

Introduction au Traitement Automatique des Langues

4 – Les niveaux de traitement – Le niveau Lexical

Introduction au traitement automatique des langues

Contenu de la matière :

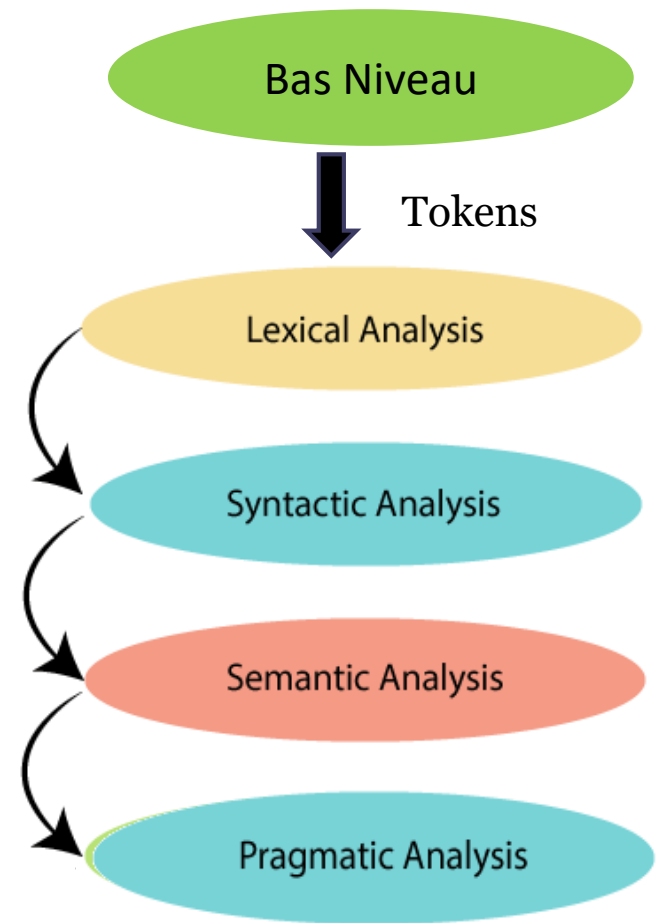
- 1) Introduction Générale
- 2) Les applications du TAL
- 3) Les niveaux de traitement - Traitements de «bas niveau»
- 4) Les niveaux de traitement - Le niveau lexical**
- 5) Les niveaux de traitement - Le niveau syntaxique
- 6) Les niveaux de traitement - Le niveau sémantique
- 7) Les niveaux de traitement - Le niveau pragmatique

Plan du cours

1. Définitions
2. Nature de mots (POS) et Trait grammatical
3. Morphologie
4. Morphèmes et leurs typologie
5. Morphologie flexionnelle
6. Morphologie dérivationnelle
7. En Pratique - Prétraitement et Normalisation
8. Lemmatisation et Stemming - Algorithmes

Définitions

- L'analyse lexicale comprend l'**identification** et l'**analyse** de la **structure** des **tokens** dans les phrases.
- Le but de cette étape de traitement lexicale est de passer des formes atomiques (**tokens**) identifiées par le segmenteur aux mots, c'est-à-dire de **reconnaître** dans chaque chaîne de caractère une (ou plusieurs) unité(s) linguistique(s), dotée(s) de **caractéristiques** propres.



Définitions

- Identifier les composants lexicaux, et leurs propriétés : c'est l'étape de **traitement lexical**
- Exemple énoncé : *Le président des conseils mangeait une pomme.*

le - **det.** masc. sing.; / **pron.** pers. masc. sing.

président - **vr̥b** 3pers. plur. prés. ind.;/ subjonctif ;/ nom masc. sing.

des - **det.** masc./fem. plur. ; / **prep.** contr. de les.

conseils - **nom.** masc. plur.

mangeait - **vr̥b** (1,3) pers. sing. imp. ind., [mang+e+ait].

pomme - **nom** fem. sing.

Trait grammatical

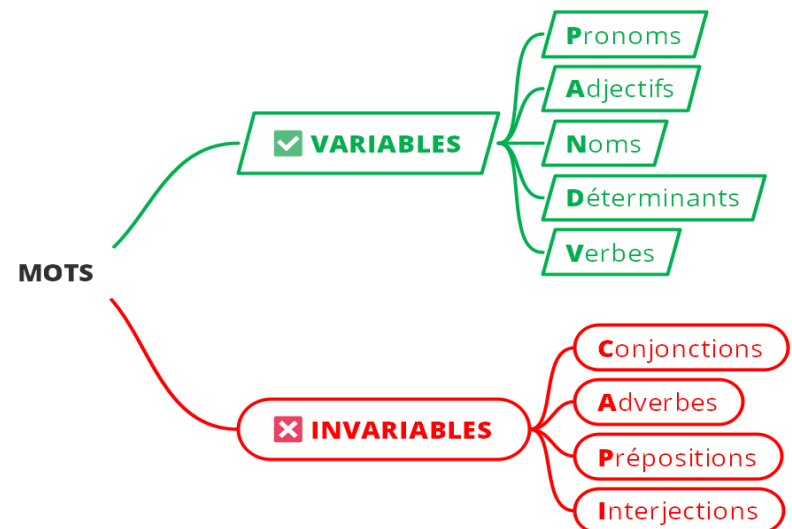
- Le **trait grammatical** est une **catégorie d'ordre grammatical** et linguistique permettant de décrire les **flexions morphologiques** des mots variables d'une langue.
- Les traits ne sont pas tous représentés dans toutes les langues, et chaque trait se subdivise en catégories de nombre variable dans les langues concernées.
- Les principaux traits grammaticaux sont les suivants :
 - la **nature** ou classe grammaticale (nom, verbe, adjectif, etc.) ;
 - le **genre** (masculin, féminin, neutre, etc.) ;
 - le **nombre** (singulier, pluriel, etc.) ;
 - la **personne** (1re, 2e et 3e) ;
 - le **cas** (nominatif, accusatif, régime, etc.) ;
 - le **temps** (imparfait, présent, futur antérieur, etc.) ;
 - le **mode** (indicatif, subjonctif, infinitif, etc.) ;
 - la **fonction** (sujet, épithète, COD, etc.) ;
 - la **voix** (active, passive) ;

Nature d'un mot - Part of speech

- La **nature d'un mot** est la **catégorie grammaticale** de mots à laquelle il appartient. En Anglais : part-of-speech (POS).
- Une catégorie de mots est la réunion de mots d'un certain type, ayant des traits grammaticaux en commun.
- Catégories grammaticale, Lexicale, Classes grammaticale, etc :

❖ English / Français :

- Noun (names)
- Pronoun (replaces)
- Adjective (describes, limits)
- Verb (states action or being)
- Adverb (describes, limits)
- Preposition (relates)
- Conjunction (connects)
- Article (describes, limits)
- Interjection (expresses feelings and emotions)



Nature d'un mot

- Catégorie moderne: Mots pleins (**open-class**) et mots outils (**closed-class**)
- ❖ **Classe ouverte**, mots pleins, est celle qui accepte généralement l'ajout de nouveaux mots.
- ❖ La classes ouverte contiennent normalement un grand nombre de mots.
- ❖ On retrouve : les **noms** , les **verbes** (à l'exclusion des verbes auxiliaires , s'ils sont considérés comme une classe distincte), les **adjectifs** , les **adverbes** et les **interjections**.
- **Classe fermée**, mots outils, est celle à laquelle de nouveaux éléments sont très rarement ajoutés.
- La classe fermée est beaucoup plus petites et contient un nombre limité de mots.
- On retrouve : les **prépositions** (ou postpositions), les **déterminants** , les **conjonctions**, et les **pronoms** .

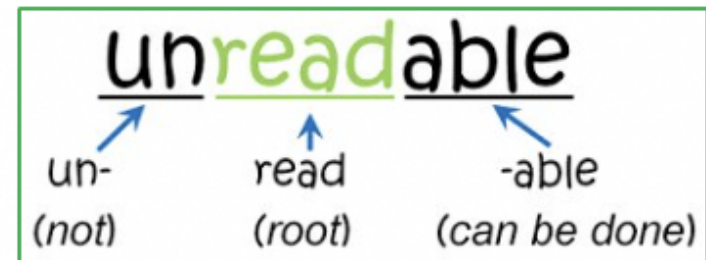
Définitions

- **Comment** cette identification est-elle réalisée ?
- La solution la plus simple pour les mots/tokens les plus fréquents est de rechercher la forme dans un **lexique** pré-compilé : **Accès lexical direct**.
- Cette solution ne résout pas tous les problèmes. Le langage est création, et de nouvelles formes/tokens surgissent tous les jours.
- Mise en œuvre d'autres **approches**, de manière à traiter aussi les formes hors-lexique.
- ➔ **Morphologie** : « étude des *formes* sous lesquelles se présentent les *mots* dans une langue, des changements dans la forme des mots pour exprimer leurs relations à d'autres mots de la phrase, des processus de *formation de mots* nouveaux, etc. ». G. Mounin
- Analyse lexicale ou morphologique.

Morphologie

➔ **Morphologie** : La morphologie est la branche de la linguistique et de la grammaire qui s'intéresse à la **formation** du mot.

- Exemple : Pour comprendre une forme telle que *Parisien*, être capables de reconnaître dans cette forme des **composants** plus **petits**, nommément une **racine**, Paris, qui réfère au nom d'une ville, et un **suffixe**, ien, qui permet de manière régulière de construire des **adjectifs** à partir de **noms** propres.
- La linguistique traditionnelle appelle ces composants plus petits les **morphèmes**, et l'étude de leurs combinaisons la morphologie.
- Morphème: le plus petit élément significatif individualisé dans un énoncé, isolé par segmentation d'un mot.



Morphèmes et leurs typologie

- **Morphème**: le plus petit élément significatif individualisé dans un énoncé, isolé par segmentation d'un mot. Concept de base en morphologie.
- Par exemple, le mot *chanteurs* est composé de trois morphèmes : *chant-* « chant », *-eur-* « celui qui fait » et *-s* (marque du pluriel).
- Exemple 2 : *base de données* n'est pas composé de trois morphèmes mais bien d'un seul morphème qui contient la signification « base de données ».
- Typologie : **Morphèmes lexicaux** ou **Morphèmes grammaticaux**.
- ❖ Les morphèmes **lexicaux** sont en nombre illimité (ouverte), tels que *lave*, *vite*, *lune*, etc. Il s'agit de noms, adjectifs, verbes ou adverbes.
- ❖ Les morphèmes **grammaticaux** sont en nombre limité (fermée), tels que *tu*, *à*, *et*, etc. Il s'agit de pronoms, prépositions, conjonctions, déterminants, affixes : des listes de mots qui ne varient pratiquement jamais.

Morphèmes et leurs typologie

- Typologie : **Morphèmes liés** ou **Morphèmes libres**.
- ❖ **Lié** s'il ne se manifeste pas comme lemme et n'existe jamais à l'état libre mais est toujours rattaché à un autre morphème appelé base.
 - Ex : -ons dans ouvr-ons, ou re- dans re-faire, ou un radical comme -cevoir (re-cevoir, per-cevoir, dé-cevoir, etc.) qui n'existe pas non plus à l'état libre.
- ❖ **Libre** s'il peut constituer un mot à lui seul: le ou beau sont libres.

Morphologie

- Deux **types** de **morphologie**:

1) La morphologie **flexionnelle** (en interne) التصريف: les processus d'**ajustement** et de **variation** de mots imposés par les conditions/traits grammaticaux d'utilisation du mot, sans changer son sens ou sa catégorie grammaticale. Ex :

- étudiant => étudiants (**pluriel**)
- Petit => petite (**féminin**)

– أكل = < يأكل، أكلنا
– طالب = < طالبة، طالبة

2) La morphologie **dérivationnelle** (en externe) الاشتقاق: les processus de **création** de nouveaux mots à partir de mots existants. Les processus dérivationnels entraînent le plus souvent un changement de la catégorie grammaticale: un nom se transforme en verbe, un verbe en adjectif. Ex:

- Jouer => joue**ur**

– طلب = < طالب، مطلوب

Morphologie flexionnelle

1 - La morphologie flexionnelle:

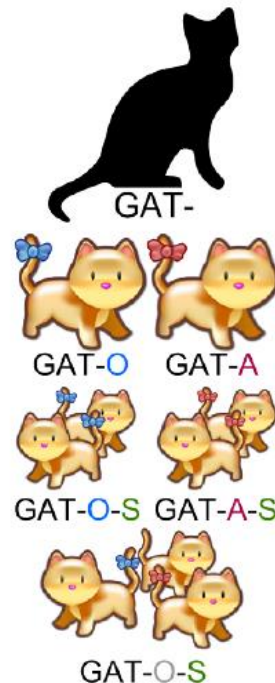
- En morphologie, on nomme **flexion** l'ensemble des modifications subies aux mots d'une langue flexionnelle pour dénoter les traits grammaticaux voulus.
- À la différence de la dérivation, la flexion ne crée pas de nouveaux mots, mais différentes formes d'un même mot.
- Il existe **2 grandes catégories de flexions**: nominale (et adjectivale) et verbale.
- ❖ La déclinaison pour le système nominal: Les noms y changent généralement de forme selon le **genre**, le **cas**, ou le **nombre**;
- ❖ La conjugaison pour les verbes: qui varient généralement en **personne**, **nombre**, **temps**, **voix**, et **mode**.

Morphologie flexionnelle

1 - La morphologie flexionnelle:

Exemple de déclinaison

La déclinaison de
mot chat en
espagnole



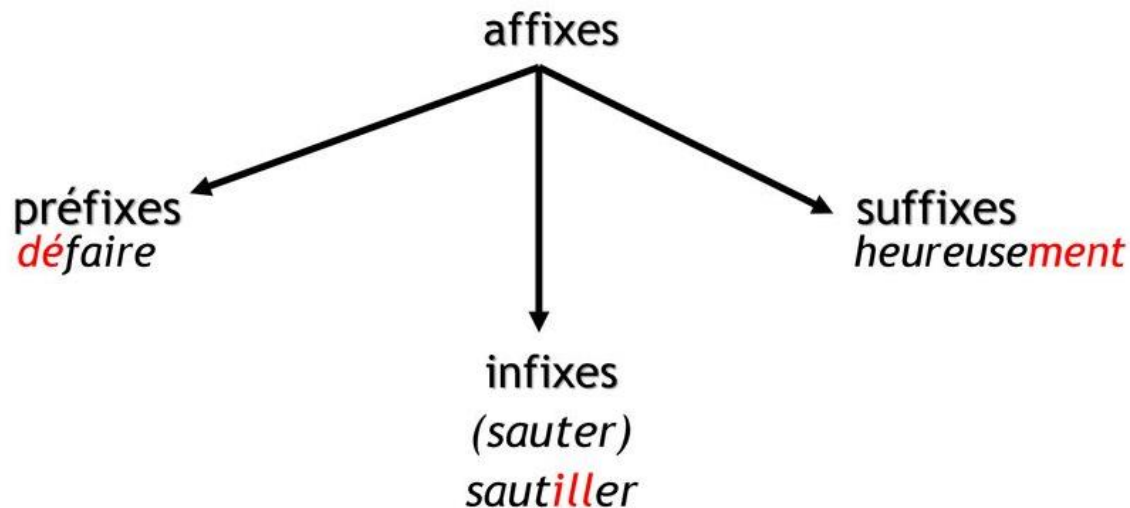
Conjugaison

الماضي	• المضارع
أنا كُتِبْتُ	• أنا أَكْتُبُ
أنت كُتِبْتَ	• أنت تَكْتُبُ
أنت كُتِبْتَ	• أنت تَكْتُبِينَ
هو كُتِبَ	• هو يَكْتُبُ
هي كُتِبَتْ	• هي تَكْتُبُ

Morphologie flexionnelle

1 - La morphologie flexionnelle:

- La génération des nouvelles formes se fait par les méthodes suivantes :
 - **L'affixation**: l'ajout des affixes (préfixes, infixes, et des suffixes)
 - ✓ Préfixes, placés avant leur base
 - ✓ Infixes, placés à l'intérieur d'une base, ou entre une base et un affixe d'un terme déjà lexicalisé
 - ✓ Suffixes, placés après la base



Morphologie flexionnelle

1 - La morphologie flexionnelle:

- La génération des nouvelles formes se fait par les méthodes suivantes :
 - **Duplication** : Dupliquer le mot ou bien une partie de mot. Ex:

✓ زلزل

✓ الشدة

✓ كيف كيف

✓ Ping-pong

✓ Bye-bye

✓ zigzag

Morphologie flexionnelle

1 - La morphologie flexionnelle:

- La génération des nouvelles formes se fait par les méthodes suivantes :
 - **Altération** : Changer une lettre ou bien une partie du mot. Ex :
 - ✓ Franc => Franche
 - ✓ Verlan : pourri > ripou
 - **Variation super segmentale** : Changer l'intonation du mot
 - ✓ رايب
 - ✓ Azekka

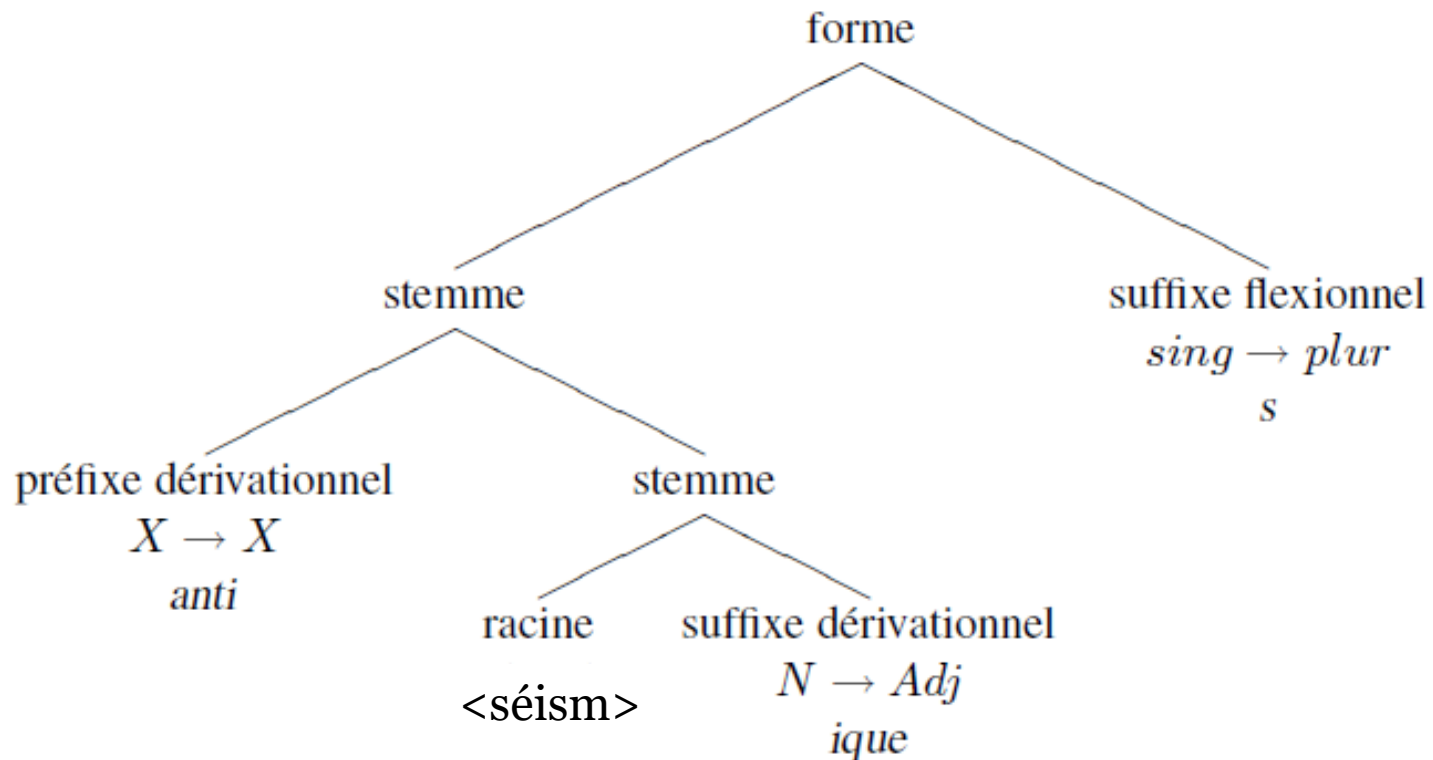
Morphologie dérivationnelle

2 - La morphologie dérivationnelle:

- La dérivation lexicale, ou encore dérivation, sert à créer des nouveaux mots ; on peut former de mots nouveaux en ajoutant des **morphèmes dérivationnels** à des mots existants.
- Elle affecte la **signification** du mot et à l'occasion **la catégorie grammaticale**.
- Ex : danser, dans-eur
- Ex : إعلام, استعمال, معلومة, عالم, علم
- Les méthodes de dérivation :
 - ❖ L'affixation: l'ajout des affixes (préfixes et suffixes) et patterns dérivationnels:
Exemples en anglais:
 - adjective-to-noun: -ness (slow → slow**ness**)
 - adjective-to-adverb: -ly (personal → personall**ly**)

Morphologie

- Une décomposition **arborescente** de la forme : **antiséismiques**



Quelques méthodes de formation de mots

= Quelques types de combinaisons de morphèmes :

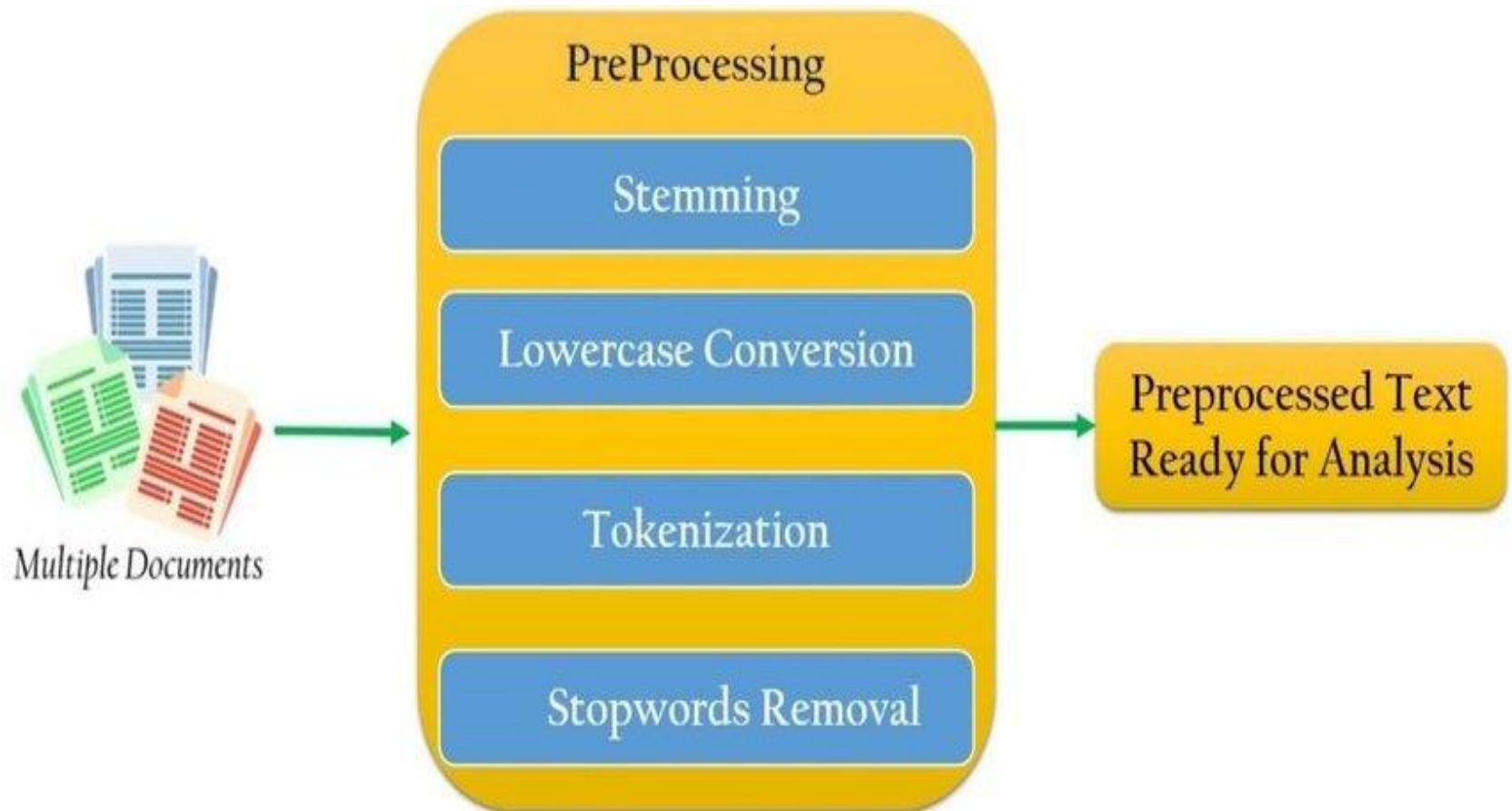
- **Dérivation.**
- **Flexion.**
- **Composition** : permet de jumeler deux items lexicaux (racines) pour en former un nouveau. Ex : Plate-forme, hors-la-loi, Smartphone, en effet, etc.
- **Mot-valise** : consiste à coller le début d'un mot à la fin d'un autre pour créer un nouveau mot. Ex: autobus, franglais, transistor (transfer resistor), etc.
- **Troncation** : permet de raccourcir un item lexical. Ex: fac, ricain, foot, etc.
- **Acronyme** : un sigle passé dans le langage courant : ovni, radar, covid, etc.
- Etc.

L'emprunt lexicale

- Un **emprunt** est un mot ou une expression qu'un locuteur ou une communauté emprunte à une autre langue, sans le traduire, mais en l'adaptant généralement aux règles morphosyntaxiques, phonétiques et prosodiques de sa langue (dite «langue d'accueil»).
- Est le procédé consistant pour une langue à adopter dans son lexique un mot ou une expression d'une autre langue.
- L'emprunt peut être : – **direct** (une langue emprunte directement à une autre langue), – **indirect** (une langue emprunte à une autre langue via une ou plusieurs langues vecteurs).
- Ex : قهوة => kahve => caffè => café
- Ex : Internet
- Ex : Espagnol => Dardja : Suma => “سُومَة”

En Pratique - Prétraitement et normalisation

- Etapes de **prétraitement (PreProcessing)** d'un énoncé textuel :



Prétraitement et normalisation

- La **normalisation du texte** est le processus de transformation du texte en une seule forme canonique qu'il n'aurait peut-être pas eu auparavant.
- Normaliser le texte avant de le stocker ou de le traiter permet de séparer les problèmes, car l'entrée est garantie d'être cohérente avant que les opérations ne soient effectuées sur celui-ci.
- La normalisation de texte nécessite de savoir quel type de texte doit être normalisé et comment il doit être traité par la suite.
- Normaliser les diverses manières d'écrire un même mot; convertir tout le texte en minuscule; corriger les fautes d'orthographe évidentes ou les incohérences typographiques et à expliciter certaines informations lexicales (ex: l' => le/la);
- **lemmatisation; stemming**; etc.

Lemmatisation et Stemming

- **Stemming** consiste à retrouver le **stemme** d'une forme/mot. = la racinisation ou désuffixation, un processus qui consiste à réduire les mots par leur radical (appelés également stemmes, bases ou racines). Pas nécessairement un mot dans le dictionnaire.
- **Lemmatisation** consiste à retrouver le **lemme** d'un mot (i.e. forme canonique / forme dictionnaire d'un mot), c'est-à-dire à lui retirer son ou ses suffixes. Regrouper les mots d'une même famille.
- La lemmatisation est plus informative que le stemming. Mais prend plus de temps à s'exécuter.

Stemming

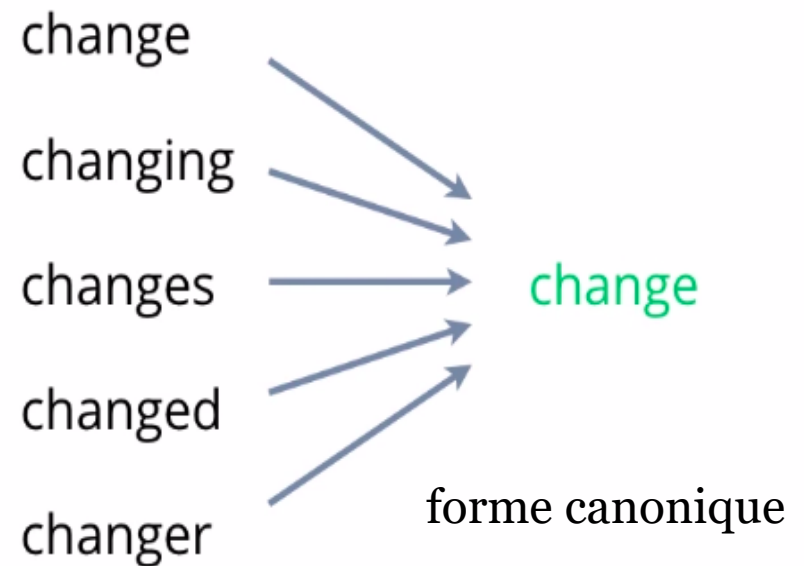
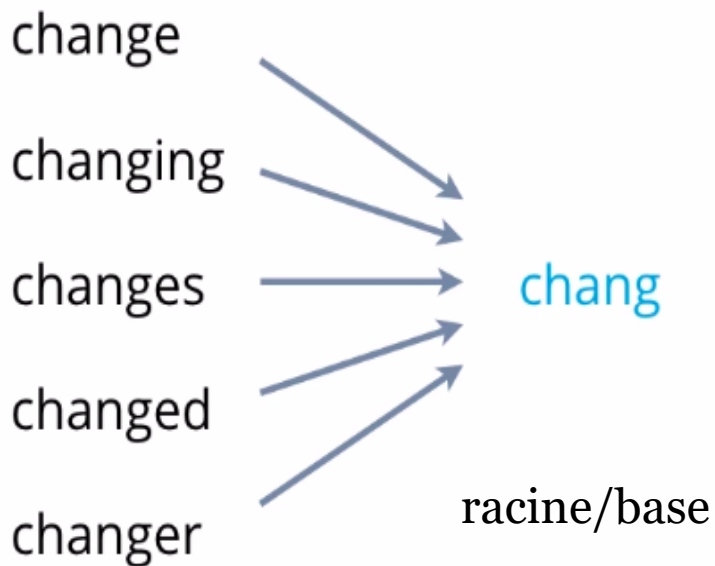
adjustable → adjust
formality → formaliti
formaliti → formal
airliner → airlin △

Lemmatization

was → (to) be
better → good
meeting → meeting

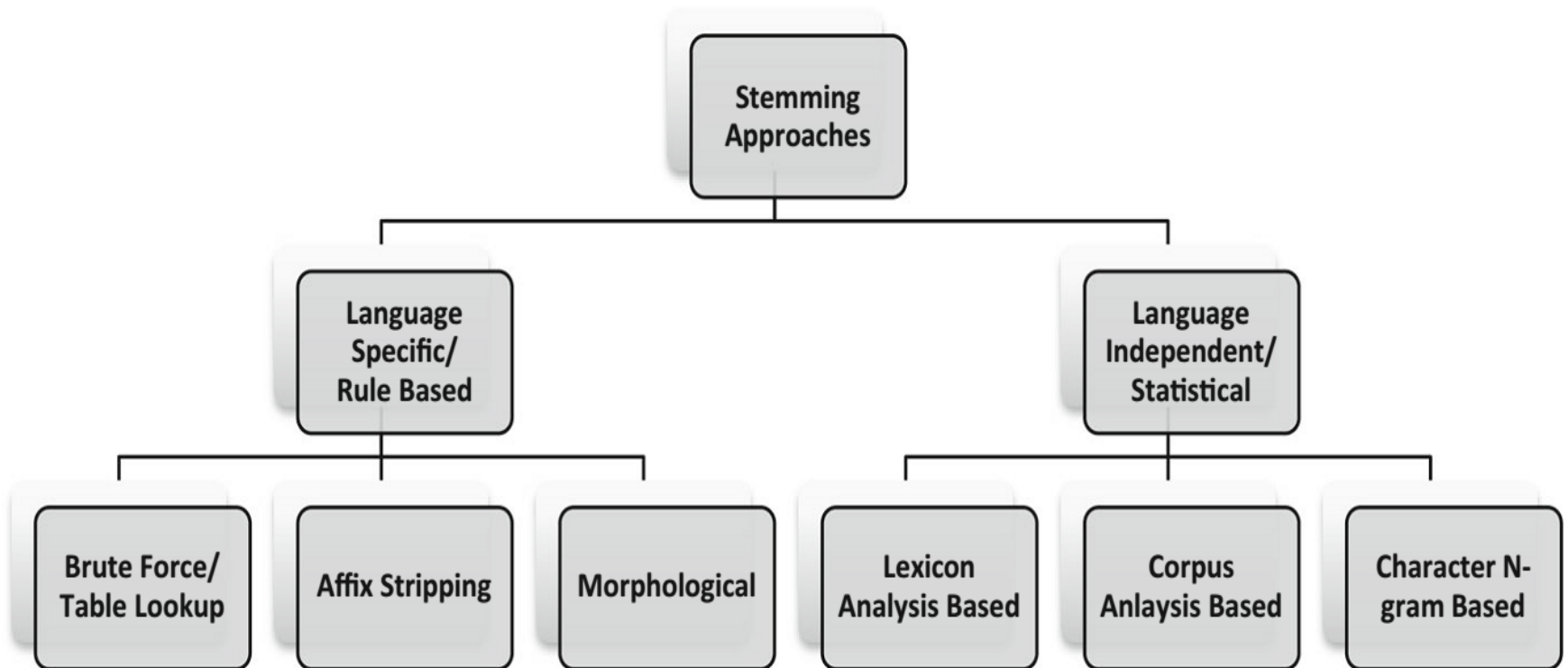
Lemmatisation et Stemming

Stemming vs Lemmatization



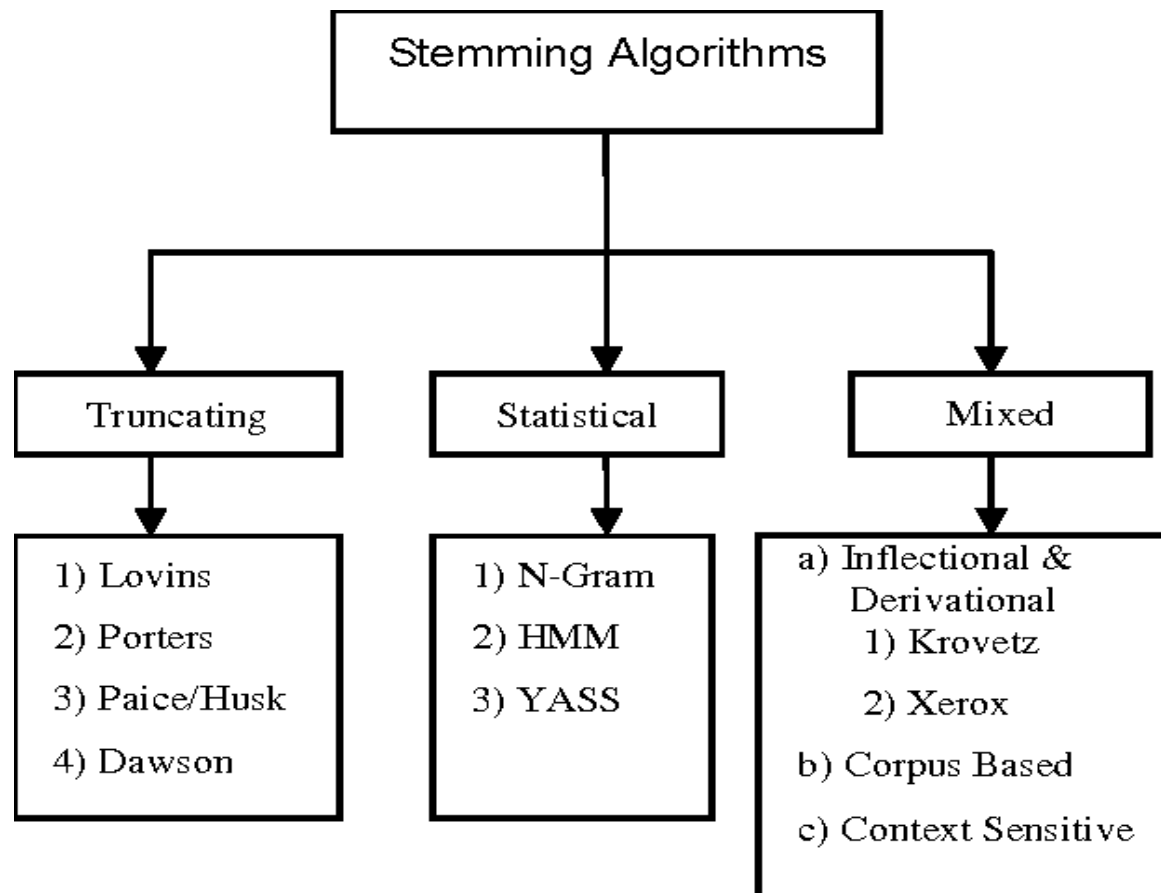
Lemmatisation et Stemming

- **Stemming** : quelques algorithmes et approches – Stemmers



Lemmatisation et Stemming

- **Stemming** : quelques algorithmes et approches – Stemmers



Lemmatisation et Stemming

- **Stemming** : Porter's Stemmer algorithm
- C'est l'un des algorithmes de stemming le plus populaire et le plus utilisé développé en 1980 par Martin Porter.
- Suffix removal algorithm – **Suffix Stripping**. Langue anglaise.
- Il est divisé en un certain nombre **d'étapes** linéaires qui sont utilisées pour produire le stemme final. **Rule based** algorithm.
- Une **consonne** est une lettre autre que A, E, I, O, U, et autre que Y précédé d'une consonne. Ex: Dans le mot 'TOY', les consonnes sont T and Y, et dans le mot 'SYZYG'Y' elles sont S, Z and G.
- Une **voyelle** est une lettre qui n'est pas une consonne.
- Une liste de consonnes supérieures ou égales à la longueur 1 sera désignée par un **C** et une liste similaire de voyelles par un **V**.

Lemmatisation et Stemming

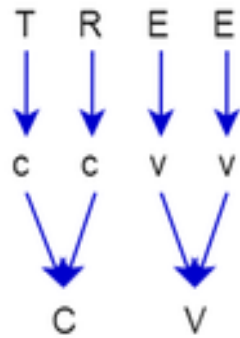
- **Stemming : Porter's Stemmer algorithm**

- N'importe quel mot peut être représenté par la forme unique :

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

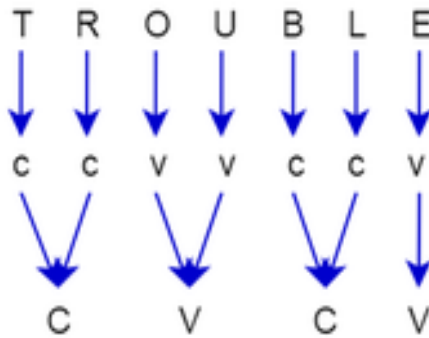
- **m** : désigne m répétitions de VC. La valeur **m** est appelée la **mesure** d'un mot et peut prendre toute valeur supérieure ou égale à zéro, et est utilisée pour décider si un suffixe donné doit être supprimé.
- **[]** dénotent la présence facultative de leur contenu.
- Les règles de suppression de suffixes sont de la forme **(condition) S1 -> S2**.
- Cela signifie que si un mot se termine par le suffixe S1 et que la racine avant S1 satisfait la condition donnée, S1 est remplacé par S2.

[C] (VC)^m [V]



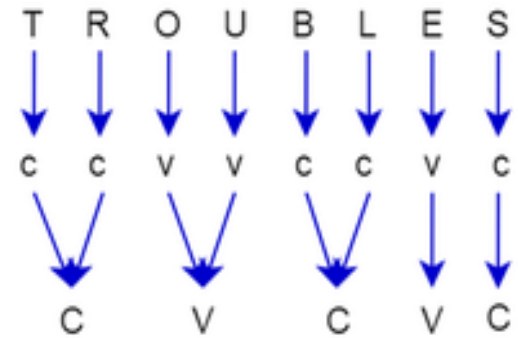
CV

[C] (VC)^m [V] → m = 0



C (VC) V

[C] (VC)^m [V] → m = 1



C (VC) (VC)

[C] (VC)^m [V] → m = 2

Porter Stemming Algorithm

Règle : **(m > 1) EMENT** →
(S1 est «EMENT» et S2 est nul)

REPLACEMENT devient REPLAC,
puisque m = 2 dans la racine avant
S1.

(m > 0) ATIONAL	→	ATE
(m > 0) TIONAL	→	TION
(m > 0) ENCI	→	ENCE
(m > 0) ANCI	→	ANCE

Lemmatisation et Stemming

- **Stemming** : Porter's Stemmer algorithm
- Les **conditions** peuvent contenir : **(condition) S1 -> S2**
 - ✓ ***S** - la racine se termine par S (et de même pour les autres lettres).
 - ✓ ***v*** - la racine contient une voyelle.
 - ✓ ***d** - la racine se termine par une double consonne (par exemple -TT, -SS).
 - ✓ ***o** - la racine se termine par cvc, où le deuxième c n'est pas W, X ou Y (par exemple -WIL, -HOP).
- Et la partie condition peut également contenir des expressions: **and**, **or**, et **not**:
 - ✓ (m> 1 and (*S or*T)) teste une racine avec m> 1 se terminant par S ou T.
 - ✓ (*d and not (*L or*S or *Z)) teste une racine se terminant par une double consonne et ne se terminant pas par les lettres L, S ou Z.
- Dans un ensemble de **règles** écrites, une seule est respectée, et ce sera celle avec le plus long S1 correspondant pour le mot donné.

Porter's Stemmer algorithm - 5 Steps

Step 1a

- 1. SSES → SS
- 2. IES → I
- 3. SS → SS
- 4. S →

Step 1b

- 1. (m>0) EED → EE
- 2. (*v*) ED →
- 3. (*v*) ING →

If the second or third of the rules in Step 1b is successful, the following is performed.

- 1. AT → ATE
- 2. BL → BLE
- 3. IZ → IZE
- 4. (*d and not (*L or *S or *Z)) → single letter
- 5. (m=1 and *o) → E

Step 1c

- 1. (*v*) Y → I

Porter's Stemmer algorithm - 5 Steps

Step 2

1. (m>0) ATIONAL	→	ATE
2. (m>0) TIONAL	→	TION
3. (m>0) ENCI	→	ENCE
4. (m>0) ANCI	→	ANCE
5. (m>0) IZER	→	IZE
6. (m>0) ABLI	→	ABLE
7. (m>0) ALLI	→	AL
8. (m>0) ENTLI	→	ENT
9. (m>0) ELI	→	E
10. (m>0) OUSLI	→	OUS
11. (m>0) IZATION	→	IZE
12. (m>0) ATION	→	ATE
13. (m>0) ATOR	→	ATE
14. (m>0) ALISM	→	AL
15. (m>0) IVENESS	→	IVE
16. (m>0) FULNESS	→	FUL
17. (m>0) OUSNESS	→	OUS
18. (m>0) ALITI	→	AL
19. (m>0) IVITI	→	IVE
20. (m>0) BILITI	→	BLE

Porter's Stemmer algorithm - 5 Steps

Step 3

1. (m>0) ICATE	→	IC
2. (m>0) ATIVE	→	
3. (m>0) ALIZE	→	AL
4. (m>0) ICITI	→	IC
5. (m>0) ICAL	→	IC
6. (m>0) FUL	→	
7. (m>0) NESS	→	

Porter's Stemmer algorithm - 5 Steps

Step 4

1. (m>1) AL →
2. (m>1) ANCE →
3. (m>1) ENCE →
4. (m>1) ER →
5. (m>1) IC →
6. (m>1) ABLE →
7. (m>1) IBLE →
8. (m>1) ANT →
9. (m>1) EMENT →
10. (m>1) MENT →
11. (m>1) ENT →
12. (m>1 and (*S or *T)) ION →
13. (m>1) OU →
14. (m>1) ISM →
15. (m>1) ATE →
16. (m>1) ITI →
17. (m>1) OUS →
18. (m>1) IVE →
19. (m>1) IZE →

Step 5a

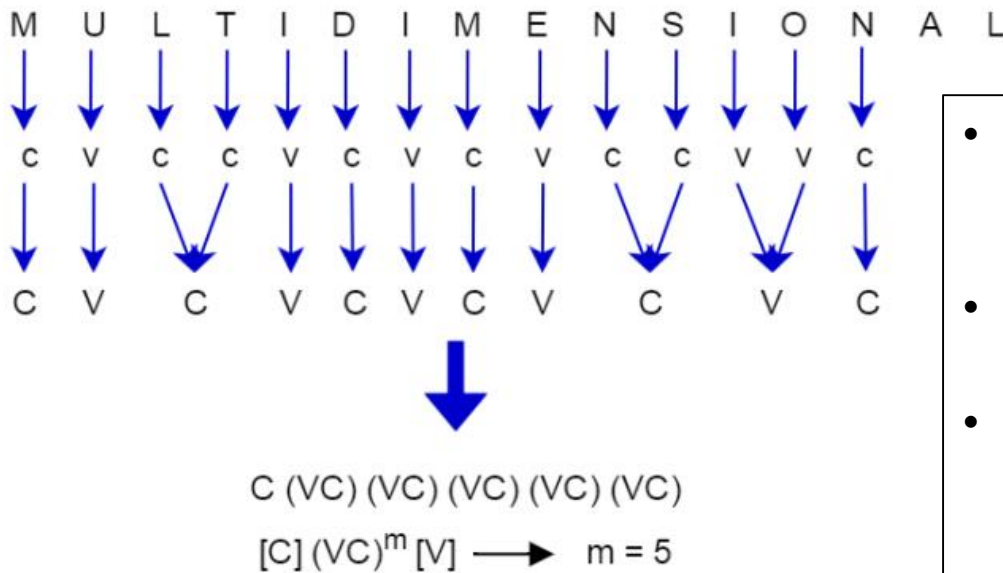
1. (m>1) E →
2. (m=1 and not *o) E →

Step 5b

1. (m > 1 and *d and *L) → single letter

Porter's Stemmer algorithm - 5 Steps

Exemple 1 : mot = **MULTIDIMENSIONAL**



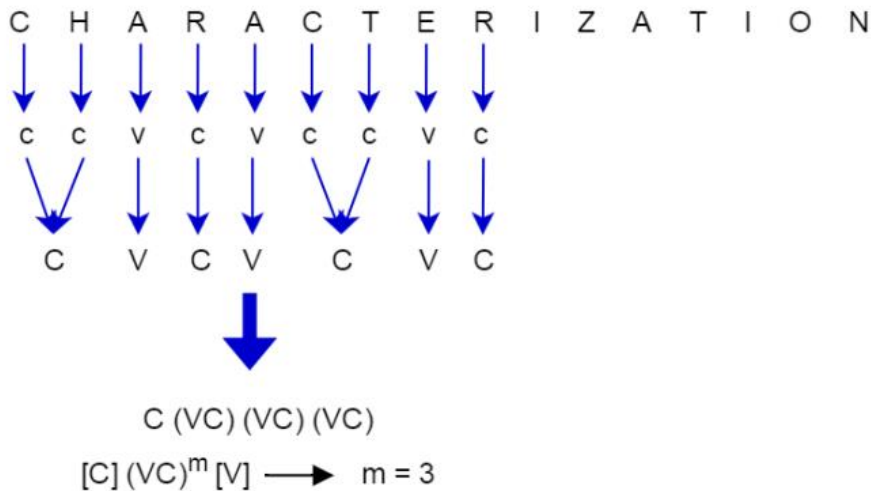
- Le suffixe ne correspondra à aucun des cas trouvés aux étapes 1, 2 et 3.
- Ensuite, il s'agit de l'étape 4.
- La racine du mot a $m > 1$ (puisque $m = 5$) et se termine par «**AL**».
- Par conséquent, à l'étape 4, «**AL**» est supprimé (remplacé par null).
- L'appel de l'étape 5 ne changera pas davantage la racine.

Enfin, la sortie sera :

MULTIDIMENSIONAL → **MULTIDIMENSION**

Porter's Stemmer algorithm - 5 Steps

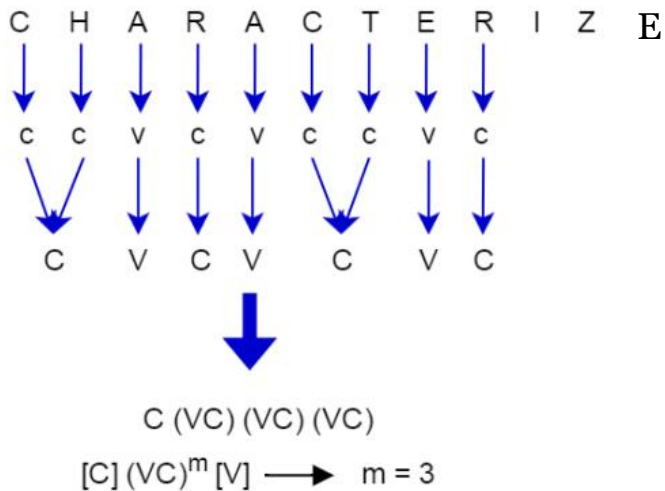
Exemple 1 : mot = **CHARACTERIZATION**



- Le suffixe ne correspondra à aucun des cas trouvés à l'étape 1.
- Il passera donc à l'étape **2**.
- La racine du mot a **m > o** (puisque m = 3) et se termine par «**IZATION**».
- Par conséquent, à l'étape 2, «**IZATION**» sera remplacé par «**IZE**».
- Ensuite, la nouvelle racine sera **CHARACTERIZE**.

Porter's Stemmer algorithm - 5 Steps

Exemple 1 : mot = **CHARACTERIZATION** -> **CHARACTERIZE**



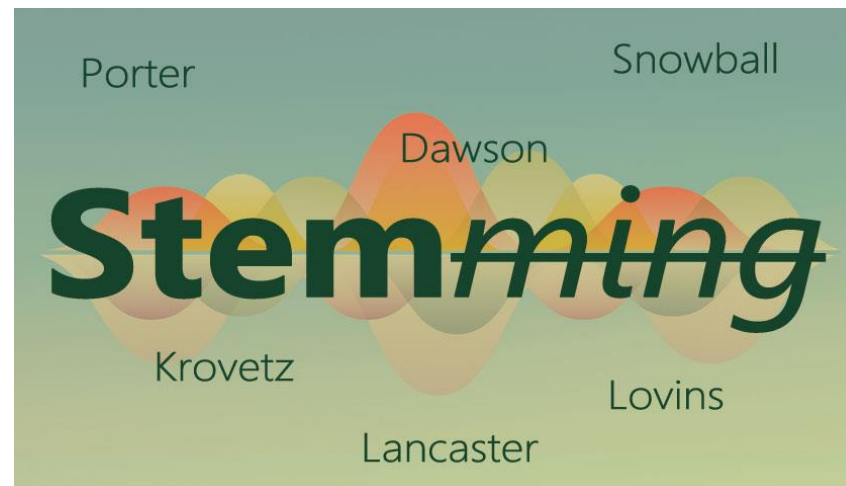
CHARACTERIZATION ->

CHARACTERIZE → **CHARACTER**

- Ensuite, la nouvelle racine sera **CHARACTERIZE**.
- L'étape 3 ne correspondra à aucun des suffixes et passera à l'étape 4.
- Maintenant **m** > 1 (puisque m = 3) et la racine se termine par «**IZE**».
- Ainsi, à l'étape 4, «IZE» sera supprimé (remplacé par **null**).
- Aucun changement ne se produira à la racine dans les autres étapes.
- Enfin, la sortie sera **CHARACTER**.

Lemmatisation et Stemming

- **Stemming** : quelques algorithmes et techniques – Stemmers
- **Liste** d'algorithmes: <https://snowballstem.org/algorithms/>
- **Demo** : <https://snowballstem.org/demo.html>
- **Demo** NLTK : <https://text-processing.com/demo/stem/>
- Porter Stemmer **Online** : https://9ol.es/porter_js_demo.html



Lemmatisation et Stemming

- **Lemmatization**: Quelques approches - Lemmatizers
- Deux principales approches :
- **Méthode basée sur des règles** – Rule Based: utilise un ensemble de règles qui indiquent comment un mot doit être modifié pour extraire son lemme. Exemple: si le mot est un verbe, et qu'il se termine par -ing, faire quelques substitutions... Cette méthode est très délicate et ne donne probablement pas les meilleurs résultats (difficile à généraliser en anglais, ex : bring).
- **Méthode basée sur le corpus** – Corpus Based: utilise un corpus taggé (ou un ensemble de données annoté) pour fournir le lemme de chaque mot. Fondamentalement, il s'agit d'une énorme liste de mots (lexicon) et de leur lemme associé pour chaque PoS (ou non, une approche naïve). Ceci, bien sûr, nécessite l'accès à un corpus annoté.

Lemmatisation et Stemming

- **Lemmatization**: Quelques approches - Lemmatizers
- Il existe de nombreuses façons d'aider une machine à gérer les lemmes :
 - ✓ Méthode Rule Based
 - ✓ Méthode Copus Based
 - ✓ **Neural lemmatizers** : utilise sequence-to-sequence (seq2seq) neural networks.
- Quelques Implémentations :
 - WordNet Lemmatizer NLTK, SpaCy Lemmatizer, Gensim Lemmatizer, Stanford CoreNLP Lemmatization, TextBlog Lemmatizer, etc.
 - Demo WordNet NLTK : <http://textanalysisonline.com/nltk-wordnet-word-lemmatizer>

Lemmatisation et Stemming

- **Lemmatization: WordNet Lemmatizer**
- **Wordnet** est une grande base de données lexicale, disponible gratuitement et publiquement pour la langue anglaise (et autres), visant à établir des relations sémantiques structurées entre les mots.
- Il offre également des capacités de lemmatisation et est l'un des lemmatiseurs les plus anciens et les plus couramment utilisés.
- **WordNet Lemmatizer** utilise la base de données WordNet pour rechercher (mapping) des lemmes de mots. Corpus based method.
- Doit aussi prendre en compte le **contexte** dans lequel on souhaite lemmatiser : **Part-Of-Speech** (POS). Car le même mot peut avoir plusieurs lemmes en fonction du sens / du contexte.

Lemmatisation et Stemming

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- [S: \(n\) helping](#), [portion](#), [serving](#) (an individual quantity of food or drink taken as part of a meal) *"the helpings were all small"; "his portion was larger than hers"; "there's enough for two servings each"*

Verb

- [S: \(v\) help](#), [assist](#), [aid](#) (give help or assistance; be of service) *"Everyone helped out during the earthquake"; "Can you help me carry this table?"; "She never helps around the house"*
- [S: \(v\) help](#), [aid](#) (improve the condition of) *"These pills will help the patient"*
- [S: \(v\) help](#), [facilitate](#) (be of use) *"This will help to prevent accidents"*
- [S: \(v\) help oneself](#), [help](#) (abstain from doing; always used with a negative) *"I can't help myself--I have to smoke"; "She could not help watching the sad spectacle"*
- [S: \(v\) serve](#), [help](#) (help to some food; help with food or drink) *"I served him three times, and after that he helped himself"*
- [S: \(v\) help](#) (contribute to the furtherance of) *"This money will help the development of literacy in developing countries"*
- [S: \(v\) avail](#), [help](#) (take or use) *"She helped herself to some of the office supplies"*
- [S: \(v\) help](#) (improve; change for the better) *"New slipcovers will help the old living room furniture"*

Lemmatisation et Stemming

- **Lemmatization:** WordNet Lemmatizer

Mapping from text-word to lemma

help (verb) help (noun), helping (noun)

text-word	to	lemma
help		help (v), help (n)
helps		help (v), helps (n)**
helping		help (v), helping (n)
helped		help (v)
helpings		helping (n)

***help (n): usually a mass noun, but part of compound **home help** which is a count noun, taking the "s" ending.*

Prétraitement et Normalisation

Description du traitement	Étape	Description du traitement	Étape
1. (Grefenstette et Tapanainen, 1994)		4. Gate 3.0 (Cunningham et al., 2002)	
1. Supprimer les étiquettes SGML	I2	1. Segmentation en mots	E2
2. Recoller les césures	I3	2. Lemmatisation	E2
3. Marquer les nombres et les abréviations	A1	3. Identification des entités nommées	A1
4. Segmenter le texte en phrases	E1	4. Découpage en phrases	E1
2. Multext multilingual segmenter tools (v 1.3.1), 1997		5. SXPIPE 1.0 (Sagot et Boullier, 2005)	
1. Séparer le texte selon les espaces	A2	1. Identification des courriels, dates, adresses, etc.	A1
2. Isoler les ponctuations	A1	2. Détection des frontières des phrases	A2
3. Fusionner les ponctuations composées	A1	3. Identification des mots inconnus	A1
4. Séparer les expressions avec ponctuations	A1	4. Identification des acronymes, noms propres	A1
5. Fusionner les abréviations composées	A1	5. Identification des mots étrangers	A1
6. Identifier les abréviations	A1	6. Découpage en mots et correction orth.	E2/I3
7. Recombiner les unités multimots	E2	7. Identification des nombres en lettres	A1
8. Identifier les dates	A1	8. Identification des mots composés	E2
9. Identifier les nombres	A1	9. Réaccentuation et recapitalisation	I3
10. Identifier les énumérations	A2		
11. Détecter les frontières des phrases	E2		

Prétraitement et Normalisation

3. (Adda et al., 1997)		6. Notre modèle (Amrani et al., 2004)	
0. Encodage des accents et autres diacritiques	I1	1. Remplace les caractères non ASCII et entités	I1
0. Prétraitement des nombres et unités	I3	2. Convertit le document en format texte linéaire	I2
0. Correction du formatage et des ponctuations	I3	3. Met un paragraphe par ligne	A2
0. Traitement des ponctuations non ambiguës	I3	4. Normalise le paragraphe - Incohérences	I3
0. Séparation en articles, paragraphes, phrases	E2	5. Normalise le paragraphe - Ambiguïtés	A1
1. Traitement des ponctuations ambiguës	A1	6. Recherche des frontières des phrases	A1, A2
2. Traitement des débuts de phrase capitalisés	A1	7. Met une phrase par ligne	E1
3. Traitement des nombres	A1	8. Normalise les phrases	E2
4. Traitement des acronymes	A1	7. (Mikheev, 2000)	
5. Traitement des capitales emphatiques	A1	1. Classer les expressions terminées par un point	A1
6. Décomposition	E2	2. Classer les mots capitalisés après un point	A1
7. Pas de distinction de casse	I3	3. Assigner les fins de phrases	E1
8. Pas de diacritiques	I1		

Tab. 1 - *Enchaînements des traitements proposés dans différentes chaînes de prétraitement des textes.*

Stop Words - Mots Vides

- Les mots vides sont les mots du texte qui n'ajoutent aucun sens à la phrase et leur suppression n'affectera pas le traitement du texte aux fins définies.
- est un mot qui est tellement commun qu'il est inutile de l'indexer ou de l'utiliser dans une recherche.
- En français, des mots vides évidents pourraient être « le », « la », « de », « du », « ce », etc.
- En anglais : all, am, an, and, any, are, etc.
- Ils sont supprimés du vocabulaire (tokens) pour réduire le bruit et réduire la dimension du texte traité.
- Liste des mots vides en différentes langues :
<https://www.ranks.nl/stopwords>

Références

Speech and Language Processing - Livre de Dan Jurafsk -
<https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf>

Cours - *François Yvon* – Une petite introduction au Traitement Automatique des Langues Naturelles,
<https://perso.limsi.fr/anne/coursM2R/intro.pdf>

Article – Marcel Cori - Des méthodes de traitement automatique aux linguistiques fondées sur les corpus
- <https://www.cairn.info/revue-langages-2008-3-page-95.htm>

Article - Pascale Sébillot - Le traitement automatique des langues face aux données textuelles volumineuses et potentiellement dégradées : qu'est-ce que cela change ?
- <https://hal.archives-ouvertes.fr/hal-01056396/document>

Cours – Morphologie - <https://elc.hypotheses.org/155>

A Comparative Study of Stemming Algorithms - A. Jivani, 2011.

Porter Stemming Algorithm - <https://vijinimallawaarachchi.com/2017/05/09/porter-stemming-algorithm/>