# TP1 - Simlex vs. distance lexicale

**Morgane Bona-Pellissier**, Master 1 pluriTAL
morgane@bona-pellissier.net

L'ensemble de ce projet est disponible sur le repository suivant :
`https://github.com/crispyfunicular/semantique-distributionnelle/tp1`

## Consignes

https://www.linguist.univ-paris-diderot.fr/~amsili/Ens/LZSET06/

## Structure du code Python

- **input** : Le code prend en entrée un fichier .cvs contenant la liste des paires de mots fournie (données SimLex-999)
- **output** : Le code fournit deux tableaux en sortie :
  - l'un dans le fichier table.csv avec la commande `f.write` (`sep=";"`), et
  - l'autre directement dans le terminal avec la commande `print` (`sep="\t"`).
- Les paires de termes ont été conceptualisées commes des objets Python, dotés chacun de :
  - un premier mot (`word1`) ;
  - un sencond mot (`word2`) ;
  - une étique morph-syntaxique ou partie de discours ("part-of-speech" ou "POS") : nom (N), verbe (V), adjectif (A) ou adverbe (`pos`) ;
  - un score de similarité lexicale ou degré de synonymie perçu par des locuteurs natifs (`simlex`) ;
  - les objets Synset de WordNet (`wn.synset`) correspondant au premier sens (01) de chaque mot (`word1_synset` et `word2_synset`) ;
  - le score de similarité de chemin entre les deux sens (`path_score`) ;
  - le score de Leacock-Chodorow (LCH), qui pondère la distance par la profondeur maximale de la taxonomie (`lch_score`) ;
  - le score de Wu-Palmer, basé sur la profondeur des deux sens et celle de leur ancêtre commun le plus spécifique (`wup_score`) ;

```
    class WordPair:
    def __init__(self, word1, word2, pos, simlex):
        self.word1 = word1
        self.word2 = word2
        self.pos = pos
        self.simlex = simlex
        self.word1_synset = wn.synset(f"{word1}.{pos.lower()}.01")
        self.word2_synset = wn.synset(f"{word2}.{pos.lower()}.01")

        self.path_score =
self.word1_synset.path_similarity(self.word2_synset)
```

```
        self.lch_score =
self.word1_synset.lch_similarity(self.word2_synset)
        self.wup_score =
self.word1_synset.wup_similarity(self.word2_synset)
```

- Les dix paires de termes sur lesquelles effectuer nos mesures ont été sélectionnées aléatoirement à l'aide de la bibliothèque `random`, celle-ci ayant été « figée » dans le `main()` par la fonction `seed` afin de faciliter notre étude.

```python
def get_random_words(input_file) -> list[WordPair]:
    # Liste "greater than 9"
    gt_9 = []

    # Liste "less than 2"
    lt_2 = []

    i = 0
    with open(input_file, "r", encoding="utf-8") as f:
        for line in f:
            i += 1
            line_lst = line.split(";")

            # On saute la première ligne correspondant au titre des
colonnes
            if i == 1:
                continue

            if len(line_lst) < 3:
                raise Exception(f"Liste trop courte : {len(line_lst)}
{line}")
            word1 = line_lst[0]
            word2 = line_lst[1]
            pos = line_lst[2]
            simlex = float(line_lst[3])

            # Evite les paires dont le POS n'est pas le même (ex :
new.a.01 et ancient.s.01)
            # car le score LCH nécessite que les deux mots aient le
même POS
            try:
                pair = WordPair(word1, word2, pos, simlex)
            except Exception:
                continue
            if pair.simlex > 9:
                gt_9.append(pair)
            if pair.simlex < 2:
                lt_2.append(pair)
    return sample(lt_2, 5) + sample(gt_9, 5)
```

## Discussion des résultats

```
mot 1       mot 2       SimLex  path    LCH     WUP
get         put         1.98    0.33    2.16    0.33
forget      know        0.92    0.33    2.16    0.50
multiply    divide      1.75    0.33    2.16    0.75
modest      flexible    0.98    0.33    -0.41   0.50
container   mouse       0.3     0.20    2.03    0.75
inform      notify      9.25    0.50    2.56    0.89
vanish      disappear   9.8     1.00    3.26    1.00
quick       rapid       9.7     0.33    -0.41   0.50
cow         cattle      9.52    0.50    2.94    0.97
student     pupil       9.35    1.00    3.64    1.00
```

### Synonymes presque parfaits

```
mot 1       mot 2       SimLex  path    LCH     WUP
vanish      disappear   9.8     1.00    3.26    1.00
student     pupil       9.35    1.00    3.64    1.00
```

Les deux paires de mots ci-dessus obtiennent des scores Wu-Palmer (WUP) maximaux, ce qui signifie que chacun des membres d'une paire se trouve au même endroit dans la hiérarchie que l'autre. Nous pouvons parler dans ce cas de « synonymes parfaits ».
Les scores obtenus sont élevés en partie parce que nous avons fait le choix de retenir systématiquement la paire de synsets maximisant le score de path. Le fait de capturer la proximité sémantique la plus forte nous permet d'éviter les où deux mots partageant pourtant une similarité lexicale forte obtenaient des scores path et WUP très faibles car le sens qui était alors retenu arbitrairement ("01" d'office) étaient en fait très éloignés. Ainsi, avant nos ajustements, la paire "creator"-"maker" obtenait un score de path de 0.07 et un score WUP de 0.13 car le premier sens de "creator" était spirituel, divin, et l'ancêtre commun qu'il partageait avec le premier sens de "maker".
On peut estimer que notre parti pris ajoute un biais légitime, dans la mesure où, dans une liste de mots isolés comme le SimLex, l'esprit humain sélectionne naturellement les sens les plus compatibles entre eux pour évaluer leur ressemblance.

### Calcul des scores LCH et WUP pour les adjectifs

Bien que nous ayons à deux reprises pris soin d'éviter les paires dont au moins l'un des éléments aurait hérité d'un POS « stallite » (voir ci-après), nos résultats sont à prime abord déconcertants pour les paires d'adjectifs.

```
for synset1 in synsets1:
    for synset2 in synsets2:
        if not synset1.pos() == synset2.pos():
            continue
```

```
# Evite les paires dont le POS n'est pas le même (ex : new.a.01 et
ancient.s.01)
# car le score LCH nécessite que les deux mots aient le même POS
try:
    pair = WordPair(word1, word2, pos, simlex)
except Exception as e:
    print(e)
    continue
```

En effet, nous obtenons des scores LCH (-0.41) négatifs et des scores WUP écrasés à 0.50 pour les paires d'adjectifs suivantes :

```
mot 1       mot 2       SimLex  path    LCH     WUP
modest      flexible    0.98    0.33    -0.41   0.50
quick       rapid       9.7     0.33    -0.41   0.50
```

Cela peut s'expliquer par la manière dont WordNet organise les noms et les verbes, d'une part, et les adjectifs, d'autre part.

> « Les **noms** et **verbes** sont organisés en hiérarchies. Des relations d'hyperonymie (« est-un ») et d'hyponymie relient les « ancêtres » des noms et des verbes avec leurs « spécialisations ». Au niveau racine, ces hiérarchies sont organisées en types de base. Le réseau des noms est bien plus profond que celui des autres parties du discours.
> [...]
> L'organisation des **adjectifs** est différente. Un sens « tête » joue un rôle d'attracteur ; des adjectifs « satellites » lui sont reliés par des relations de synonymie. On a donc une partition de l'ensemble des adjectifs en petits groupes. » (Chaumartin, 2011, p. 3-4)

Or, la formule logarithmique de Leacock-Chodorow (LCH) s'appuie précisément sur une « profondeur maximale ». Comme Chaumartin (2011) l'explique, le réseau des adjectifs n'est pas profond. NLTK utilise une valeur de profondeur très petite pour les adjectifs, ce qui fait mathématiquement basculer le score LCH dans le négatif.

En conséquence, si le score WUP reste techniquement calculable pour les adjectifs, sa pertinence reste limitée du fait qu'il soit écrasé à 0.50 en raison de l'absence de hiérarchie ascendante profonde. Le score LCH doit cependant être systématiquement écarté pour cette catégorie morpho-syntaxique car il ne peut s'appuyer sur une taxonomie en arbre en renvoie des valeurs négatives non exploitables du fait de sa formule logarithmique.

### Similarité taxonomique vs. similarité sémantique

```
mot 1       mot 2       SimLex  path    LCH     WUP
multiply    divide      1.75    0.33    2.16    0.75
```

Nous constatons avec étonnement que la paire **"multiply"-"divide"** affiche une similarité lexicale (SimLex) basse (1.75) mais qu'elle obtient néanmoins un score WUP élevé (0.75). Cette différence peut s'expliquer par le fait que WordNet mesure la similarité taxonomique (les deux mots se réfèrent tous deux à des opérations) plutôt que la similarité sémantique (ils sont opposés). Dans la structure en arbre propre à WordNet, cela se traduit par le fait que deux antonymes partagent un hyperonyme direct, leur « père ». Il s'agit donc de deux co-hyponymes d'« opération ». La distance qui les sépare est par conséquent minimale, ce qui explique qu'ils partagent une similarité *taxonomique* forte, telle que reflétée par le score WUP, là où le jugement humain du SimLex retient une similarité *sémantique* faible en raison de la relation d'opposition entre les deux antonymes.

Une « anomalie » similaire ressort de l'examen des scores obtenus par la paire **"container"-"mouse"**. En effet, bien qu'elle présente un score SimLex faible (0.3), elle obtient néanmoins un score WUP étonnament élevé (0.75). Ce phénomène s'explique par la structure de WordNet où, comme le souligne Chaumartin (2011), « [L]e réseau des noms est bien plus profond que celui des autres parties du discours ». Puisque pour le score WUP, l'arbre est très profond, le point de jonction (l'ancêtre commun le plus spécifique) entre les deux concepts peut se situer à un niveau qui semble « proche » de la racine de façon relative même s'il possède une profondeur absolue importante et peut sembler sémantiquement bien trop générique pour un humain.

```
mot 1       mot 2       SimLex  path    LCH     WUP
container   mouse       0.3     0.20    2.03    0.75
```

## Conclusion

Ce TP met en évidence les divergences qui peuvent exister entre la perception humaine (SimLex) de la similarité lexicale entre deux mots et les scores obtenus par calcul mathématique de la distance taxonomique entre ces mêmes mots (path, LCH et WUP). En effet, dans le premier cas, la similarité est évaluée sur la base des traits sémantiques partagés, tandis que, dans le second cas, la similarité est évaluée sur la base de la similarité est évaluée sur la base de la proximité au sein d'une ontologie structurée.

En outre, nos résultats montrent que l'efficacité des scores est étroitement liée à la catégorie morphosyntaxique des termes et à la profondeur de l'arborescence associée. En effet, pour les noms et les verbes, et sous réserve de rechercher en amont l'association de sens (synset) la plus pertinente pour deux mots donnés (recherche de proximité maximale), il est possible d'obtenir des synonymies quasi parfaites.

Cependant, deux limites majeures ont été identifiées :

- La **structure taxonomique vs sémantique** : WordNet favorise la parenté structurelle (partage d'un hyperonyme). Cela crée des résultats surprenants (que l'on pourrait considérer comme de « faux positifs ») pour les antonymes (p. ex. : "multiply"-"divide") ainsi que pour les termes ontologiquement proches mais sémantiquement éloignés (p. ex. : "container"-"mouse").
- Le **cloisonnement des catégories** : WordNet repose sur des formules nécessitant une structure hiérarchique particulièrement profonde, comme celle des noms ; celle des adjectifs étant « plate » de par son organisation particulière en tête-satellites, les calculs du score de path sont peu informatifs et la mesure LCH inexploitable.

En définitive, si WordNet reste un outil intéressant pour l’analyse lexicale, ce TP confirme qu’une mesure purement hiérarchique ne peut totalement se substituer au jugement humain. Ce dernier intègre des dimensions sémantiques telles que le contexte, l’usage et l’opposition que la simple distance dans un arbre ne peut modéliser parfaitement.

## Bibliographie

Chaumartin, F. (2011). « WordNet et son écosystème: un ensemble de ressources linguistiques de large couverture ». Colloque BD lexicales, 2007, Montréal, Canada. hal-00611240