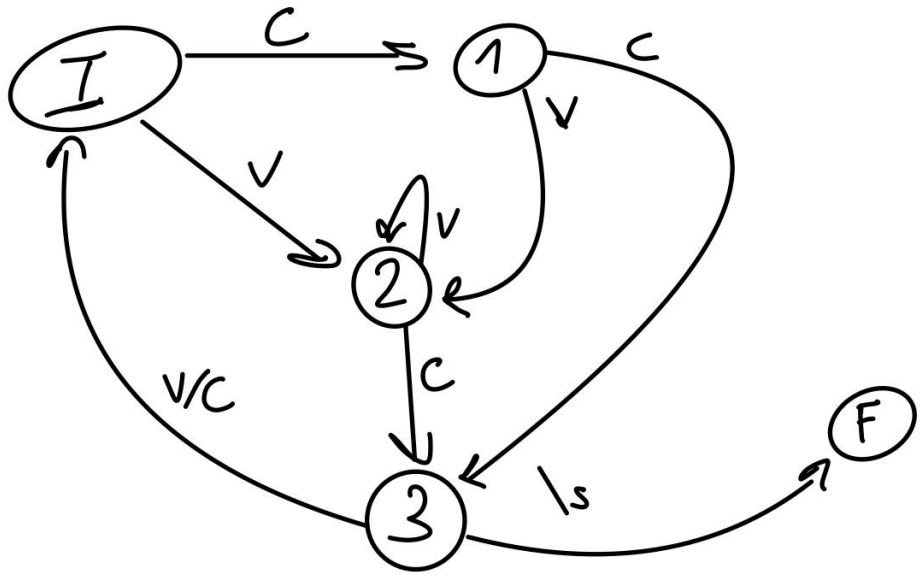
Algorithme de syllabation

Linguistique de corpus L3 S2

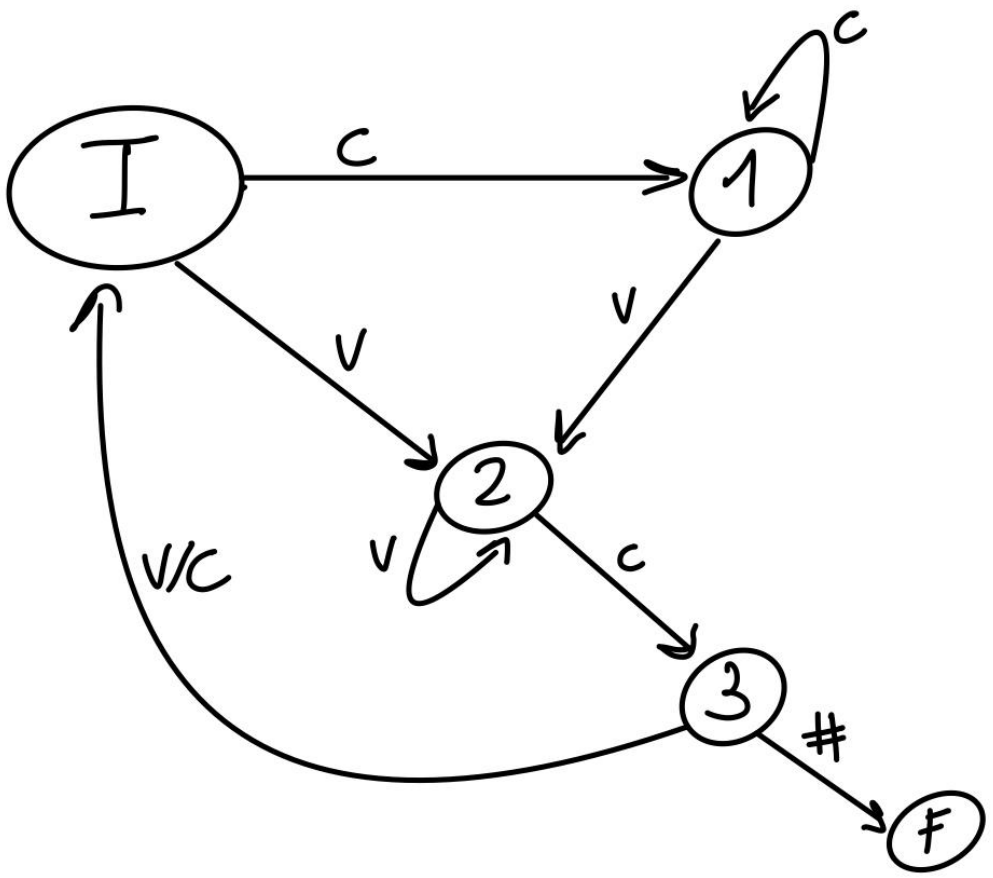
Postulat d’origine

- Mapping des mots en C ou V
- Syllabe commence par consonne ou voyelle
- Les voyelles peuvent se répéter, pas les consonnes
- Si une deuxième consonne est detectée -> fin de syllabe



Algorithme actuel de l'automate

- Similaire au premier, mais permet la répétition des consonnes
- A l'état 3, la syllable, voire le mot se clot.



Pré-traitement

- Substition des caractères par leur catégorie :
- Consonne -> C
- Voyelles -> V
- Fin de mot -> #
- Combination des letters de même catégorie adjacentes
- Ex : champagne -> ch-a-m-p-a-gn-e
- Exception pour “n” et “m” qui ont un traitement spécial.

Post-traitement

- Repêchage des syllabes orphelines
- Ajout des lettres orphelines en fin de phrase à la syllabe

Résultats positifs dans l’état

- Champagne -> cham pa gne
- Philosophie -> Phi lo so phie
- Eclectique -> e cle cti que
- Cigarette -> ci ga re tte

Le reste…

- Honneur -> hon neur
- Ensemble -> en se mble
- Reconciliation -> re co nci lia tion

Problèmes rencontrés

- Graphie limitée du français -> beaucoup de bigraphes et trigraphes
- Différences entre la phonétique et la graphie

-> Difficulté à établir des règles généralistes à la fois inclusives

Versions du script

- syllabation.py (fait le job)
- syllabation2.py (a voulu gérer les diagraphes mais ne gère rien du tout)
- syllabation3.py (a tout détruit pour mieux reconstruire, algorithme initial amélioré)
- syllabation3.5.py (l’enfer du debugage, ne rien toucher ça marche !)

La suite

- Formalisation en classe
- Amélioration des règles de syllabation pour les nasales avec “m” et “n”
- Support des balises XML
- Calcul des métriques pour évaluer la qualité du modèle

Merci pour votre attention !