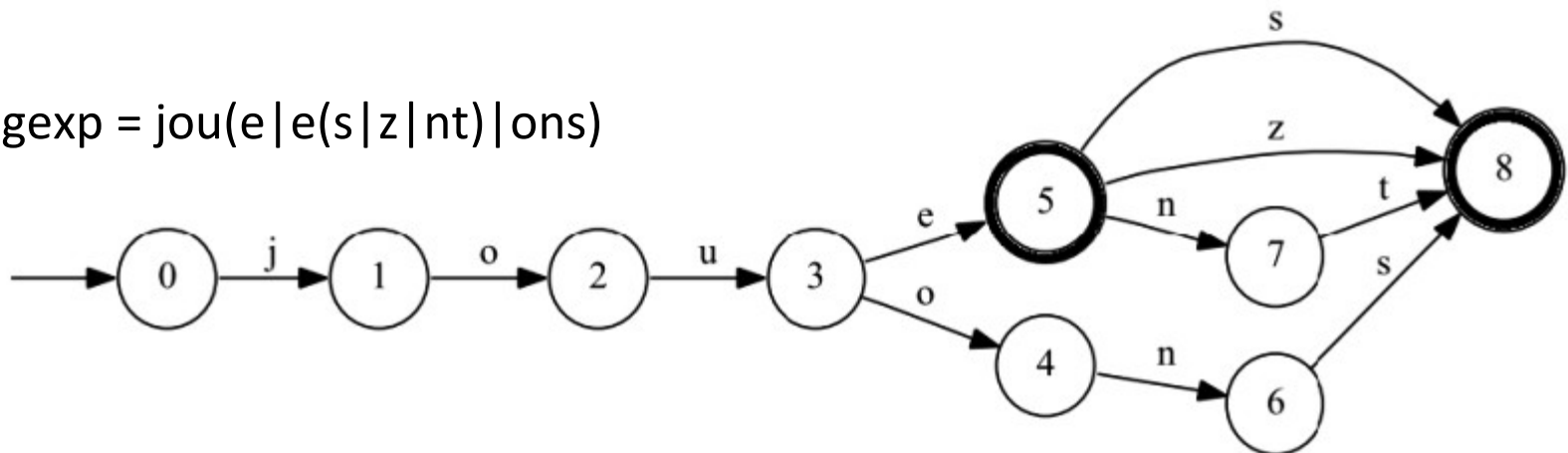
Définition informelle

- Un alphabet
- Un ensemble d'états
- Transitions allant d'un état à un autre, étiquetées par une lettre de l'alphabet
- États "spéciaux" : initiaux et finaux

Reconnaissance

- ensemble de mots reconnus par un automate : langage rationnel
- mot reconnu par un automate s'il existe un chemin de l'automate allant d'un état initial à un état final, étiqueté par les lettres du mot (dans l'ordre)

Regexp = `jou(e|e(s|z|nt)|ons)`



Automate déterministe minimal

Automate déterministe

- Un seul état initial
- De chaque état, ne sort au plus qu'une seule transition étiquetée par un symbole donné de l'alphabet
- Pas de ε -transition

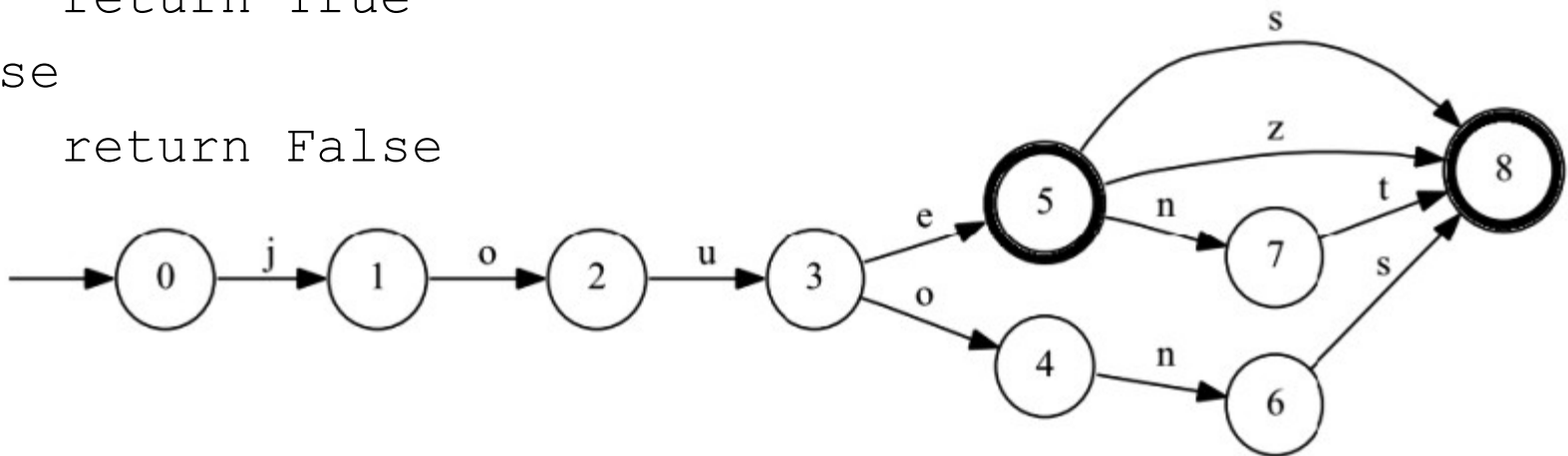
Automate déterministe minimal

Automate déterministe reconnaissant un langage rationnel L tel qu'il n'existe pas d'automates finis déterministes avec moins d'états reconnaissant le même langage.

Reconnaissance d'une séquence par un automate

- Entrées : séquence de lettres (sequence) ; automate fini (dfa)
- Sortie : booléen

```
def isRecognized(sequence, dfa) :  
    state = dfa.getInitialState()  
    for symb in sequence :  
        state = dfa.getNextState(state, symb)  
        if state is None :  
            return False  
    if state.isFinal() :  
        return True  
    else  
        return False
```



Automates et expressions rationnelles

Equivalence

A partir d'un automate fini, il est possible de construire une expression rationnelle reconnaissant le même langage rationnel (et inversement).

En pratique

Expression rationnelle \rightarrow automate

Exemple :

$bo(n|ns|nne|nnes)$

Automates et expressions rationnelles

Equivalence

A partir d'un automate fini, il est possible de construire une expression rationnelle reconnaissant le même langage rationnel (et inversement).

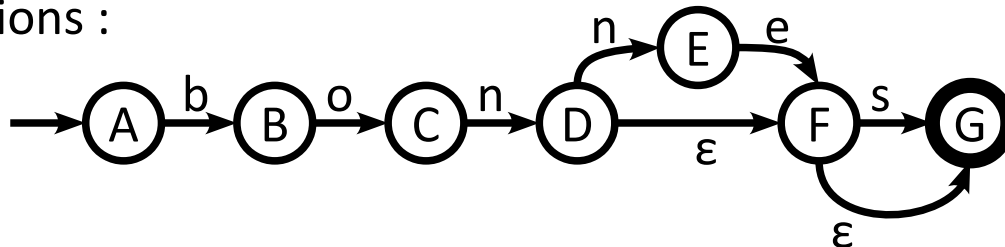
En pratique

Expression rationnelle \rightarrow automate

Exemple :

$bo(n | ns | nne | nnes)$ $L = b.o.n.(\epsilon | n.e).(\epsilon | s) = \{\text{bon, bons, bonne, bonnes}\}$

Avec ϵ -transitions :



Automates et expressions rationnelles

Equivalence

A partir d'un automate fini, il est possible de construire une expression rationnelle reconnaissant le même langage rationnel (et inversement).

En pratique

Expression rationnelle \rightarrow automate

Exemple :

$bo(n|ns|nne|nnes)$

Sans ϵ -transitions :

Automates et expressions rationnelles

Equivalence

A partir d'un automate fini, il est possible de construire une expression rationnelle reconnaissant le même langage rationnel (et inversement).

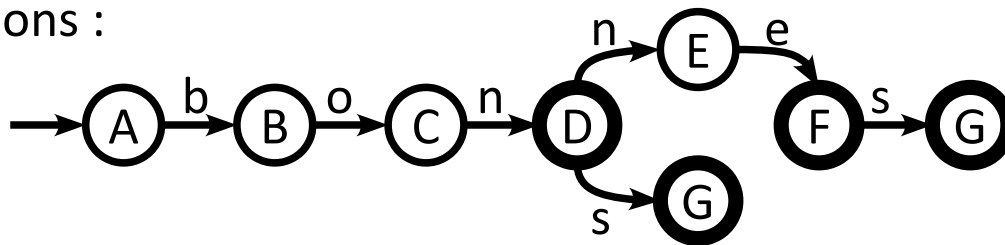
En pratique

Expression rationnelle \rightarrow automate

Exemple :

$bo(n | ns | nne | nnes)$

Sans ϵ -transitions :



Transducteurs finis

Extension des automates

Mais : deux alphabets (entrée/sortie), et étiquettes sur les transitions de type entrée/sortie.

On reconnaît une séquence avec les entrées, on **produit les sorties associées**.

Attention : pas toujours les mêmes propriétés que les automates (ex. déterminisme)

Plan

- Aspects administratifs
- Introduction à l'ingénierie linguistique
- Plan du cours
- Langages rationnels
- Expressions rationnelles
- Automates et transducteurs
- Utilisation en TAL

Tokenisation

Principe :

Découper un texte en tokens.

Produire la liste des tokens d'un texte.

La définition d'un token dépend de l'application

Utilisation d'expressions rationnelles :

Exemple : `/\w+/`

En Python, on utilise le module NLTK

Tokenisation

```
from nltk import word_tokenize, wordpunct_tokenize
s = ("Good muffins cost $3.88\nin New York. Please
buy me\n two of them.\n\nThanks.")
t = regexp_tokenize(s, pattern='\w+|\$[\d\.]+|\S+')
```

t contient :

```
['Good', 'muffins', 'cost', '$3.88', 'in', 'New',
'York', '.', 'Please', 'buy', 'me', 'two', 'of',
'them', '.', 'Thanks', '.']
```


Tokenisation

```
# tokenisation du texte /tmp/test.txt
# encode en utf-8
import nltk
import codecs

f = codecs.open('/tmp/test.txt', 'r', 'utf-8')
text = f.read().lower()
f.close()
l = nltk.regexp_tokenize(text, "\w+")
print l
```

Représentation de dictionnaires

- Représenter un **ensemble de mots du français**
(solution = automate)
- Donner à ces mots une **catégorie grammaticale**
(solution = transducteur)
- Donner à ces mots une **phonétisation**
(solution = transducteur)
- Représenter des **expressions de date**
(solution = automate)

Exemple : Jean a rendez-vous avec Luc le 17 avril 2010

Représentation de dictionnaires

Thèse de Dominique Revuz (1991) :

- Dictionnaire DELAF : 600 000 mots
- Représentation par arbre : 2 000 000 noeuds
- Représentation par automate (fusion des terminaux sans successeurs) : 1 000 000 états
- Représentation par automate minimal : 50 000 états

Indexation

Indexation simple

But : trouver les documents contenant certains mots

→ À chaque mot, on associe l'ensemble des documents où il apparaît.

Indexation complexe

But : trouver les documents contenant une séquence de mots

→ À chaque mot, on associe l'ensemble des documents où il apparaît, ainsi que ses positions.

Exercice

Créer l'expression régulière en Python qui extrait d'une chaîne de caractères contenant une adresse : le *numéro*, le *type de voie*, le *nom de voie*

Exemple d'adresses :

- 8 rue des Prés Jefsons
- 123, rue Saint-Jacques
- 66, rue Camille Desmoulins
- 240 avenue de Lodève
- 161 rue Ada
- Place Eugène Bataillon
- 49 ter rue Haguenot
- 78 rue Sénac-de-Meilhan
- 163 av de Luminy
- 5, bd Descartes