Nicolas AUDOUX
Yana RAGOZINA
Thomas PAUL
SI5 IA-ID

Ingénierie des Connaissances

# Extraction d'Informations et Graphe de Connaissances

## 1/ Introduction

L'objectif de ce TD est double : d'une part, comprendre en profondeur le fonctionnement des algorithmes et modèles d'extraction d'informations (IE) et, d'autre part, apprendre à les utiliser pour créer et enrichir un graphe de connaissances.

## 2/ Procédure

- ### Découpage
  Nous avons décidé de découper le texte phrase par phrase. Pour cela, nous avons enregistré le texte dans un fichier txt en prenant soin de faire un retour à la ligne entre chaque phrase. On a ensuite effectué un split sur les retours à la ligne pour avoir notre découpage final. L'avantage de cette approche est qu'on est sûr qu'aucune phrase ne sera coupée à cause de sa longueur (le nombre maximum de tokens étant de 4096). Cependant, il est possible qu'on perde de l'information si une phrase commence par "il" en reprenant le sujet de la phrase précédente.

- ### Extraction d'entités et de relations
  Pour chaque phrase, nous avons effectué l'extraction des relations sur les tokens récupérés grâce à REBEL. Nous avons ensuite sauvegardé les triplets dans une liste de dictionnaire de la forme {head: …, type: … , tail: …}.

- ### Création du graphe de connaissances
  Pour créer le graphe, nous avons parcouru notre liste de triplets et avons créé un nœud pour les valeurs head et tail (qui correspondent respectivement au sujet et à la valeur du prédicat) ainsi qu'un lien dirigé allant du sujet à la valeur du prédicat. Nous avons mis comme label pour ce lien la valeur du type qui correspond au prédicat.
  La liste des nœuds étant un set, il n'est pas possible d'avoir deux nœuds ayant la même valeur. Ainsi, on est assuré que si deux triplets partagent un nœud en commun, ce dernier ne sera pas dupliqué.

- ### Analyse et interprétation
  Les triplets extraits permettent d'avoir un graphe qui résume assez bien le texte. On retrouve en effet la majorité des informations et on comprend facilement que le texte est

lié aux chercheurs ayant fait avancer le domaine de l'IA. Cependant, on remarque que certaines informations sont manquantes. C'est notamment le cas pour les informations concernant Geoffrey Everest Hinton qui sont incomplètes et séparées car ayant plusieurs sujets lui faisant référence ("Geoffrey Everest Hinton", "Geoffrey Hinton" et "Hinton"). Une solution pour éviter ce problème serait de prendre plus qu'une phrase pour extraire les triplets car cela permettrait de repérer plus facilement si deux ressources sont identiques. Cela impliquerait de faire un travail de vérification pour être sûr de ne pas ajouter deux fois le même triplet (notamment si on utilise une fenêtre glissante pour extraire nos triplets).

## 3/ REBEL : Relation Extraction By End-to-end Language generation

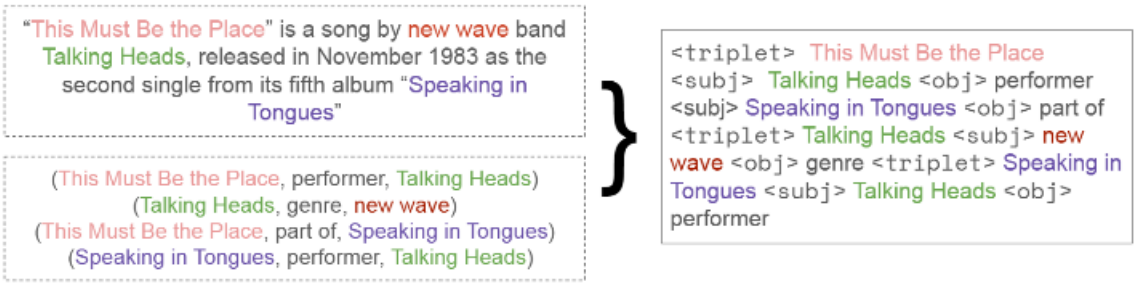
- Modèle REBEL

Le modèle REBEL est un modèle Seq2Seq basé sur BART, un type particulier de Transformers (modèle du type encodeur-decodeur). BART est notamment utilisé pour les tâches de type Seq2Seq ayant pour but de transformer une certaine séquence d'entrée en une séquence de sortie recherchée. BART permet également d'effectuer des tâches plus fines telles que l'extraction de relations dans un texte, comprenant plus de 200 types de relations.

REBEL est défini comme une approche autorégressive qui traduit l'extraction de relations en une tâche de type Seq2seq. En effet, entraîné sur une base de données supervisée (Distant Supervision), REBEL explore les inférences en langage naturel pour pouvoir décomposer une séquence de texte en ensemble de triplets. De même, le modèle est capable de réaliser des tâches de classification. Ainsi, l'extraction des relations peut être vue comme une tâche générative: il suffit d'extraire les relations et puis de faire une classification afin de produire des triplets à partir d'un texte en entrée.

Le processus de création des triplets peut être représenté ainsi : chaque phrase en entrée, contenant des entités, est traduite en ensemble de triplets. Cette traduction est basée sur les relations implicites contenues dans le texte passé en entrée. Les triplets résultants font référence aux entités dans le texte et aux relations entre eux. Ils peuvent être également vus comme un ensemble de tokens que le modèle cherche à décoder. Les tokens utilisés sont définis comme suivant : <*triplet*>, marquant le début d'un nouveau triplet avec une nouvelle entité principale (sujet), <*subj*>, marquant la fin de l'entité principale et le début de l'entité secondaire (objet), et <*obj*>, marquant la fin de l'entité secondaire et le début de la relation entre l'entité principale et l'entité secondaire (prédicat).
De cette manière, le modèle cherche à minimiser le nombre de tokens à décoder, produisant ainsi un ensemble des relations dans le texte sous forme de triplets. Elle prend donc en entrée un texte brut et donne en sortie un ensemble de triplets. Les

triplets sont linéarisés dans l'ordre d'apparition des entités qu'ils contiennent afin de maintenir l'ordre de cohérence des relations.




Exemple d'extraction des triplets à partir d'un texte

- REBEL dataset

Le modèle REBEL est pré-entraîné sur un abstract de Wikipedia. Ce dataset est obtenu grâce à l'extraction de chaque page Wikipedia (partie avant le sommaire de la page) à l'aide de Wikiextractor. Ensuite, les entités présentes dans le contenu de chaque page sous forme d'hyperlien sont liées ensemble à Wikidata à l'aide de Wikimapper. Wikidata continent des relations entre ces entités. Les relations sont ensuite extraites pour la construction du dataset.