

Représentation des connaissances - TP5

Résolution d'anaphores

31 mars 2010

1 Exercices sur papier

Marquez le référent de chaque pronom anaphorique dans les phrases suivantes :

1. A l'assermentation, Lucien Bouchard a dit qu'il comptait sur François Legault pour donner aux entreprises les diplômes qu'il leur faut.
2. Le nouveau ministre va-t-il mettre le réseau scolaire au service des entrepreneurs ?
3. François Legault ne va pas tarder à être confronté au fait que les universités ne se restructurent pas aussi facilement que les entreprises. Québec les finance en fonction du nombre d'étudiants qu'elles attirent.
4. Dans un site comme celui de eBay, il est impossible de savoir si la description du produit sur lequel vous misez correspond à la réalité. Fred Breitner y a déniché un vieil appareil, mais qui fonctionnait toujours.
5. La canadienne Amanda Vincent dirige un projet de conservation auquel se sont même associés des travailleurs sociaux !
6. La tomate californienne contient un gène de résistance à la kanamycine, un antibiotique couramment utilisé pour traiter la tuberculose. Celui-ci pourrait-il se transmettre aux mangeurs de tomates ?
7. Jacques était allé voir son oncle ; celui-ci était malade.
8. Jacques était allé voir son oncle ; il était malade.

2 Implémentation d'un résolveur simple

La suite du TP vise la construction d'un système de résolution d'anaphores pour l'anglais. En particulier il s'agira d'implémenter une version (très) simplifiée de l'algorithme de Lappin et Leass (1994) vu en cours. Votre tâche est considérablement facilitée, puisque la détection et la caractérisation des mentions référentielles ont déjà été opérées. Une part importante du code vous est fournie, y compris des API permettant de lire et charger les données du corpus annoté, ainsi qu'une méthode qui évalue les résultats.

Les classes et méthode importantes pour ce TP sont les suivantes :

Fichier	Classe	Méthode	Fonction
resolver.py	HeuristicResolver	link(self, mentions)	Renvoie une liste de pairs (pronom, référent) - à compléter. L'argument mentions est une liste d'instances de la classe Mention.
lexicon.py	-	-	Contient des expressions régulières pour reconnaître différents types de pronoms (masc/fém/neutre, sg/pl) Comporte également des listes de prénoms masculins et féminins.
mention.py	Mention	getType() is_pronoun() is_pn() is_nominal() get_n_preceding_mentions() get_mdistant() get_number() get_gender() is_compatible_with	La classe qui représente les mentions référentielles (les noms et pronoms) dans un document. Renvoie le type d'une mention (nom, pronom etc). Renvoie True si la mention est un pronom. Renvoie True si la mention est un nom propre. Renvoie True si la mention est un nom ou un nominal. Renvoie la liste des n mentions précédentes. Renvoie la distance entre deux mentions (en termes de position dans le texte). Renvoie le nombre d'une mention référentielle (nom ou pronom) - à compléter. Renvoie le genre d'une mention référentielle (nom ou pronom) - à compléter. Renvoie True si deux mentions sont compatibles (ex. en genre et en nombre) - à compléter.

Pour exécuter le script sur le corpus de test : `python resolver.py data/test`. Il affiche la liste des documents traités suivi des mesures de précision, rappel et F-score pour l'évaluation de l'algorithme.

1. Dans la classe Mention (mention.py), complétez les méthodes `get_number()` et `get_gender()`. Vous vous servirez des expressions régulières contenues dans le fichier `lexicon.py`.
2. Complétez la méthode `is_compatible_with()` qui vérifie si deux mentions s'accordent en genre et en nombre (et éventuellement sur d'autre critère si vous en trouvez).
3. Ecrivez une méthode `score_candidates()` qui prend en paramètre une liste de mentions candidats et une mention pronominale. Elle calculera un score pour chacun des référents candidats. Elle doit renvoyer un dictionnaire contenant chaque candidat avec son score. Implémentez les heuristiques suivantes pour calculer le score.
 - (a) La mention candidate doit s'accorder en genre avec le pronom.
 - (b) La mention candidate doit s'accorder en nombre avec le pronom.
 - (c) Plus le candidat est loin du pronom, moins il est probable comme référent.
4. Ecrivez une méthode `get_max()` qui renvoie la mention référentielle maximale avec son score (un tuple).
5. Dans la classe `LLResolver` (resolver.py), complétez la méthode `link()`. Elle devra associer chaque pronom rencontré (dans la liste des mentions référentielles du document) à un référent (un nom) selon le score qui lui a été attribué (celui avec le score maximum).