Analyse sémantique automatique du langage naturel Marie Candito
M2 Linguistique Informatique – Univ. Paris Diderot

# Désambiguïsation de verbes

Projet réalisable en binôme. A rendre pour le 5 janvier : un petit rapport et le code. Soutenance le 11 ou 12 janvier.

## 1 Objectif et données

On considère la tâche de désambiguisation lexicale, en utilisant des données annotées pour 3 verbes français (annotation par Lucie Barque, Benoît Crabbé, Adrien Roux et Vincent Segonne, and le cadre du M2 de ces 2 derniers).
On considère une méthode purement supervisée: classifieur de type réseau de neurones versus une technique par propagation de label.

En effet, le volume de données annotées est petit (172, 223 et 199 occurrences annotées pour respectivement *abattre, aborder, affecter*). Les 3 verbes seront traités indépendamment les uns des autres, et les phrases des occurrences sont indépendantes les unes des autres.

Les annotations sont fournies au sein de phrases analysées syntaxiquement: il s'agit de graphes de dépendances "profondes" (obtenus en utilisant l'analyseur de M. Coavoux produisant des arbres de dépendances syntaxiques, et le module de conversion en syntaxe profonde de Ribeyre et al.[1]).

On fournit également des vecteurs de mots vecs100-linear-frwiki obtenus sur un dump wikipedia, de 650 millions de mots[2], téléchargeable à
http://www.linguist.univ-paris-diderot.fr/~mcandito/vecs100-linear-frwiki.bz2

**Dans la version de base, vous ferez au choix un classifieur de type réseau de neurones ou bien par propagation de label, avec contexte linéaire, dont vous comparerez les performances à l'heuristique de sens le plus fréquent dans les données d'entraînement (ou graines). Vous pourrez constater que le contexte linéaire ne fonctionne pas très bien.**

Une amélioration est d'utiliser le contexte syntaxique (surface / profond).
Une version plus poussée est d'implémenter les 2 algos, et de comparer les performances entre

- apprentissage supervisé sur n exemples
- propagation de label en partant de n graines (n=30, n=50)

**Découpage:** Le découpage des données en train (ou graines) et test se fera en fixant le nb d'exemples de train, et en les sélectionnant de manière répartie sur tous les exemples (facile avec numpy.linspace).

---

[1] Corentin Ribeyre, Marie Candito, et Djamé Seddah. 2014. Semi-Automatic Deep Syn- tactic Annotations of the French Treebank. Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories. Tübingen Universität, Tübingen, Germany

[2] Obtenus par M. Coavoux, via word2vec, fréquence minimale 100, skip-gram, sur le dump frwiki-20140804-corpus.xml.bz2 téléchargé http://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/.

**NB: le peu de données annotées fait que les évaluations seront sur un ens. de l'ordre de 150 exemples, les différences de précision seront la plupart du temps non significatives. On ignore ce point dans ce TD.**

## 2 Représentation vectorielle des occurrences à désambiguïser

Etudiez les données annotées. Les 2 algos visés supposent de produire, pour une occurrence v à désambiguïser, une représentation vectorielle du contexte de v en utilisant en particulier les mots du contexte linéaire et/ou du contexte syntaxique de v.
Pour le contexte syntaxique, on peut utiliser l'arbre de dépendance "de surface" ou bien le graphe de dépendances profondes.

Détaillez le contenu possible de la représentation vectorielle de v, avec ou sans utilisation de vecteurs de mots.

Dans l'implémentation:

- lire les fichiers conll (d'abord seulement l'arbre de surface)
- pour chaque exemple, extraire la représentation vectorielle, sous forme de séquence d'identifiants numériques (identifiant de mot, ou de partie du discours ou autre, selon les traits que vous utilisez)
    - toutes les séquences doivent avoir les mêmes longueurs (utilisation de valeurs nulles pour se ramener à la taille maximale sur toutes les données ("padding")), et si vous utilisez plusieurs types d'information (mot, partie du discours…) vous devez conserver l'information sur quel tronçon correspond à quel type de trait
- lors du chargement, gérer au passage la correspondance entre identifiants numériques et mots / parties du discours / classes (=les sens)

## 3 Propagation de label

Voir l'article initial de (Zhu et Gharamani, 2002)[3] . La similarité utilisée comporte un paramètre sigma : sim $(x_i, x_j) = \exp( - dist(x_i,x_j) / sigma)$, avec $dist(x_i,x_j)$ = distance euclidienne.

Vous ferez un réglage manuel de cet hyperparamètre, même si c'est mal adapté: attention si les distances sont toutes trop grandes, les similarités seront approximées à 0, empêchant de calculer la similarité normalisée (division) par 0.
A l'inverse, vous constaterez que si les distances sont trop petites, l'algo tend à classer tous les exemples dans la classe majoritaire des exemples.

## 4 Apprentissage supervisé: réseau de neurones

- On intègre au réseau une couche d'embedding pour représenter les mots
    - Initialisés avec vecteurs pré-entrainés, et modifiés à l'apprentissage
    - Initialisés avec vecteurs pré-entrainés, et non modifiés à l'apprentissage
    - Initialisés au hasard, et modifiés à l'apprentissage

[3] Learning from Labeled and Unlabeled Data with Label Propagation. Xiaojin Zhu, Zoubin Ghahramani. *CMU CALD tech report CMU-CALD-02-107*, 2002