DM1 Sémantique Computationnelle

Cong Jinyu

16 octobre 2024

1 Introduction

Dans le cadre de cette étude, nous analysons les relations sémantiques entre des paires de mots concrets à l'aide de deux approches distinctes : la distance lexicale calculée dans WordNet, et la similarité sémantique fournie par SimLex-999. SimLex-999 est un ensemble de données conçu pour évaluer la similarité sémantique entre les mots, où chaque paire est annotée avec un score reflétant leur similarité perçue par des humains. En complément, nous utiliserons les mesures de similarité lexicale disponibles dans la bibliothèque nltk de WordNet, qui propose trois métriques populaires : Path Similarity, Leacock-Chodorow Similarity (LCH), et Wu-Palmer Similarity (WUP).

Le but de cette étude est de comparer les scores de similarité fournis par SimLex-999 avec les distances lexicales calculées à partir de WordNet pour deux groupes de paires de mots : un groupe avec des distances SimLex-999 supérieures à 9 (faible similarité) et un autre avec des distances inférieures à 2 (forte similarité). Ces paires seront analysées en termes de distance lexicale dans WordNet afin d'examiner dans quelle mesure ces deux approches capturent la similarité entre les mots, en cherchant à identifier des divergences ou des convergences entre les jugements humains et les mesures automatiques de similarité sémantique.

2 Les troix façons de mesures de similarité

2.1 Path similarity (Similarité par chemin)

Définition : La Path Similarity mesure la similarité en fonction de la longueur du chemin le plus court entre deux synsets dans la hiérarchie de WordNet. La similarité varie entre 0 et 1, où 1 signifie que les deux mots sont très similaires (par exemple des synonymes).

Formule :

$$PathSimilarity = \frac{1}{longueur\_chemin\_le\_plus\_court(synset1, synset2)}$$

Plus le chemin est court, plus la similarité est grande.

2.2 Leacock-Chodorow Similarity (LCH)

Définition : La LCH Similarity prend en compte la longueur du chemin entre deux concepts, mais ajuste cette longueur en fonction de la profondeur maximale de la hiérarchie dans WordNet. Cette mesure exprime la similarité en fonction du logarithme négatif de la longueur du chemin.

Formule :

$$LCHSimilarity = -\log(\frac{longueur\_chemin\_le\_plus\_court(synset1, synset2)}{2 \times profondeur\_maximale})$$

Cela permet de prendre en compte non seulement la distance entre les concepts, mais aussi la profondeur de leur position dans la hiérarchie de WordNet.

2.3 Wu-Palmer Similarity (WUP)

Définition : La Wu-Palmer Similarity évalue la similarité en fonction de la profondeur du synset commun le plus proche (ou ancêtre commun le plus bas, LCA) dans la hiérarchie de WordNet. Elle compare la distance de chaque synset à cet ancêtre commun avec la profondeur de l'ancêtre dans la hiérarchie.

Formule :

$$WUPSimilarity = \frac{2 \times profondeur(LCA)}{profondeur(synset1) + profondeur(synset2)}$$

Plus la profondeur de l'ancêtre commun est élevée, plus les deux concepts sont considérés comme similaires.

2.4 Résumé des troix façons de mesures

Path Similarity mesure la similarité en fonction du chemin le plus court entre deux concepts dans WordNet. Plus le chemin est court, plus la similarité est élevée.

Leacock-Chodorow Similarity (LCH) ajuste la similarité en fonction du chemin le plus court et de la profondeur maximale de WordNet, normalisant ainsi la distance.

Wu-Palmer Similarity (WUP) prend en compte la position de l'ancêtre commun des deux concepts dans la hiérarchie de WordNet pour calculer la similarité. Plus cet ancêtre est profond, plus les concepts sont jugés proches.

3 Explication du code python

D'abord, importez le module nltk et le sous-module wordnet de nltk conformément aux exigences, puis importez le module pandas afin de créer un data frame pour les informations extraites.

Ensuite, créez un dictionnaire pour stocker les informations de chaque paire de mots. Les clés du dictionnaire seront "word1", "word2", "simlex", "path_sim", "lch_sim", et "wup_sim", correspondant respectivement au premier mot de la paire, au deuxième mot, à la similarité extraite de SimLex-999, et aux similarités calculées pour la paire de mots selon trois méthodes différentes.

Après avoir lu le fichier SimLex ligne par ligne, chaque ligne est séparée par des tabulations. Le premier élément de chaque ligne correspond au mot1, le deuxième au mot2, le troisième à la catégorie grammaticale du mot, et le quatrième à la similarité SimLex. J'ai pris en compte la catégorie grammaticale des mots, car si la paire de mots est composée d'adjectifs, les trois similarités calculées via nltk sont toutes identiques. La raison de ce phénomène reste pour le moment inconnue, mais il est possible que cela soit dû au fait que les paires d'adjectifs ont la même profondeur dans la hiérarchie des concepts de WordNet. Par conséquent, j'ai choisi exclusivement des paires de mots ayant une catégorie grammaticale de nom.

Ensuite, chaque mot est converti en synset (ensemble synonyme) afin de calculer les similarités entre les deux synsets à l'aide de nltk, et les trois résultats de similarité calculés sont placés dans les listes correspondantes du dictionnaire. Selon cette démarche, j'ai identifié cinq paires de mots dont la similarité SimLex est supérieure à 9, et cinq paires de mots dont la similarité est inférieure à 2. Ainsi, la création du data frame est achevée. Le tableau créé par pandas est comme tableau 1.

4 Analyse des données

Dans le tableau, on peut observer que, pour les cinq paires de mots avec un score SimLex supérieur à 9, à l'exception de taxi et cab, les autres paires de mots présentent des similarités élevées après calcul avec nltk. Cependant, il existe des exceptions, comme par exemple le path similarity entre area et region, qui n'est que de 0,25, soit 1/4. Cela signifie que la distance absolue entre area et region dans WordNet est de 4.

Pour les cinq autres paires de mots dont le score SimLex est inférieur à 2, leurs scores de path similarity et de lch similarity sont assez faibles, ce qui est prévisible, car elles représentent des paires de mots qui sont souvent des antonymes évidents, comme night et day, ou bottom et top. Cependant,

dog et cat, ou mouse et cat, ne sont pas strictement des antonymes, donc je trouve normal que leur wup similarity dépasse 0,8.

Ensuite, j'ai remarqué que les tendances de la path similarity et de la lch similarity sont relativement cohérentes : lorsqu'une paire a une path similarity faible, sa lch similarity est également faible. Je pense que cela s'explique par le fait que ces deux algorithmes de calcul de similarité utilisent la distance absolue entre deux mots dans WordNet. En revanche, la wup similarity se base sur la profondeur des deux mots dans WordNet ainsi que sur la profondeur de leur ancêtre commun, ce qui donne des résultats souvent différents des deux premières mesures. Par exemple, si un mot a une profondeur de 20 et l'autre de 21, et que leur ancêtre commun a une profondeur de 18, alors leur wup similarity sera calculée comme suit :

$$\frac{2 \times 18}{20 + 21} = 0.88$$

Cependant, la distance absolue entre ces deux mots dans le réseau sémantique pourrait être très longue, ce qui expliquerait pourquoi leur path similarity et lch similarity sont très faibles alors que leur wup similarity est élevée.

Je pense que la similarité faible entre taxi et cab pour toutes les mesures s'explique par le fait que cab est une expression familière désignant taxi, mais aussi parce que cab est un mot polysémique avec d'autres significations. Il est possible que mon code n'ait pas converti cab en un synset correspondant à son sens proche de taxi.

5 Conclusion

Dans WordNet, deux mots qui sont des antonymes peuvent tout de même avoir une similarité élevée. Cela s'explique par plusieurs facteurs, car WordNet ne se limite pas à déterminer si deux mots sont des antonymes. Si deux mots appartiennent au même champ sémantique, même s'ils sont des antonymes, leur similarité peut être élevée. En revanche, lorsque deux mots n'ont pratiquement aucun lien entre eux, comme maman et étoile, leur similarité sera logiquement très faible, car ces mots ne sont presque jamais utilisés dans le même contexte.

word1	word2	simlex	path_sim	lch_sim	wup_sim
student	pupil	9.35	1.000000	3.637586	1.000000
teacher	instructor	9.25	1.000000	3.637586	1.000000
area	region	9.47	0.250000	2.251292	0.727273
cow	cattle	9.52	0.500000	2.944439	0.971429
taxi	cab	9.2	0.076923	1.072637	0.454545
night	day	1.88	0.166667	1.845827	0.545455
bottom	top	0.7	0.142857	1.691676	0.625000
dog	cat	1.75	0.200000	2.028148	0.857143
mouse	cat	1.12	0.166667	1.845827	0.814815
floor	ceiling	1.73	0.166667	1.845827	0.705882

TABLE 1 – DataFrame des 10 paires de mots