Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria – sede di Modena
Corso di Laurea VOD in Ingegneria Informatica

PROGETTO E REALIZZAZIONE DELL' ALGORITMO DI ANNOTAZIONE AUTOMATICA TUCUXI

Relatore:
Chiar.mo Prof. Sonia Bergamaschi

Candidato:
Stefano Gatta

Correlatore:
Dott. Ing. Serena Sorrentino

Disambiguazione lessicale

- ✓ **Processo appartenente all'area della linguistica computazionale (WSD)**
- ✓ **Fondamento: polisemia degli elementi lessicali del linguaggio naturale**

Assegnare ad ogni termine, a partire da un testo o da una porzione di testo, il significato più corretto in base al ***contesto*** in cui il termine compare.

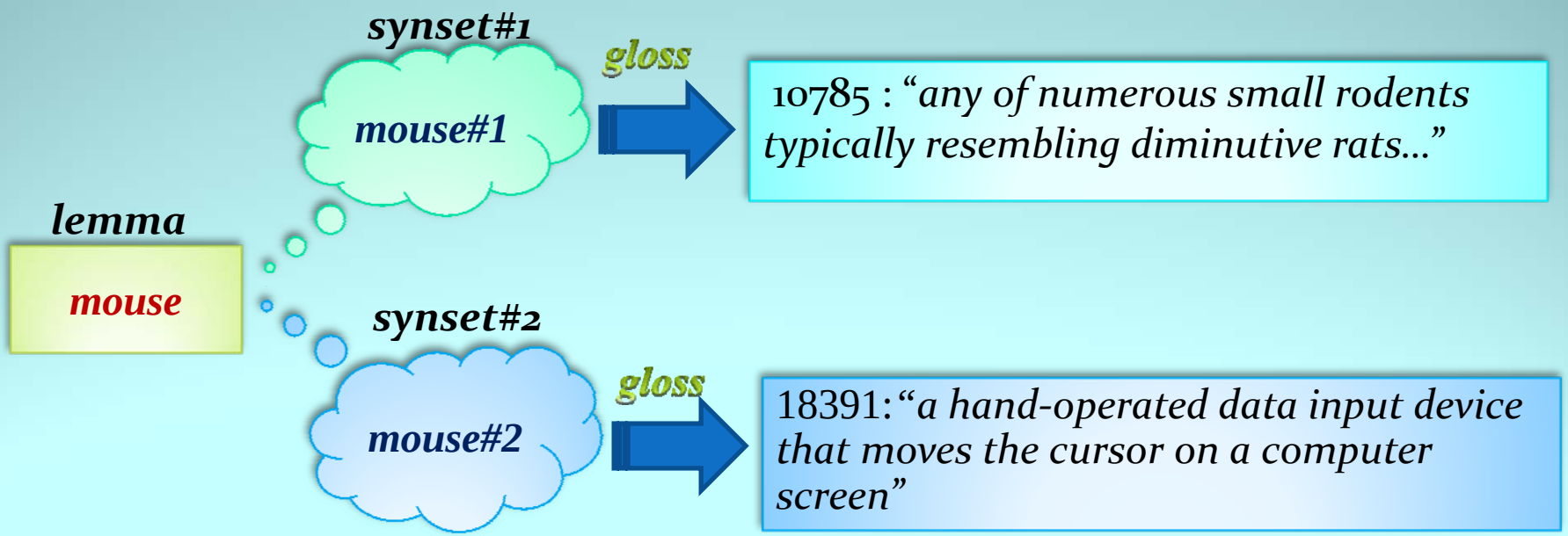
manualmente : ***corpora*** **e testi di riferimento precedentemente annotati**

automaticamente : **algoritmi che usano diversi approcci, tra cui LKB (basati su una risorsa di conoscenza lessicale)**

WordNet

L'ontologia lessicale WordNet

- Effettua una prima distinzione fondamentale tra **word form** (lemma) e **word meaning** (significato)
- È organizzata in insiemi di concetti che possono essere espressi da uno o più lemmi: **synset** (set of synonymous)
- Ad uno stesso lemma sono associati uno (lemma *monosemico*) o più (lemma *polisemico*) synset



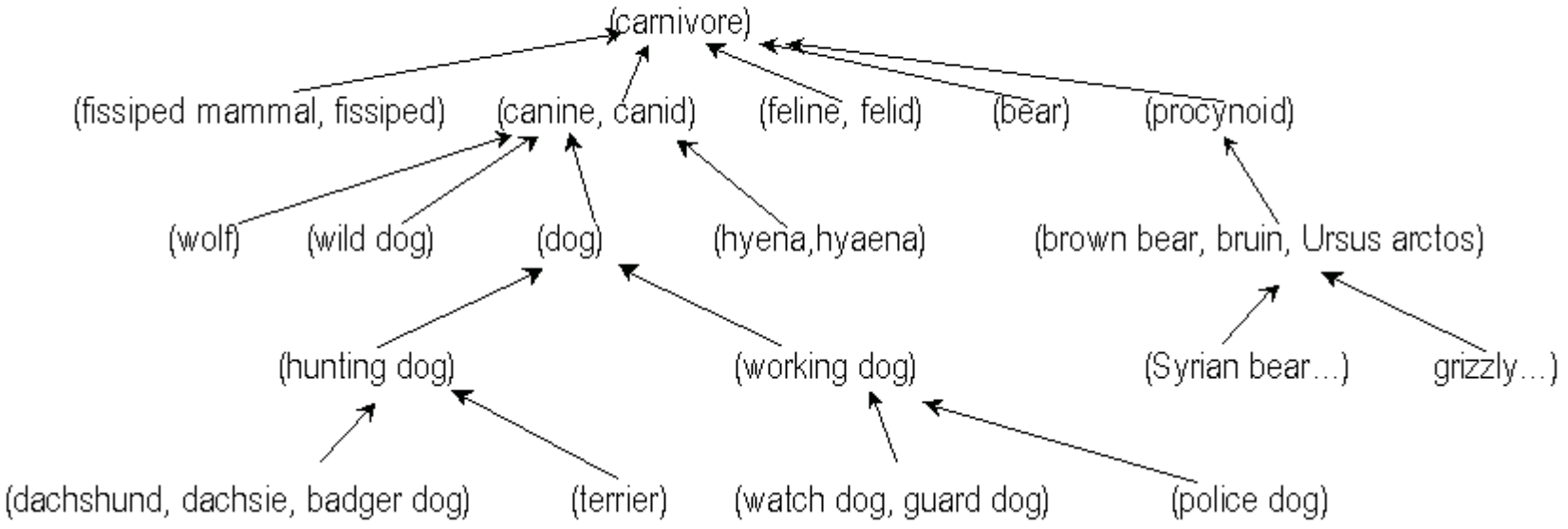
L’ ontologia lessicale WordNet

- Divide gli elementi lessicali in 4 categorie sintattiche : **nomi, verbi, aggettivi, avverbi**
- Offre la possibilità di risalire alle **relazioni** esistenti tra elementi della stessa categoria sintattica

Relazioni lessicali tra lemmi

Relazioni semantiche tra synset

Gerarchia di relazioni tra synset



L'ontologia lessicale WordNet

Carenze di WordNet:

- Insieme limitato di relazioni codificate
- Livello eccessivo di granularità dei synset (es. 40 synset per un lemma)

WordNet Domains

✓Introdotto nel 2002 da Magnini e Strapparava dell'ITC-irst di Trento.

✓ Sfrutta il concetto di **dominio**

✓Associa, a ciascun synset di WordNet, una etichetta che ne indichi il dominio di appartenenza.

Applicazioni della disambiguazione lessicale

- Metodi di *crawling* del Web basati sul contenuto semantico delle *keywords*
- Supporto ai sistemi di Integrazione Intelligente delle Informazioni

✓ **Soluzione al problema dell' eterogeneità tra le sorgenti di informazione**

Il mediatore Momis

MOMIS

(*Mediator EnvirOment for Multiple Information Sources*)

- ✓ integra sorgenti di dati strutturate e semi-strutturate in uno schema globale virtuale
- ✓ consente all'utente di sottomettere le query direttamente allo schema globale ricevendo una risposta unificata.

Problemi da affrontare → **Problemi ontologici**

Problemi da affrontare → **Problemi semantici**

Problemi semantici ← **Estrazione della semantica**

Estrazione della semantica → **Disambiguazione lessicale**

Disambiguazione lessicale = **Annotazione lessicale delle sorgenti**

Annotazione lessicale delle sorgenti ← Processo manuale noioso e *time-consuming*

Obiettivo : **rendere automatico l'oneroso compito dell'annotazione delle sorgenti di dati**

Il tool ALA di Momis

Automatic Lexical Annotator

- Utilizza WordNet come risorsa lessicale di riferimento
- Combina 5 algoritmi di disambiguazione che sfruttano approcci diversi
- Fornisce un'annotazione di tipo probabilistico

✓ Manca un approccio basato sulle relazioni semantiche tra gli elementi degli schemi

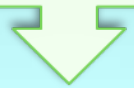
TUCUXI

(InTelligent HUnter Agent for Concept Understanding and LeXical ChaIning)

L'algoritmo Tucuxi

- Sviluppato dal DBGROUP dell'università degli studi di Modena e Reggio Emilia
- Agente per una ricerca più efficace e non supervisionata di sorgenti HTML
- Suddiviso in tre moduli indipendenti

Il primo si occupa della disambiguazione lessicale



re-implementazione del primo modulo

L'algoritmo Tucuxi

Presupposti:

- Sfruttare la proprietà di **Coesione** tra i significati associati ai nomi degli elementi degli schemi sorgenti

Metodi:

- Sfruttare le relazioni di Wordnet tra i synset associati ad ogni termine del testo per valutarne la *coesione*

Obiettivi:

- ✓ Attribuire al nome di ogni elemento il senso maggiormente *coeso* al contesto di riferimento

L'algoritmo Tucuxi

INPUT:

***Contesto lessicale*: lista contenente i nomi degli elementi degli schemi**

- **elementi appartenenti alla sola categoria sintattica dei nomi**
- **opportunamente selezionati da un *POS parser***

	Termini candidati	corretto assegnato
35	*course*	*d in a series of lessons…*
35	*education*	*rt knowledge*
35	*class*	*d in a series of lessons*
27	*information*	*red through study or experience*

OUTPUT:

Lista costituita dai nomi degli elementi e dal significato ritenuto corretto

L'algoritmo Tucuxi

Procedimento logico:

1) **Recupero delle relazioni tra i synset del contesto lessicale**

   Sinonimia

   Iponimia\Iperonimia

   Meronimia\Olonimia

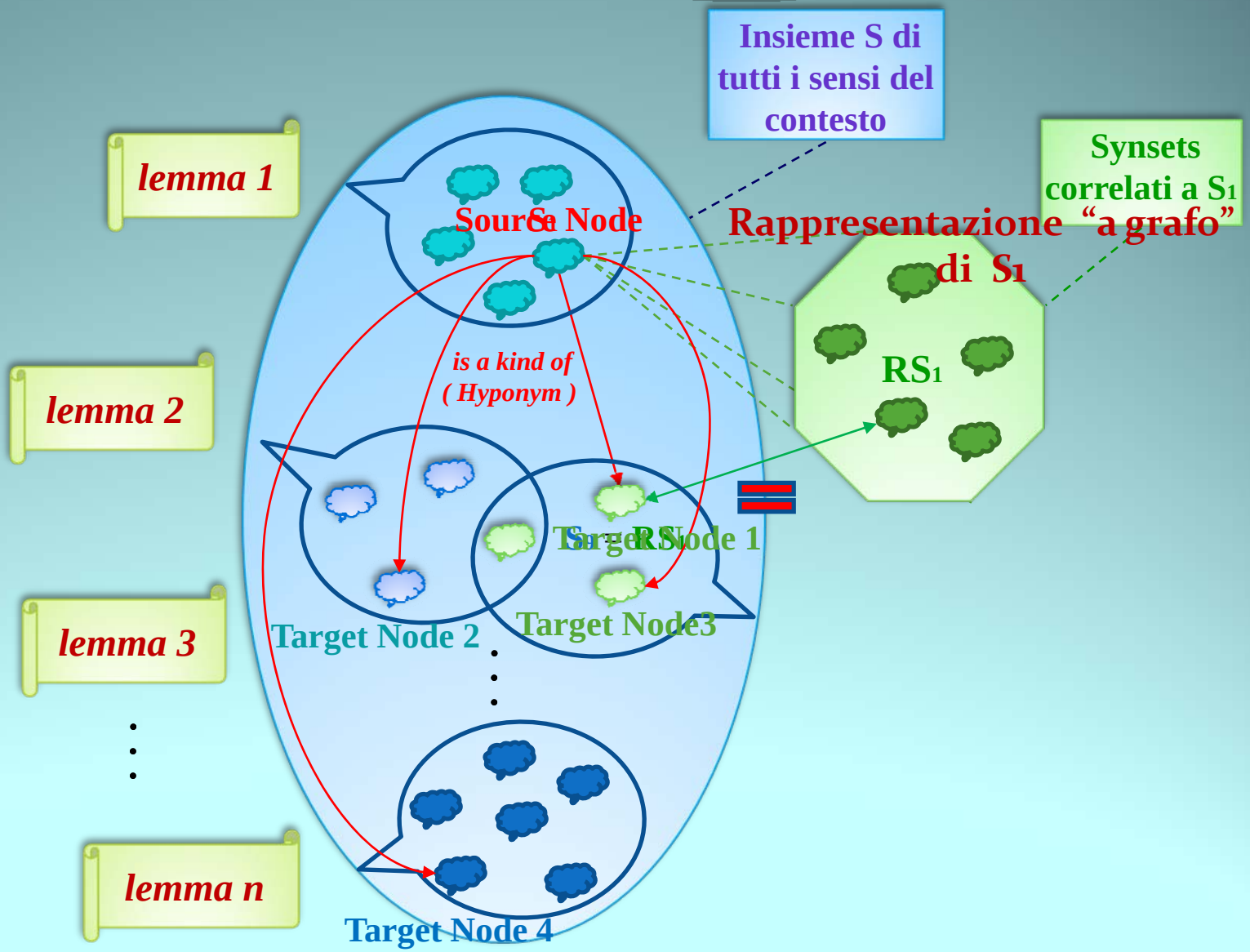
   - ✓ componente —*is a part of*→ oggetto
   - ✓ elemento —*is a part of*→ insieme
   - ✓ materiale —*is a part of*→ oggetto

   Domini in comune

2) **Attribuzione di un punteggio di coesione ad ogni synset**

3) **Estrazione dei termini disambiguati**

1) Recupero delle relazioni



Attribuzione del punteggio di coesione

Si attribuisce un punteggio di coesione ad ogni singolo synset:

- ✓ Ad ogni tipologia di relazione (arco) è attribuito un peso ***RW***

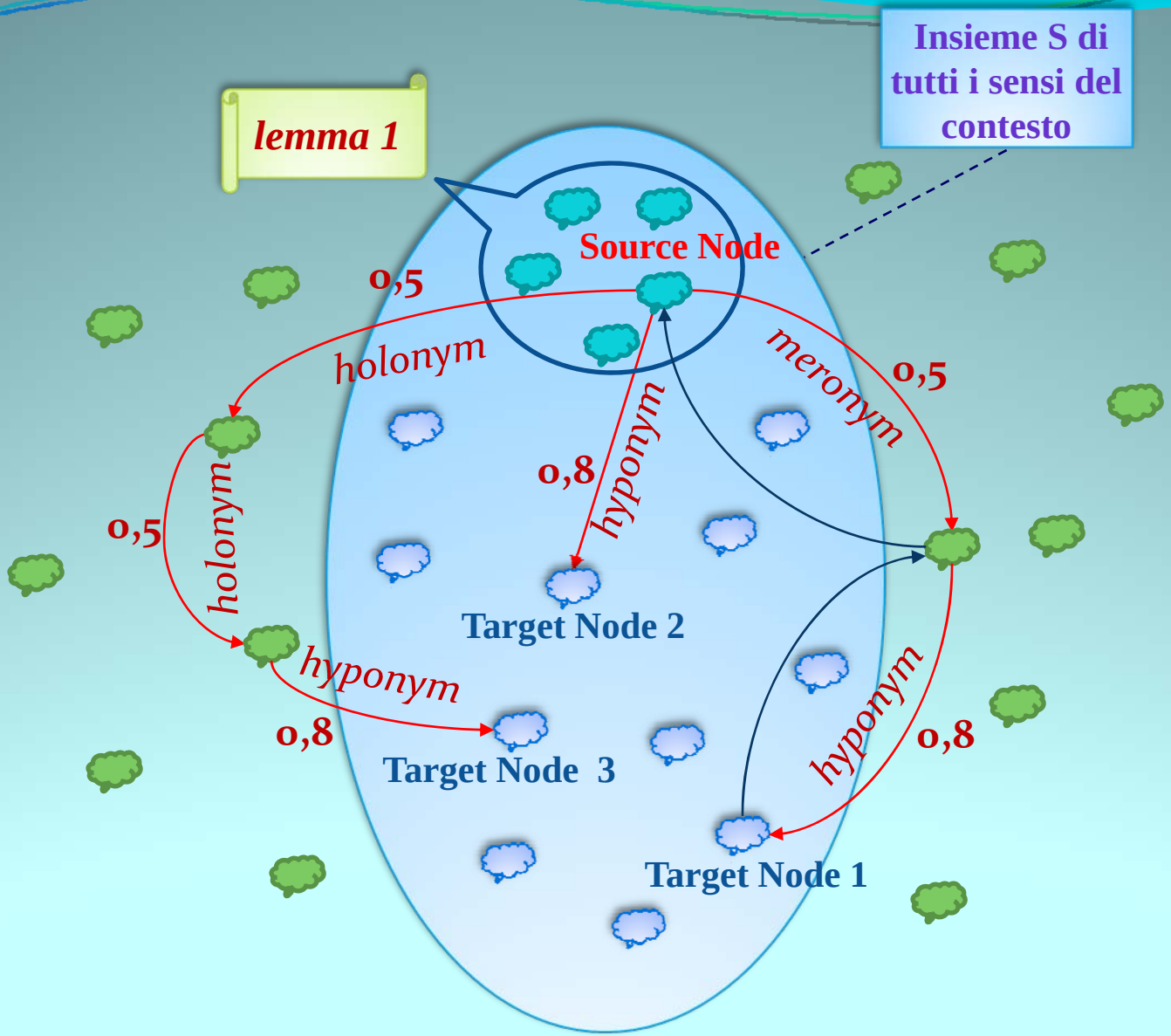
RELATIONSHIP TYPE	*RW*
SYNONYMY	1
HYPERONIMY/HYPONIMY	0,8
SUBST MERONYMY/HOLONYMY	0,5
MEMB MERONYMY/HOLONYMY	0,5
PART MERONYMY/HOLONYMY	0,5
DOMAIN IN COMMON	0.01

- ✓ Si considerano anche relazioni indirette:
  ad ogni cammino di lunghezza pari a ***k*** archi è attribuito un punteggio parziale pari a :

$$weight(sourceNode, path_i) = \frac{\sum_{j=0}^{k-1} RW(s_j, s_{j+1})}{lenght(path_i)}$$

- ✓ Ad ogni synset è assegnato un **punteggio di coesione** dato dalla somma di tutti i punteggi parziali

Attribuzione del punteggio di coesione



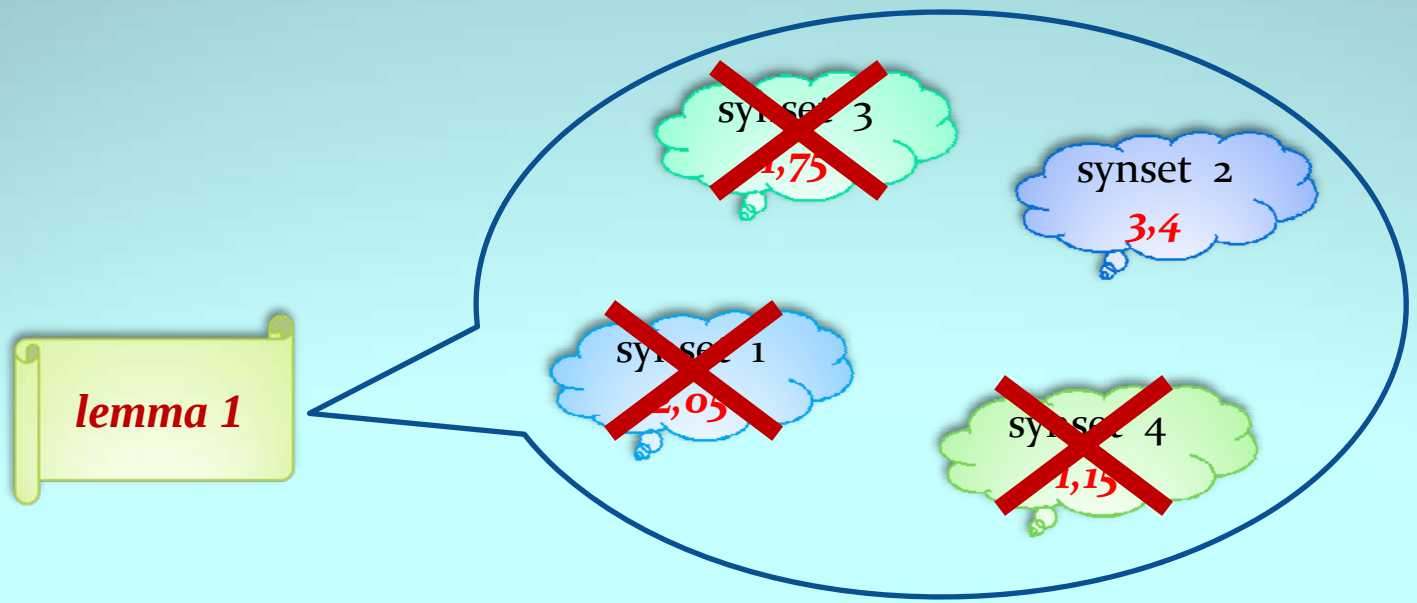
$weight(Source\ Node_{path}) = (0,8 + 0,8) / 2 = 0,8$

$w(Source\ Node) = (0,8 + 0,8 + 0,8) = 2,4$

Estrazione dei termini disambiguati

✓Si seleziona, tra i synset relativi ad uno stesso lemma, quello che presenta il **punteggio di coesione** più elevato.

✓Nel caso in cui un lemma non presenti alcuna relazione col contesto, non viene disambiguato (anche se *monosemico*)



Test effettuati

- Testato su 5 diverse sorgenti di dati
- Possibilità di configurare l'algoritmo selezionando:
  a) le tipologie di relazioni da prendere in considerazione
  b) i pesi da assegnare alle tipologie di relazioni considerate (da 0.01 a 1)
  c) il livello di profondità delle relazioni : il numero massimo di archi in un cammino che congiunge due synset (max = 3)

Obiettivi

- ✓ Valutare le prestazioni dell'algoritmo scegliendo differenti configurazioni (valutare quanto e in che modo incidano le tipologie di relazioni)
- ✓ Trovare una configurazione che massimizzi le prestazioni:

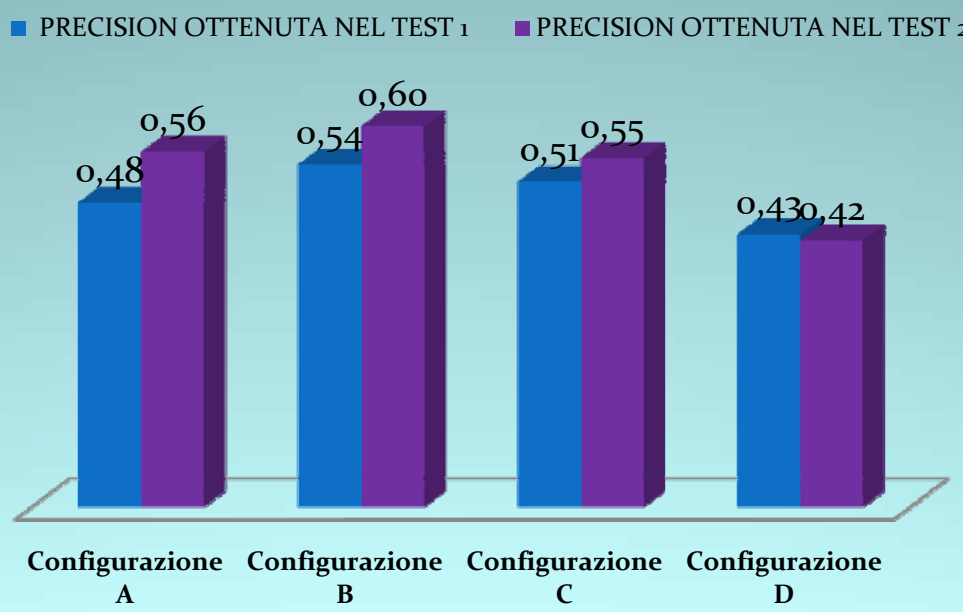
$$Precision = \frac{Numero\ di\ annotazioni\ corrette}{Numero\ di\ annotazioni\ effettuate}$$

$$Recall = \frac{Numero\ di\ annotazioni\ corrette}{Numero\ di\ annotazioni\ attese}$$

Risultati ottenuti

Test 1 : lunghezza massima cammini = 2 archi

Test 2 : lunghezza massima cammini = 3 archi



Osservazione

✓ Considerare una lunghezza massima per i cammini pari a 3 archi porta, generalmente, ad un aumento delle prestazioni

Risultati ottenuti

✓ La configurazione che massimizza la precision e la recall è quella che sfrutta esclusivamente le **relazioni semantiche**

RELATIONSHIP TYPE	WEIGHT
SYNONYMY	NC
HYPERONIMY/HYPONIMY	0,8
SUBST. MERONYMY/HOLONYMY	0,5
MEMB. MERONYMY/HOLONYMY	0,5
PART MERONYMY/HOLONYMY	0,5
DOMAIN IN COMMON	NC

NC = non considerata

Risultati ottenuti

Precision e Recall relativi al Test 2

	Configurazione A	Configurazione B	Configurazione C	Configurazione D
Precision	56%	60%	55%	43%
Recall	43%	44%	43%	32%

Osservazione

- ✓ Valori relativamente bassi di **recall** dovuti a:
  a) numero elevato termini composti
  b) uso di acronimi e abbreviativi

Conclusioni

- L'algoritmo proposto è in grado di disambiguare il 70 % dei termini con una accuratezza del **60%**
- Può fornire un utile supporto agli algoritmi di ALA
- Può essere applicato a sorgenti di testo
- Le tipologie di relazioni considerate sono fondamentali per l'esito del processo di annotazione
- Non è, allo stato attuale, ancora possibile sviluppare un algoritmo che ottenga il 100% di **precision** e di **recall**

Sviluppi futuri

- ✓**Nuovo meccanismo di attribuzione del punteggio**
- ✓**Considerare altri tipi di relazioni *derivati***
- ✓ **Utilizzare contemporaneamente più risorse lessicali**
- ✓ **Disambiguare efficacemente i termini composti: "Semi-automatic compound nouns annotation for data integration systems" S.Bergamaschi, S.Sorrentino, 2009**
- ✓**Gli incoraggianti risultati ottenuti dagli approcci CWSD che sfruttano una LKB potrebbero determinare in futuro un incremento delle prestazioni**

Per una analisi più approfondita la tesi è disponibile sul sito

www.dbgroup.unimo.it

Grazie per l'attenzione