- **Problema:**

  Fiind dat sirul de intrare
  In aceasta familie masa se ia la ore fixe.
  sa se conceapa un soft care sa inteleaga sensul sirului de intrare.

- **Indicatie:**
  Pentru a se intelege sensul sirului de intrare, softul trebuie, mai intai, sa determine cu ce sens este folosit aici cuvantul polisemantic *masa*. (Trebuie sa se determine faptul ca *masa* este folosit cu sensul de mancare si nu cele de mobila, multime – masa de oameni, masa din fizica etc.)

- **Terminologie:**
  ***Masa* este un cuvant polisemantic (cu mai multe sensuri), deci ambiguu. Precizarea sensului cu care *masa* este folosit in sirul de intrare dat reprezinta operatia de dezambiguizare a sensului.**
- **WSD ≡ Word Sense Disambiguation (dezambiguizare automata a sensului cuvintelor polisemantice)**

O retea semantica pentru reprezentarea si procesarea limbajului natural
WordNet (WN)

- **WordNet (WN) este o baza de date lexicala a limbii engleze, conceputa si creata de Profesorul George Miller, la Universitatea Princeton, in anii '90. Ea este permanent actualizata de catre un colectiv de cercetatori.**

- **WN cuprinde toate "cuvintele cu continut" ale limbii engleze, adica toate substantivele, adjectivele, verbele si adverbele. Substantivele si verbele sunt organizate in ierarhii, iar adjectivele si adverbele in clustere.**

- **"Piatra de temelie" in WN o constituie *synset*-ul, sau multimea de sinonime.**

- **Aici cuvinte sinonime inseamna cuvinte care se refera la un acelasi *concept*. (De pilda, cuvintele *scaun* si *masa* sunt considerate aici sinonime, intrucat ambele se refera la conceptul de "mobila".)**

- **Fiecare *synset* din WN se refera la un anumit *concept*.**

- **Toate sinonimele care apar intr-un synset au aceeasi parte de vorbire (substantiv, adjectiv, verb sau adverb).**

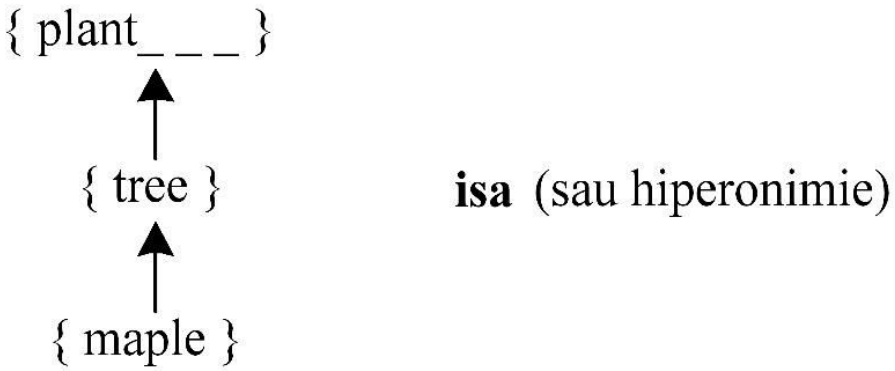
- **Un cuvant polisemantic (cu mai multe sensuri) se refera la un alt concept prin fiecare dintre sensurile lui. Prin urmare, un cuvant polisemantic va interveni in mai multe synset-uri WN. El apartine cate unui synset prin fiecare sens al lui. Spre exemplu, cuvantul polisemantic *masa* poate apartine fiecaruia dintre synset-urile care se refera la conceptele de *mancare, mobila, multime, masa din fizica* etc.**

- **Un synset WN contine toate sinonimele care lexicalizeaza conceptul la care synset-ul se refera si care au aceeasi parte de vorbire (substantiv, verb etc.), precum si un string numit *glosa*, care seamana cu o definitie clasica a sensului (cum sunt cele din dictionar). Glosa poate contine si un exemplu de utilizare a sinonimelor din synset.**

- **Synset-urile WN sunt legate intre ele prin *relatii semantice.* Aceste relatii leaga intre ele conceptele la care se refera synset-urile si care sunt aceleasi in toate limbile (conceptele sunt independente de limba).**

Synset-uri de substantive

Synset-urile substantivale sunt legate intre ele prin relatii semantice cum ar fi *hiperonimia* si inversa acestei relatii, *hiponimia*, care reprezinta relatia ISA din inteligenta artificiala in domeniul procesarii limbajului natural. Pe baza acestei relatii este construita ierarhia substantivala din WN:



Hiperonimul este conceptul parinte, iar hiponimul este conceptul fiu. Mecanismul de inferenta este mostenirea proprietatilor. (Hiponimele mostenesc toate proprietatile hiperonimului).

Synset-uri substantivale – continuare:

- **alta relatie semantica importanta in cazul substantivelor este *meronimia* sau relatia *parte-din* (“the *part-of* relation”). Inversa acestei relatii este *holonimia*.**

Exemplu: substantivele *clanta* si *usa*.

(Clanta este parte din usa. Substantivul *usa* este holonim al substantivului *clanta*. Iar *clanta* este meronim al lui *usa*).

WordNet

- **Structura de retea semantica a WN este data de relatiile semantice care leaga synset-urile intre ele.**

- **Prin intermediul relatiei de hiperonimie conceptele mostenesc toate proprietatile conceptelor parinte. Mostenirea proprietatilor transforma WN intr-o baza de cunostinte.**

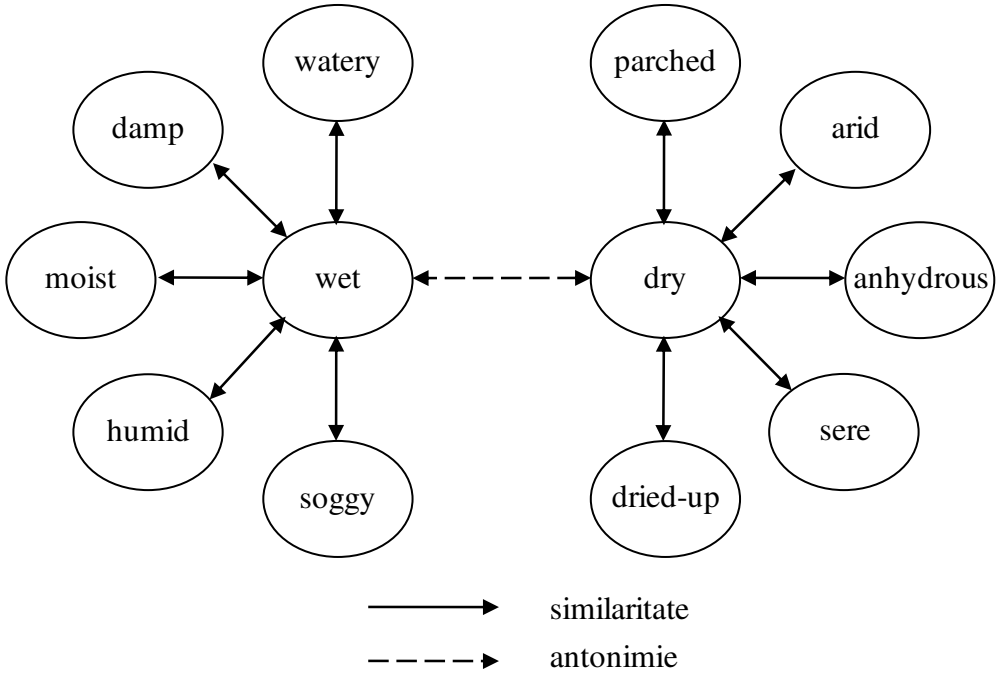
- **Concluzie: WN este o baza de date lexicala a limbii engleze, o retea semantica si o baza de cunostinte. Faptul ca WN reprezinta o baza de cunostinte o face extrem de utila in diverse aplicatii din IA.**

SYNSET-uri de adjective in WN

Relatii semantice de o alta natura leaga intre ele synset-urile de adjective din WN. Principala relatie semantica este considerata a fi *antonimia*.

Prin intermediul relatiei de antonimie synset-urile de adjective sunt organizate in *cluster*-e. Mai jos este dat ca exemplu cluster-ul format in jurul antonimelor

***wet / dry*:**



Synset-uri de adjective

Relatia de *similaritate* exista in WN numai intre synset-uri de adjective. Ea aduce, in plus, antonime indirecte. (In exemplul dat, *wet* si *dry* sunt *antonime directe*, iar *wet* si *arid* sunt *antonime indirecte*. Adjectivul *moist* nu are un antonim direct, dar un antonim indirect al lui poate fi gasit urmand drumul *moist* $\rightarrow$ *wet* $\rightarrow$ *dry*).

S-a dovedit ca relatia de *antonimie* furnizeaza informatie importanta procesului de dezambiguizare automata a sensului.

Alte relatii semantice importante in cazul adjectivelor sunt: *also-see*, *pertaining-to*, *attribute*.

SYNSET-URI DE VERBE SI ADVERBE

- **Synset-urile de verbe sunt organizate in ierarhii, ca si cele de substantive, prin intermediul relatiei *entailment* (to entail = a atrage dupa sine).**

- **Alta relatie semantica importanta pentru verbe este relatia *cauzala.***

- **Synset-urile de adverbe sunt organizate in cluster-e, ca si cele de adjective.**

Similaritate si inrudire
(intre concepte si cuvintele care le lexicalizeaza)

- **Inrudirea semantica este un concept mai general decat similaritatea semantica. Similaritatea este un caz particular de inrudire.**
  Exemplu: *Masina* si *anvelopa* nu sunt similare, dar sunt inrudite (anvelopa este o "parte din" masina). *Doctorii* si *spitalele* nu sunt concepte similare, dar sunt inrudite.

- **In literatura exista diverse masuri pentru calcularea similaritatii si/sau inrudirii. Vom vedea o asemenea masura in algoritmul care urmeaza.**

- **S-a demonstrat ca, pentru a realiza dezambiguizarea automata a sensului cuvintelor, conceptul de inrudire este mai util decat cel de similaritate.**

- **Studiem, in continuare, un algoritm de dezambiguizare a sensului cuvintelor bazat pe conceptul de inrudire si care foloseste cunostinte furnizate de reteaua semantica WordNet.**

- **Problema:**
  Fiind dat sirul de intrare
  In aceasta familie masa se ia la ore fixe.
  sa se conceapa un soft care sa inteleaga sensul sirului de intrare.

- **Indicatie:**
  Pentru a se intelege sensul sirului de intrare, softul trebuie, mai intai, sa determine cu ce sens este folosit aici cuvantul polisemantic *masa*. (Trebuie sa se determine faptul ca *masa* este folosit cu sensul de mancare si nu cele de mobila, multime – masa de oameni, masa din fizica etc.) Algoritmul care urmeaza dezambiguizeaza cuvantul ambiguu *masa*.
  Cuvantul de dezambiguizat ≡ cuvant tinta

Algoritmul Lesk extins

- **Banerjee si Pedersen (2003) prezinta o noua masura a inrudirii semantice intre concepte, care se bazeaza pe numarul de cuvinte pe care le au in comun definitiile lor (suprapuneri de glose – "gloss overlaps").**

- **Aceasta masura primeste ca input doua concepte (reprezentate de doua synset-uri WN) si ofera ca output o valoare numerica, care cuantifica gradul lor de inrudire semantica. Aceasta valoare numerica este apoi folosita pentru a realiza dezambiguizarea sensului.**

- **Masura de inrudire si algoritmul de dezambiguizare corespunzator reprezinta o varianta a Algoritmului lui Lesk clasic (care se bazeaza pe suprapuneri de glose). Aceasta varianta extinde glosele conceptelor luate in consideratie.**

- **Extinderea gloselor se face astfel incat sa fie luate in consideratie si glosele altor concepte, inrudite cu cele date prin intermediul relatiilor semantice furnizate de WN.**

Algoritmul lui Lesk

Suprapunerile de glose (definitii) au fost introduse de Lesk (1986) cu scopul de a servi in dezambiguizarea sensului.

Algoritmul lui Lesk atribuie un sens unui cuvant tinta (de dezambiguizat), intr-un context dat, prin compararea gloselor diverselor sensuri ale cuvantului cu cele ale celorlalte cuvinte din context. Sensul cuvantului tinta a carui glosa are cele mai multe cuvinte in comun cu glosele cuvintelor invecinate este cel atribuit cuvantului tinta.

Exemplu: consideram urmatoarele glose ale cuvintelor *car* si *tire*

car*: *four wheel motor vehicle usually propelled by an internal combustion engine

tire*: *hoop that covers a wheel, usually made of rubber and filled with compressed air

- ✓ **glosele acestor concepte au in comun cuvantul *wheel***

Limitare: Algoritmul Lesk original ia in consideratie numai suprapunerile dintre glosele cuvantului tinta si cele ale cuvintelor din jurul lui, in contextul dat. Definitiile de dictionar sunt insa scurte si nu furnizeaza un vocabular suficient de mare (vor fi putine cuvinte comune).

- **Masura suprapunerii extinse de glose introdusa de Banerjee si Pedersen (2003) extinde glosele conceptelor luate in consideratie prin includerea gloselor conceptelor inrudite cu acestea. Conceptele inrudite sunt alese prin intermediul relatiilor semantice din WN.**

Masura suprapunerii extinse de glose

- **Atunci cand masuram inrudirea dintre doua synset-uri de input, cautam suprapuneri nu numai intre glosele lor, dar si intre glosele synset-urilor hiperonime, hiponime, meronime, holonime etc. ale lor.**

- **Alte relatii WN de luat in consideratie sunt attribute, similar-to, also-see, antonymy etc.**

- **Observatie: NU toate aceste relatii au aceeasi importanta in procesul de dezambiguizare. Alegerea relatiilor depinde de partea de vorbire a cuvantului tinta (substantiv, adjectiv, verb, adverb), dar si de tipul de aplicatie pentru care se studiaza inrudirea.**

- **Vom aplica aici aceasta masura a inrudirii in procesul de dezambiguizare.**

Exemplu de utilizare a relatiilor semantice din WN:

In cazul substantivelor testele empirice au demonstrat ca este util in dezambiguizare sa folosim glosele hiponimelor si meronimelor synset-urilor de input. Observati ca se recomanda folosirea hiponimelor (conceptul fiu) si nu a hiperonimelor (conceptul parinte) sau a ambelor. Cu alte cuvinte, s-a demonstrat ca hiponimia furnizeaza mai multa informatie decat hiperonimia, desi ambele formeaza relatia ISA.

Mecanism de acordare a scorurilor

- **Algoritmul Lesk iniţial compară glosele unei perechi de concepte şi calculează un scor prin numărarea cuvintelor pe care acestea le au în comun. Acest mecanism de acordare a scorurilor nu diferenţiază între suprapuneri de cuvinte unice şi suprapuneri de grupuri de cuvinte, ci tratează fiecare glosă ca pe un „sac de cuvinte". Spre exemplu, acorda scorul 3 conceptelor *drawing paper* si *decal*, care au urmatoarele glose:**

  drawing paper: paper that is specially prepared for use in drafting
  decal: the art of transferring designs from specially prepared paper to a wood or glass or metal surface.

  Aici exista 3 suprapuneri, cuvantul *paper* si grupul de cuvinte *specially prepared*.

- **Banerjee si Pedersen (2003) atribuie unei suprapuneri de *n* cuvinte scorul n^2 (pentru ca o suprapunere de *n* cuvinte consecutive este mult mai rara decat suprapunerile de cuvinte unice). Pentru perechea anterioara de glose se atribuie scorul 5 (in loc de 3).**

Algoritm de acordare a scorurilor

1. **Fiind date două şiruri, se detectează cea mai lungă suprapunere dintre acestea. Dacă două sau mai multe astfel de suprapuneri au aceeaşi lungime maximă, atunci este raportată acea suprapunere care intervine prima în primul şir care se compară. Cea mai lungă suprapunere detectată este înlăturată, iar în locul ei se plasează un marcaj în fiecare dintre cele două şiruri constituind input-ul. Cele două şiruri astfel obţinute sunt apoi din nou verificate pentru suprapuneri, iar acest proces continuă până când nu mai există suprapuneri între ele.**

2. **Se atribuie scoruri tuturor suprapunerilor găsite (dimensiunile suprapunerilor detectate sunt ridicate la pătrat) şi aceste scoruri sunt adunate pentru a se determina *scorul perechii de glose date*.**

Calcularea scorului de inrudire dintre doua synset-uri A si B

1. **Definim mulţimea RELS ca pe o mulţime nevidă de relaţii constând din una sau mai multe dintre relaţiile puse la dispoziţie de WordNet:**

$$RELS \subset \{r \mid r \text{ este o relaţie definită în WordNet}\}.$$

Presupunem că funcţia fiecărei relaţii *r* (*r* ∈ RELS) este cea dată de numele relaţiei, care acceptă un synset de input şi întoarce glosa synset-ului (sau synset-urilor) înrudite cu cel de input prin relaţia desemnată. Spre exemplu, dacă *r* ∈ RELS reprezintă relaţia de hiperonimie, atunci *r*(*A*) întoarce glosa unui synset hiperonim al lui *A*. *r* mai poate reprezenta „relaţia glosă”, caz în care *r*(*A*) întoarce glosa synset-ului A.

OBS.: Dacă mai multe synset-uri sunt înrudite cu cel de input prin intermediul aceleiaşi relaţii, glosele acestora vor fi concatenate şi returnate ca un unic şir de caractere. Această concatenare este efectuată deoarece nu se doreşte să se facă o diferenţiere între synset-uri care sunt înrudite cu cel de input printr-o aceeaşi relaţie, de interes fiind doar glosele acestora. Atunci când niciun synset nu este înrudit cu cel de input prin relaţia dată, va fi returnat şirul vid.

Definim o nouă mulţime nevidă, de *perechi de relaţii*, selectate din mulţimea RELS. Singura constrângere în formarea acestor perechi de relaţii va fi aceea că, dacă este aleasă perechea (r_1, r_2), atunci va trebui aleasă şi perechea (r_2, r_1), astfel încât să se asigure reflexivitatea măsurii de înrudire (r_1, $r_2 \in$ RELS). Cu alte cuvinte, trebuie să se verifice *înrudire* (A, B) = *înrudire* (B, A). Vom numi această nouă mulţime RELPAIRS. Ea se defineşte după cum urmează:

2. RELPAIRS = {(R_1, R_2) $\mid R_1, R_2 \in$ RELS; dacă (R_1, R_2) $\in$ RELPAIRS, atunci (R_2, R_1) $\in$ RELPAIRS}.

3. Presupunem că *scor ( )* este o funcţie care acceptă ca input două glose, găseşte grupurile de cuvinte care se suprapun ale acestora şi întoarce un scor calculat conform Algoritmului de acordare a scorurilor descris anterior.

- **Folosind toate aceste elemente, calculam scorul de inrudire dintre synset-urile A si B dupa cum urmeaza:**

$$\textit{înrudire}\ (A, B) = \sum_{\forall (R_1, R_2) \in \mathrm{RELPAIRS}} \textit{scor}\ (R_1(A), R_2(B)).$$

Această măsură de înrudire se bazează pe mulţimea tuturor perechilor posibile de relaţii furnizate de reţeaua semantică WordNet. În practică, vor fi alese acele relaţii care s-au dovedit a fi cele mai utile corespunzător fiecărei părţi de vorbire.

<u>Exemplu:</u>

Vom presupune că mulţimea de relaţii aleasă este RELS = {gloss, hype, hypo} (unde *hype* şi *hypo* reprezintă relaţiile de hiperonimie şi, respectiv, hiponimie). În continuare, vom presupune că mulţimea perechilor de relaţii RELPAIRS este dată de RELPAIRS = {(gloss, gloss), (hype, hype), (hypo, hypo), (hype, gloss), (gloss, hype)}. Atunci, înrudirea dintre synset-urile *A* şi *B* poate fi calculată după cum urmează:

$$\textit{înrudire}\ (A,B) = \textit{scor}\ (\textit{gloss}(A), \textit{gloss}(B)) + \textit{scor}\ (\textit{hype}(A), \textit{hype}(B)) + \textit{scor}\ (\textit{hypo}(A), \textit{hypo}(B)) + \textit{scor}\ (\textit{hype}(A), \textit{gloss}(B)) + \textit{scor}\ (\textit{gloss}(A), \textit{hype}(B)).$$

Se observă că alegerea acestor perechi de relaţii asigură egalitatea *înrudire* (*A*,*B*) = *înrudire* (*B*,*A*).

Aplicatie in dezambiguizarea sensului cuvintelor

Vom selecta o fereastră de context în jurul cuvântului ţintă (de dezambiguizat) şi vom presupune că această fereastră de context constă din $2n + 1$ cuvinte notate w_i, $-n \leq i \leq +n$, unde cuvântul ţintă este w_0. Pentru fiecare cuvânt cu conţinut din fereastra de context este identificată o mulţime de sensuri candidate. Fie $|w_i|$ numărul de sensuri candidate ale cuvântului w_i. Aceste sensuri vor fi notate prin $s_{i,j}$, $1 \leq j \leq |w_i|$. Fiecărui sens k posibil al cuvântului ţintă îi vom atribui un scor notat $SenseScore_k$ şi calculat prin adunarea scorurilor de înrudire obţinute la compararea sensului analizat al cuvântului ţintă cu fiecare sens al fiecărui cuvânt cu conţinut care nu este cuvânt ţintă din fereastra de context. Scorul sensului $s_{0,k}$ este calculat după cum urmează:

$$SenseScore_k = \sum_{i=-n}^{n} \sum_{j=1}^{|w_i|} \hat{i}nrudire(s_{0,k}, s_{i,j}), \quad i \neq 0$$

Acel sens care are scorul cel mai ridicat este considerat a fi cel mai adecvat pentru cuvântul ţintă. Dacă mai multe sensuri au scor egal, atunci, prin convenţie, se alege acel sens care este considerat cel mai important (uzual) conform WordNet. O combinaţie candidată de glose care nu prezintă nicio suprapunere va primi scorul zero.

Complexitate

Dacă, în medie, există α sensuri pentru fiecare cuvânt, iar fereastra de context conţine N cuvinte, vom avea $\alpha^2 \times (N - 1)$ perechi de mulţimi de synset-uri care trebuie comparate, ceea ce reprezintă o creştere liniară în raport cu N.