Analyse sémantique automatique

Pascal Amsili / Marie Candito

13-14

Contenu

- **Sémantique lexicale (en bref)**
  - unité lexicale
  - signifiants et ambiguïté
  - relations lexicales
  - représentation du sens d'une UL
  - rôles sémantiques
- **Sémantique lexicale computationnelle**
  - ressources pour l'anglais
    - réseau lexical : Wordnet
    - classes sémantiques : VerbNet
    - rôles sémantiques : FrameNet, PropBank
  - pour le français?
  - tâches
    - Calcul de similarité lexicale
    - **WSD** : désambiguisation lexicale (word sense disambiguation)
    - **SRL** : étiquetage de rôles sémantiques (semantic role labeling)
    - **RTE** : Reconnaissance d'inférence textuelle (recognizing textual entailment)
    - Lexical Acquisition
      - acquisition de cadres de sous-catégorisation
      - acquisition de préférences lexicales
      - reconnaissance de composés (multi-word units)

Plan du cours

- Introduction
- Définition formelle de la tâche de WSD
- Méthodes
  - WSD par apprentissage supervisé
  - WSD via thesaurus
  - WSD par apprentissage semi-supervisé
- Evaluation

Introduction

- **WSD** = identification du sens d'un *signifiant* dans un certain *contexte d'occurrence*

- Quel type d'ambiguïté traite le WSD?
  - le signifiant étant en général forme+lemme+cat
  - => WSD traite :
    - homonymie (avocat)
    - polysémie (bouteille, connaître)
  - cf. pour les autres types d'ambiguïté :
    - homographie de mots-formes (« couvent »)
      - généralement désambiguisé en fournissant les catégories morpho-syntaxiques
      - tâche de tagging, fonctionne assez bien
    - homophonie de mots-formes (ver/vert)
      - pas problématique à l'écrit (mais pb en reconnaissance vocale)

Motivations

- Traitement de TAL
  - l'unité de base est souvent le token, puis le lemme une fois le tagging réalisé
  - or les lemmes sont **ambigus**
  - par ex. les systèmes « bag of words » sont en général des « sacs de tokens » ou des « sacs de lemmes non désambiguisés »
- L'ambiguïté rajoute du bruit pour tout traitement de type TAL un peu sophistiqué
  - => Le WSD est donc potentiellement utile pour ces traitements
  - par exemple
    - traduction automatique
    - extraction d'information
    - question answering
    - classification de textes

Bref historique

- Problème identifié aux tout débuts de la **traduction automatique**
  - Weaver, 1949 : « a word can often only be translated if you know the specific sense intended (English *bill* could be *billet/addition* in French)
  - Bar-Hillel, 1960, rapport ALPAC, déclare le problème insoluble
    - *Little John was looking for his toy. Finally, he found it. The box was in the pen. John was very happy.*

"Assume, for simplicity's sake, that pen in English has only the following two meanings: (1) a certain writing utensil, (2) an enclosure where small children can play. I now claim that no existing or imaginable program will enable an electronic computer to determine that the word pen in the given sentence within the given context has the second of the above meanings, whereas every reader with a sufficient knowledge of English will do this 'automatically'."

Bref historique

- Systèmes par règles :
  - travaux dans les années 70/80 en intelligence artificielle
  - « word experts » : systèmes dédiés à la désambiguisation d'un mot
    - règles utilisant des mots du contexte
- Acquisition de règles
  - Yarowsky, 92 :
    - acquisition de règles de desamb des sens du thésaurus Roget,
    - en utilisant défs du thésaurus + stats de coocurrences extraites de corpus
- Années 90 :
  - mise à disposition de WordNet => fournit un inventaire de sens de référence
  - émergence des techniques par apprentissage supervisé
- Les travaux récents se concentrent sur :
  - réduire les besoins en données annotées
  - généraliser les données annotées
  - approches semi- et non-supervisées
  - utilisation de gros corpus (le Web)

Description de la tâche

- la tâche se formalise comme:
  - entrée =
    - un signifiant (en général la graphie d'un lemme, plus sa cat)
    - son **contexte** d'utilisation
    - et une **liste de sens** pour cette graphie (sense inventory)
  - sortie = le sens correct dans le contexte fourni

- rem : à distinguer de la tâche dite de « **sense discrimination** »
  - dégager les sens d'un mot, sans liste de sens pré-existante

Inventaires de sens

- WSD autonome (stand-alone WSD)
  - listes de sens fournies par un thésaurus ou un dictionnaire électronique

- WSD pour une tâche donnée
  - l'inventaire des sens est adapté à la tâche
    - traduction automatique
      - => ensemble de traductions possibles dans la langue cible
    - synthèse vocale
      - => ensemble d'homographes
    - indexation automatique par termes vedettes
      - => ensemble de termes possibles pour un descripteur non vedette

Contexte

- pour un humain, le contexte est tout le discours précédent
- pour la machine, le contexte peut être formalisé comme
  - la liste des mots formes / lemmes / lemmes+tag dans une fenêtre de x mots autour du mot cible
  - voire les dépendances syntaxiques autour du mot cible

Deux types de WSD

- « **Lexical sample WSD** »
  - réduction artificielle de la tâche:
    - = WSD ciblé sur un ensemble de mots(graphies)
  - cette réduction permet l'utilisation d'un apprentissage supervisé :
    - apprentissage sur données labelées:
      - pour chacun des mots cibles, sélection de phrases avec le mot
      - annotation du bon sens du mot cible

- « **All-words WSD** »
  - WSD pour tout mot sémantiquement plein
  - comparable au tagging
    - catégorie = sens
    - mais chaque lemme a son propre ensemble de sens
  - les techniques supervisées n'ont jamais assez de données
  - => voir les techniques via thésaurus
  - ou les techniques avec amorce (bootstrapping)

WSD supervisé

Classification supervisée

- le WSD se ramène à classer un objet (graphie de lemme) parmi x classes (ses x sens)
- données d'apprentissage =
  - paires objets+contexte, associées à leur classe
    - pour le WSD: des graphies en contexte, associées à leur sens
- apprentissage ou estimation de paramètres
  - (voir cours sur classification linéaire)
- classification : utilisation des paramètres appris pour classer un nouveau couple objet+contexte

Algorithmes d'apprentissage supervisé

- divers algorithmes classiques d'apprentissage ont été appliqué au WSD
  - arbres de décision
  - classifieur bayésien naïf
  - perceptrons / réseaux de neurones
  - modèles log-linéaires (maxent)
  - machines à vecteurs supports (SVM)
  - ...

Rappel: Classifieur bayésien naïf

- principe =
  - étant donné un objet o à classer
    - (o inclut son contexte)
  - **choisir la classe qui maximise** P(c | o)

$$\hat{c} = \arg\max_c P(c \mid o)$$

  - en utilisant Bayes, et le fait que P(o) est constant pour toute classe, on obtient :

$$\hat{c} = \arg\max_c P(c \mid o) = \arg\max_c \frac{P(o \mid c)P(c)}{P(o)} = \arg\max_c P(o \mid c)P(c)$$

Rappel: Classifieur bayésien naïf (2)

- Si o (+contexte) est représenté comme une liste de couples caractéristique+valeur pour l'objet o
  - les couples « caractéristique + valeur » forment les « traits » : (F_1=f_1, F_2=f_2,... F_n=f_n)
- on fait l'habituelle (fausse) hypothèse d'indépendance des traits, conditionnellement à la classe

$$\hat{c} = \arg\max_c P(o \mid c)P(c) =_{indep} \arg\max_c P(c)\prod_{i=1}^{n} P(F_i = f_i \mid c)$$

- plus précisément on passe aux logs pour éviter les produits trop faibles :

$$\hat{c} = \arg\max_c \left[\log(P(c)) + \sum_{i=1}^{n} \log(P(F_i = f_i \mid c))\right]$$

CNB : Estimation des paramètres

- Pour le WSD : instantiation par Gale et al. 1992 :
  - les objets sont des occurrences de graphies w_j
  - les classes sont des sens s_i
    - éventuellement s_i est spécifique à w_j
    - mais pas forcément (cf. synset wordnet par ex.)
  - les traits $F_i=f_i$ pour une occurrence de w_j sont « le i_{eme} élément ds la liste des n tokens pleins apparaissant dans le contexte de w_j vaut f_i »
    - le contexte peut être limité à une fenêtre autour de l'occurrence w_j
    - le même mot peut apparaître plusieurs fois ds le contexte (i.e un f_j peut correspondre au même mot qu'un f_k)
- estimation par fréquence relative
    - sur des graphies+contexte dont on connaît le sens désambiguisé:
    - rem: on considère que la position dans le contexte n'a pas d'impact

$$P_{MLE}(s_i) = \frac{nbocc(w_j \text{ avec sens } s_i)}{nbocc(w_j)}$$

$$P_{MLE}(f_k \mid s_i) = \frac{nbocc(f_k \text{ apparaît ds le contexte d'un mot avec sens } s_i)}{nbocc(s_i)}$$

CNB: lissage pour $P(f_k \mid s_i)$

- annulation en cas de mots inconnus:
  - si une graphie w_j à désambiguiser a dans son contexte un token f_k
  - tel que f_k correspond à un mot jamais vu ds le contexte d'une occurrence du sens s_1 de w_j
  - alors on estimera $P_{MLE}(f_k \mid s_1) = 0$
- => il est nécessaire de lisser
- => par ex. lissage de Laplace
  - ajouter 1 (ou une valeur 0< p <1) au numérateur
  - et modifier le dénominateur en conséquence

Performances du WSD supervisé

- le WSD supervisé est la technique la plus performante pour les mots pour lesquels on dispose de suffisamment d'exemples
- mais
  - constituer les données d'apprentissage est très fastidieux
  - dépendance du classifieur obtenu au genre/type de corpus dont sont extraits les exemples
  - échec sur les mots rares ou absents du corpus d'apprentissage
- d'ailleurs aucune donnée disponible pour le français !
  - ni pour la plupart des langues
- pour l'anglais, SemCor
  - extrait de 230000 tokens du corpus Brown
  - taggés avec synsets de WordNet

WSD via dictionnaire

WSD via dictionnaire

- Principe :
  - les définitions d'un dictionnaire/thésaurus servent de phrase contexte
  - graphie w_j, associé à x sens $sens_{ji}$
    - la définition du sens s_{ji} fournit un exemple pour w_j associé au sens s_{ji}
- Rem: l'inventaire de sens est alors forcément celui du dictionnaire

WSD via dictionnaire

- Exemple: « bank » en anglais

$bank_1$	Def: Examples:	a financial institution that accepts deposits and channels the money into lending activities « he cashed a check at the bank » « that bank holds the mortgage on my home »
$bank_2$	Def: Examples:	sloping land (especially the slope beside a body of water) « they pulled the canoe up on the bank » « he sat on the bank of the river and watched the currents »

- Occurrence à désambiguiser :

  *The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities*

- Qu'en pensez-vous?

WSD par dictionnaire : signatures

- signature sens s_i
  = ens. de mots (pleins) caractérisant le sens s_i
  = mots extraits du ou des contextes d'occurrences du sens s_i
    - extraction à partir d'exemples de mots taggés avec leur sens
    - extraction à partir de définition de dictionnaire

- pour l'exemple précédent :
  - signature de $bank_1$:
    - *financial, institution, accept, deposit, channel, money, lending, activity, cash, check, hold, mortgage, home*
  - signature de $bank_2$:
    - *sloping, land, body, water, pull, canoe, bank, sit, river*

WSD par dictionnaire : signatures

- différentes sources d'exemples pour classification supervisée :
  - définitions de dictionnaire
    - => chaque def fournit un exemple labelé : objet = graphie de lemme, classe= sens que décrit la définition)
  - occurrences en corpus
    - suppose que l'occurrence d'un lemme soit désambiguisée (associée à un sens)
    - pour peu que l'inventaire de sens soit le même que celui du dictionnaire
    - => on obtient des exemples supplémentaires

- les signatures sont un moyen de combiner les exemples associés au même couple graphie+sens
  - en un seul vecteur de mots du contexte,
  - les mots de contexte doivent être **pondérés** pour capturer leur différence de significativité

- **Exercice**: proposez comment pondérer les mots de contexte, dans ce cadre

WSD par dictionnaire : Lesk

- Lesk
  - = une famille d'algorithmes de WSD par dictionnaire
  - ce sont les algos les plus classiques
- variantes
  - Lesk standard (Lesk, 1986)
  - Simplified Lesk (Kilgarriff and Rosenszweig, 2000)
  - Corpus Lesk (Kilgarriff and Rosenszweig, 2000; Vasilescu et al., 2004)
  - ...

Simplified Lesk Algorithm

fonction simplifiedLesk(*mot*, *phrase*)
    meilleur_sens ← sens le plus fréquent de mot
                        (si l'info est dispo, sens random sinon)
    max_overlap ← 0
    contexte ← mots de phrase
    pour chaque *sens* existant pour mot
        signature ← signature(*sens*)
        overlap ← nb mots communs à *signature* / *contexte*
        si *overlap* > *max_overlap* alors
            max_overlap ← *overlap*
            meilleur_sens ← *sens*
    return *meilleur_sens*

Lesk « original »

- plus indirect
- mais fonctionne finalement moins bien (Kilgarriff & Rosenszweig, 2000)
- différence :
  - simplified Lesk : signature(sens de mot) comparé à contexte du mot
  - Lesk standard : signature(sens de mot) comparé à l'union des signatures(sens des mots du contexte de mot)

Lesk : Exemple célèbre

- Lesk, 1986 : *Automatic sense disambiguation: How to tell a pine cone from an ice cream cone*. Actes de SIGDOC 1986
  - désambiguiser *« cone »* dans le contexte de *« pine cone »*
  - étant donnés l'inventaire de sens suivant :

pine	1	kinds of evergreen tree with needle-shaped leaves
	2	waste away through sorrow or illness
cone	1	solid body which narrows to a point
	2	something of this shape whether solid or hollow
	3	fruit of certain evergreen trees

Corpus Lesk

- variante où pour compter l'overlap
  - *overlap* ← nb mots communs à *signature/contexte*
- on pondère chaque mot commun par un IDF calculé sur l'ens. des définitions
  - idf : log(nb total de déf. / nb déf. contenant le mot)

Lesk et exemples en corpus

- On peut utiliser Lesk en calculant les signatures en mettant à profit :
  - définitions
  - exemples présents dans dico
  - exemples issus de corpus taggés avec sens
- Inversement, on peut utiliser les définitions du dictionnaire pour augmenter les données d'apprentissage en WSD supervisé

WSD via dictionnaire : discussion

- intuition de départ exacte
  - les mots d'une définition sont pertinents pour désambiguiser une occurrence
- mais il en manque
  - les définitions sont « minimalistes »
  - peu de dictionnaires avec exemples complets
- diverses extensions pour répondre à ce pb:
  - en particulier utiliser les defs de mots sémantiquement reliés

WSD semi-supervisé

Motivations

- Le WSD par apprentissage supervisé, et/ou par dictionnaire requiert des données annotées manuellement
  - annotation très lourde
  - forcément parcellaire
  - disponible pour peu de langues

Algorithme par amorçage

- « **Bootstrapping** algorithm », Yarowsky, 1995
  - entraînement d'un désambiguisateur sur un petit ensemble L0 d'exemples (= couples occurrence + sens désambiguisé)
    - ensemble dit de « graines » (seed)
  - utilisation du désambiguisateur sur de nouvelles occurrences U0
  - sélection dans les résultats des couples occurrences +sens pour lesquelles le désambiguisateur est le plus sûr du sens choisi, et ajout à L0 pour obtenir L1
  - re-désambiguisation sur ensemble U0 moins ces exemples ajoutés, et extension de L1 en L2
  - etc…
  - itération jusqu'à ce qu'aucun exemple ne soit ajouté

Algorithme par amorçage (suite)

- ne peut bien fonctionner
  - que si les graines permettent d'obtenir un premier désambiguisateur fiable
    - et si les traits associés aux exemples graines sont suffisamment généraux pour être retrouvés dans de nouveaux exemples
  - et si le score de confiance est fiable

Algorithme par amorçage : graines

- L'ensemble L0 d'exemples est crucial
  - possibilité de sélectionner et tagger manuellement des occurrences avec leur sens
  - variante plus fainéante (!) : heuristiques de sélection de graines
    - « one sense per collocation » :
      - observation : certaines collocations sont particulièrement désambiguisante
        - par ex: « manger avocat » / « défendu par avocat »
      - extraction de collocations via scores d'association (PMI, Chi2 …)
      - puis affectation manuelle du sens pour chaque collocation
      - puis extraction automatique des exemples correspondants pour créer L0
    - « one sense per discourse » :
      - observation : le sens d'un mot est en général stable au sein d'un document (en tous cas pour homonymes)
      - => graines précédemment obtenues peuvent être complétées avec les autres occurrences au sein d'un même document

Problèmes ouverts

- La désambiguïsation obtenue pour un lemme ne généralise en général pas à d'autres lemme
- en effet :
  - WSD par dictionnaire :
    - généralisation sur plusieurs mots ssi utilisation de définitions/exemples de mots sémantiquement reliés aux sens à discriminer
  - WSD supervisé :
    - généralisation sur plusieurs mots ssi un sens est associé à plusieurs graphies (eg. synset wordnet)
      - si synset S contenant lemme1, lemme2
      - alors un exemple de lemme1 taggé avec sens S fournira des traits utilisables pour trouver le sens d'une occurrence de lemme2
  - Yarowsky 1995:
    - l'algorithme ne fait qu'améliorer la desambiguisation des mots déjà présents dans l'ensemble L0 des graines

Problèmes ouverts

- cette non généralisation
  - n'est pas intuitivement choquante si on ne considère que l'homonymie
    - pourquoi la désambiguisation de « avocat » aiderait-elle à la désambiguisation de « fraise » ?
  - mais intuitivement plus embêtant pour la polysémie régulière
    - la désambiguisation de « bouteille » devrait aider la désambiguisation de « verre »

Evaluation du WSD

Evaluation extrinsèque

- évaluation de l'amélioration obtenue dans une tâche utilisant le WSD
  - Traduction automatique
  - Recherche d'information, CLIR
    - voir par exemple une des tâches de la campagne CLEF : http://ixa2.si.ehu.es/clirwsd

- une telle évaluation
  - peut prouver la pertinence du WSD pour une tâche (ou pas)
  - mais l'évaluation n'est pas généralisable à d'autres tâches
  - en cas de non-gain pour une tâche, on ne peut pas forcément trancher entre
    - ce WSD n'est pas (assez) bon
    - le WSD n'est pas pertinent pour cette tâche

Evaluation intrinsèque

- Evaluation du WSD comme tâche autonome
- Evaluation supervisée
  - nécessite des données taggées avec sens,
  - avec le même inventaire de sens que le système à évaluer
  - => simple précision (accuracy): proportion d'occurrences pour lesquelles le sens prédit est bien celui des données de référence

- Données d'évaluation disponibles
  - données d'évaluation produites et rendues disponibles dans le cadre de tâches SENSEVAL
    - http://www.senseval.org/
  - tâche « lexical sample », « all-words » …
  - surtout l'anglais
    - données SENSEVAL-1, -2, -3: corpus taggés avec synset WordNet pour respectivement 34, 73 et 57 lemmes cibles
    - SemCor : SENSEVAL-3: corpus de 5000 tokens extrait du WSJ et du Brown, taggés avec sens
  - mais également:
    - Chinois (sens de la base de connaissances **HowNet**)
    - Italien (synset de ItalWordNet
    - …

Evaluation intrinsèque: pseudo-words

- données d'évaluation pas forcément disponibles
- => possibilité de construire des données artificielles
- pseudo-mots:
  - on utilise des paires de mots non ambigus
    - banane / maison
  - la paire est ambigüe, ses 2 sens sont les 2 mots de départ
  - pour toute occurrence d'un des 2 mots,
    - on remplace le mot par la paire
    - le sens associé à cet exemple est le mot de départ
- => division de ces données artificielles en
  - données d'apprentissage
  - données d'évaluation
- facile, peu coûteux
- mais la tâche évaluée est plus simple qu'en réalité
  - en particulier, pas d'évaluation sur cas polysémiques

Evaluation intrinsèque : baselines

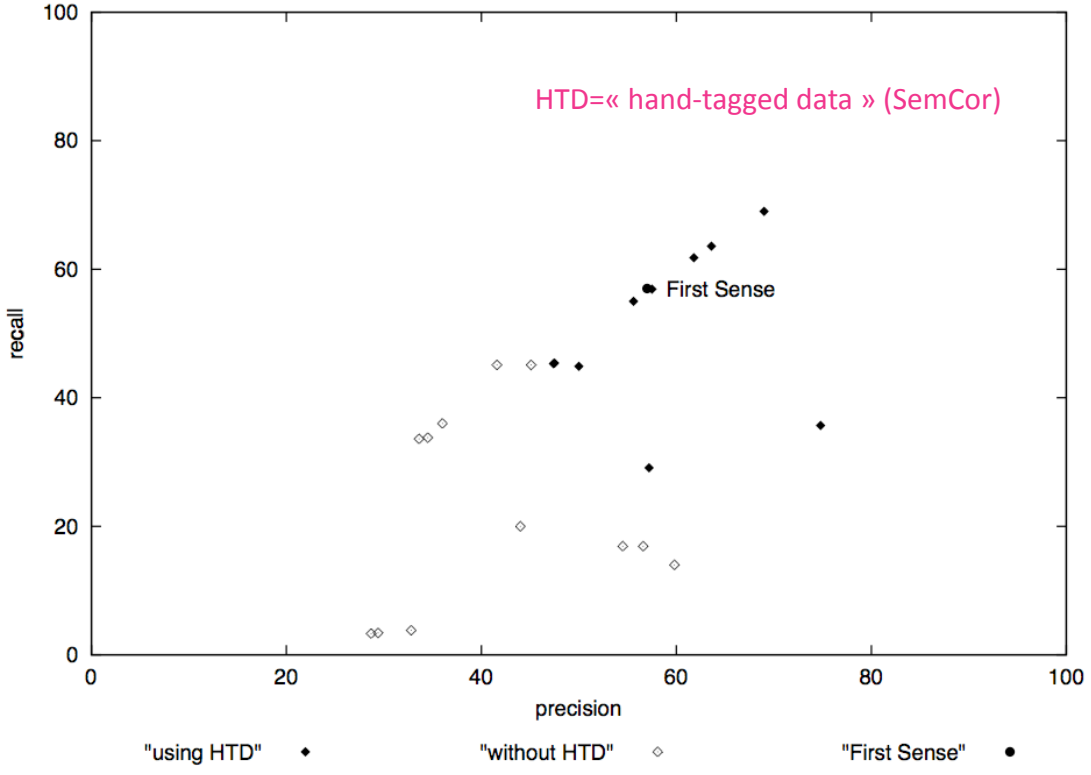
- système « baseline »
  - = système simple, facile à mettre en oeuvre, ou ancien/bien établi
  - toute évaluation devrait être confrontée à une baseline
    - que vaut un tagger obtenant une précision de 88% ?
- baselines pour le WSD:
  - Simplified Lesk
  - sens le plus fréquent
    - simplement associer à un lemme son sens le + fréquent
    - très très simple, pourvu que l'on ait un corpus taggé en sens!
    - très performante, car la majorité des distributions de sens sont biaisées (« skewed ») (plus précisément zipfienne)

« First sense baseline »

- ds WordNet :
  - most frequent sense = **first sense**
  - (pour les lemmes couverts dans le corpus SemCor)

- la baseline « most frequent sense » (ou « first sense (in WN) ») est très haute
  - cf. comparaison (McCarthy et al., ACL 2004) entre
    - les résultats des différents compétiteurs pour la tâche « all-words » en anglais à SENSEVAL-2
      - http://www.hipposmond.com/senseval2/
      - et la baseline first-sense
- D'après vous pourquoi?

First sense baseline vs SENSEVAL-2 systems (McCarthy et al. ACL 2004)



Automatic First Sense (McCarthy et al. ACL 04)

- la baseline du sens le + fréquent est puissante mais
  - nécessite un **corpus annoté en sens** pour identifier les sens les + fréquents
    - disponible pour peu de langues
  - et le résultat dépend
    - du genre/domaine du corpus
      - 1er sens de « tiger » dans SemCor = « audacious person »
      - 1er sens de « star » dans SemCor = « celestial body »
    - de la couverture du corpus
      - repli sur ordre arbitraire des sens en cas de mot inconnu du corpus annoté
- => AFS = méthode d'acquisition sur **corpus brut** du synset le plus fréquent d'un lemme **sans utilisation de corpus taggé avec sens**

Automatic First Sense (McCarthy et al. ACL 04)

- pré-requis :
  - corpus brut (lemmatisé)
  - thesaurus distributionnel fournit les K plus proches voisins d'un lemme, avec un score de similarité
  - une mesure de similarité utilisant WordNet
    - cf. cours similarité lexicale mesure utilisant l'information content du LCS (Resnik, Jiang & Conrad…)
    - mesure de type Lesk (overlap) calculée sur Wordnet …
- technique =
  - pour chaque sens d'un lemme, calculer un **score de prévalence** de ce sens dans le corpus brut
  - ordonner les sens d'après ce score

AFS : score de prévalence

- Notations
  - un lemme w,
  - les sens de w : sens(w) = {sw_1, …, sw_n}
  - les lemmes voisins distributionnels: voisins(w) = {v_1,…,v_K}
  - les scores de similarité distributionnelle : ds(w,v_1), …ds(w,v_k), … ds(w,v_K)
- Exemple : w = « avocat »
- sw1 = avocat_le_fruit
- sw2 = avocat_la_profession
- liste des voisins d'après corpus journalistique:
  - couples v_k / ds(avocat, v_k)

  procureur:0.300 Maître:0.272 responsable:0.263
  expert:0.261 diplomate:0.247 journaliste:0.245 source:0.244
  ministre:0.239 président:0.237 analyste:0.233 magistrat:0.232
  parquet:0.225 communiqué:0.223 proche:0.223 syndicat:0.222
  maire:0.222 représentant:0.222 juge:0.219 média:0.216
  autorité:0.216 député:0.214 défenseur:0.213 ministère:0.208
  Monsieur:0.208 entourage:0.207 quotidien:0.203
  syndicaliste:0.200 journal:0.200 secrétaire:0.199
  chercheur:0.198 enquêteur:0.197
- (extrait de FreDist (Henestroza et Denis, TALN 2011))

AFS : score de prévalence

- score de prévalence du sens sw_i de w,
- = somme sur tous les voisins v_k de w
  - de la wordnet-similarity de sw_i et du sens de v_k le+ proche, normalisée sur tous les sens de w pondéré par la sim distributionnelle $ds(w,v_k)$

$$prev\_score(sw_i) = \sum_{v_k \in \text{voisins}(w)} ds(w,v_k) \frac{\text{maxwns}(sw_i,v_k)}{\sum_{sw_j \in \text{sens}(w)} \text{maxwns}(sw_j,v_k)}$$

$$\text{avec } \text{maxwsn}(sw_i,v_k) = \max_{sv_{kl} \in \text{sens}(v_k)} \text{wordnetsim}(\text{sw}_i,\text{sv}_{kl})$$

Evaluation intrinsèque: plafonds

- Scores **plafonds** d'une tâche =
  - pour une tâche dont on soupçonne une **difficulté même pour un humain**, le score plafond est la performance d'**humains** sur la même tâche
    - en l'occurrence, désambiguiser en cas de polysémie peut être difficile
  - en outre, la performance obtenue manuellement doit être associée à une mesure de confiance
    - l'humain a-t-il répondu de manière incohérente en cas d'incertitude
    - plus on a d'incertitude, plus le résultat peut être incohérent
  - => pour cela on mesure l'accord obtenu sur la tâche, par 2 humains, avec les mêmes consignes
    - = « **accord inter-annotateur** »
- dans le cas du WSD
  - accord inter-annotateur = % de sens choisis par annotateur1 et annotateur2
- les niveaux d'accord inter-annotateur varient avec la granularité de l'inventaire des sens
  - sur synsets WN (anglais) : accord autour de 70% pour les 30 V les plus polysémiques et 85% pour les noms et adjectifs
  - autour de 90% pour des inventaires moins fins
- rem: les scores pour l'homonymie sont globalement bcp plus hauts que pour la polysémie