Appunti Di Caro: Terza parte TLN

Federico Torrielli

Giugno 2022

Intro

- Per ogni domanda che un interlocutore fa è necessaria una specifica analisi semantica, con delle informazioni altrettanto specifiche (frequenze, contesto, anafora, glossa, meronimi, parole, punteggiatura, sinonimi, named entities... etc)
- La maggior parte degli sforzi in NLP è dedicata all'**estrazione automatica** di queste informazioni dai testi

Granularity

Possiamo individuare diversi tipi di granularity di "pezzi" utilizzati in NLP ed associare loro tanti task. Vediamone alcuni:

- **Word**: Word Sense Disambiguation, Word Sense Induction
- **Chunk**: Multiword Expressions
- **Sentence**: Question Answering
- **Discourse**: Chatbot
- **Document**: Summarization, Segmentation...
- **Document(s)**: Topic Modeling

WSD vs. WSI

WSD: problemi

- **Specificità**: "to play" ha molti sensi, anche abbastanza uguali, ma li specifichiamo perché ci potrebbero servire per capire il senso nel contesto
- **Copertura**: approssimare o andare nello specifico del senso?
- **Soggettività**: esiste della sogg. che è difficile da incapsulare all'interno di un dizionario, poiché statico (interpretazioni personali di un senso)

WSI

WSI è il task di **identificazione automatica dei sensi** di una parola. Produce in output un **clustering** di contesti nel quale le target word occorrono.

Word Sense Induction contro Word Sense Disambiguation

WSD	WSI
Possiedo un'inventory di sensi	Non ho inventory
Human-based	Data-based
Grammar-based	Usage-based
Evaluation semplice	Evaluation Complicata

WSI: Evaluation con pseudo-word

(probabile domanda esame)

Inganno il sistema:

- **Merging**: concateno randomicamente le word (sostituisco ad esempio la parola banana nel corpus con la parola banana-sedia, e la parola sedia con banana-sedia)
- **Clustering**: ora chiedo al sistema di fare WSI per identificare i cluster sui due sensi possibili della parola concatenata
- **Cluster-to-class evaluation**: ora chiedo al sistema "separamele nel modo corretto" dal cluster ottenuto. Il sistema dovrà darmi banana da banana-sedia quando si parla di cibo e sedia da banana-sedia quando ci si dovrà sedere.

Lexical Semantics

- Risorse come **dizionari elettronici**: Wordnet, babelnet (lessema, definizione, relazione paradigmatica-sintagmatica)
- Risorse **linguistico-cognitive**: *Property norms* (studi delle scienze cognitive [McRae] sulle parole molto associate ad altre, conoscenza immediata): una property norm è una proprietà che viene immediatamente in mente quando parliamo di un oggetto ($banana \implies gialla$).
- **Common-sense knowledge**: *ConceptNet* (contiene molte informazioni di natura sintagmatiche)
- **Visual Attributes**: concetti visibili all'occhio nudo, resi attraverso immagini (*ImageNet*)
- **Word Embeddings**: associare un vettore numerico alle parole
- **Corpus Managers**: tool per organizzare i corpus, gestire materiale lessicale (tipo testi) e hanno meccanismi di accesso facilitati o meccanismi di creazione di corpora filtrati, mirati, dedicati (*SketchEngine*)

Definizioni

Come scrivere una definizione

- Ci va sempre un riferimento iperonimico per contestualizzare un'oggetto che si sta definendo: *tavolo* è un oggetto (not-living-entity...), restringendo il campo semantico. Questa caratteristica ha un nome: il **genus**. Questo rappresenta un "livello medio", non generico e non specifico, che più o meno tutti comprendiamo quando ci riferiamo al concetto: la pera è *un frutto*.
- Le definizioni utilizzano esempi, soprattutto per concetti particolarmente complessi.
- La definizione può utilizzare relazioni di tipo part-of (*la gamba del tavolo*)

Genus-Differentia

- **Genus**: iperonimo più comune che si pensa che l'interlocutore conosca.
- **Differentia**: tutto quello che differenzia un oggetto da un altro.

Come valutare una definizione

- Quando non viene compresa, non è buona
- Sulla qualità del genus, se presente (se non presente, male).
- Se è presente una **circularity** (spiegare il concetto con il concetto stesso), allora è una pessima definizione.

Significato del significato

Teorie su meaning

- Basate su relazioni, esiste un legame tra costruzione e significato
- Utilizzo del concetto di inferenza

Triangolo semiotico

Modello del significato con tre angoli: rappresentazione (il termine), concetto (interpretazione) e il referente (l'esempio che i due rappresentano).

- La **rappresentazione** è il simbolo del concetto
- Il **referente** è un istanza concreta del concetto
- Il **concetto** è l'interpretazione mentale del concetto stesso
- Tutti gli sforzi in NLP partono dalla *rappresentazione* (non certo dalle immagini mentali, e neanche dai referenti) e vanno verso il *concetto* oppure da *rappresentazione* a *referente*.

Multilingual word meaning

Progetti che si basano sulla differenza tra le lingue per supportare l'analisi e migliorarla con relazioni tra le lingue. Il materiale lessicale di alcune lingue può

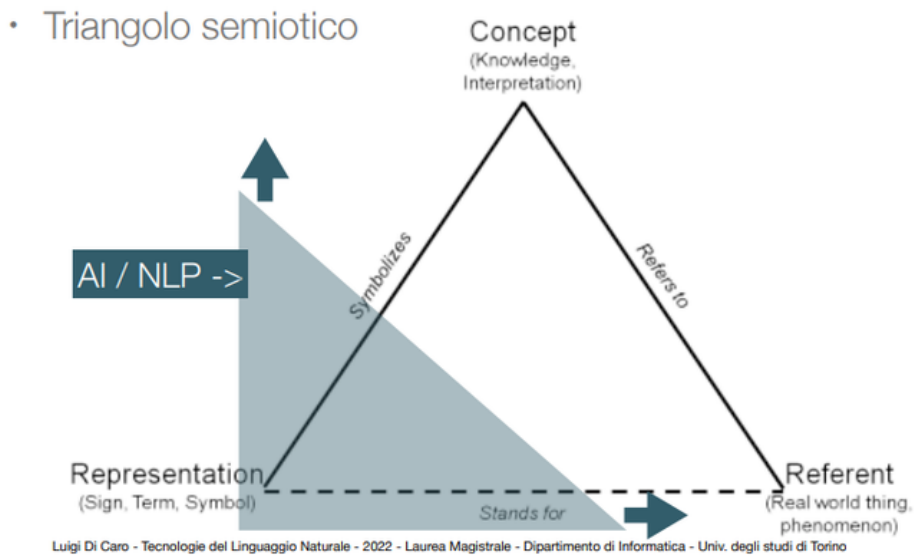

Figure 1: Il triangolo semiotico

essere utile per disambiguare su altre (molte lingue come il tedesco contengono delle parole sulle sensazioni alle volte inesistenti in altro, oppure come in culture sviluppatisi sulle coste ci siano termini molto più precisi sulla pesca piuttosto di lingue che si sono generate nell'interno dei paesi, lontano dal mare).

La costruzione del significato

Teoria ed implementazione di Pustejovski

Si basa su quattro strutture. Secondo lui, per definire la costruzione del significato servono queste quattro strutture informative:

- Struttura **ARGOMENTALE**: legame che c'è con la sintassi
- Struttura **EVENTUALE**: comportamento della frase nelle sue relazioni (transizioni, processi) legate ad eventi (anche temporali)
  - **Stato**: evento fermo nel tempo: *Mary is sick.*
  - **Processo**: evento che si muove nel tempo, un processo: *Mary walked.*
  - **Transizione**: evento che si muove nel tempo e di cui conosciamo una fine: *Mary closed the door.*
- Struttura **QUALIA**: struttura che associa specifiche proprietà ai concetti.
  - Ruolo **Constitutivo**: materiali, cose, peso, parti (*relazioni meron-imiche*)...
  - Ruolo **Telico**: funzione dell'oggetto, che cosa serve, l'obiettivo...
  - Ruolo d'**Agente**: da cosa nasce, perché, quale sarebbe la causa?
  - Ruolo **Formale**: tutte quelle proprietà che caratterizzano il concetto e

lo differenziano dagli altri (colore, forma, dimensionalità, posizione...)
- Struttura **EREDITALE**: parte tassonomica dell'informazione (la struttura gerarchica all'interno della frase)

Cosa serve generare questa teoria?

- Egli dimostra che nel momento in cui analizzo una frase e per ogni simbolo codifico tutte queste informazioni come spiegato, si può costruire un modello di inferenze di ragionamento che esplicita tutta la semantica possibilmente utile
- **PROBLEMA**: teoria difficilmente applicabile a causa dell'ambiguità delle frasi

Teoria ed implementazione di Hanks

Il significato di un'intera frase (a livello composizionale, della frase, non del singolo oggetto) nasce dal verbo. Esso è la radice del significato e tutto parte da esso. Soprattutto dalla sua **valenza** (la transitività dei verbi, per farla breve).

> Io mangio $\Longrightarrow$ Valenza 1 | Io mangio la mela $\Longrightarrow$ Valenza 2 | Io mangio la mela al mare $\Longrightarrow$ Valenza 3

Dati gli **argomenti** di un verbo possiamo specificarli con l'utilizzo degli **slot**, ciò che riempie lo slot viene detto **filler**, ed ogni filler ha associato dei **tipi semantici**, che creano una combinazione che corrisponde ad un **significato**:

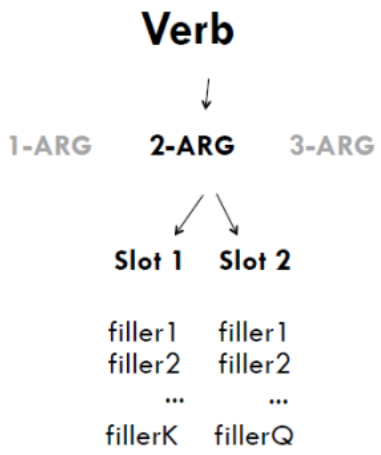

Figure 2: Combinazione di Hanks

Text Mining ed applicazioni

- Nasce dalla comunità del **data mining**, dalla basket data analysis, etc...

- Uso di tecniche statistiche applicate al testo come se fosse un dato come un altro: era della **distribution semantics**.

Approccio Top Down vs. Bottom-up

- **Top-Down**: vecchio approccio, guidato dalla grammatica, da regole codificate ad-hoc all'interno del sistema, utilizzando formalismi logici di tipo rule-based (linguaggi come Prolog, CLIPS, ... da regole verso dati)
- **Bottom-up**: con l'avvento del data mining, dal testo cerchiamo di fare inferenze semantico-statistiche verso l'alto (**da dati verso regole**).

Vector Space Model

- Rappresentazione della conoscenza testuale attraverso numeri
- Le parole sono **token**, ovvero sequenze di caratteri contigui, ovvero le parole perdono il senso intrinseco ma diventano solo elementi informativi
- Costruzione di un dizionario, un set senza duplicati, orientato su un asse con ogni parola associata ad un identificatore univoco. Questo può essere utilizzato per costruire una **matrice numerica** (corpus intero).

  Oggi ancora si usa questa tecnica, cambia solo come vengono costruiti i vettori. Come mai?

- Grazie a questa rappresentazione possiamo usare operazioni matematiche come la **cosine similarity**:

$$sim = \cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

Frequenza ed informazioni

- **Frequenza** (term frequency): il numero di occorrenze di un token diviso il numero di token (normalizzato)
- **Inverse Document Frequency**: $\log(\frac{nd}{ndt})$ Dove nd è il numero di documenti dove compare il termine, ndt è il numero di documenti totale. Se $nd == ndt$ allora la parola compare in tutti i documenti, e quindi il $\log(1) = 0$. Ci serve per normalizzare rispetto al suo valore entropico, ovvero rispetto a quante volte effettivamente compare nel documenti.

Co-occorrenza

Data una matrice diagonale come modello informativo, nelle righe e colonne ho sempre la stessa informazione (ovvero le parole, le stesse, sia su un asse che sull'altro). Nelle celle vado ad inserire valori di co-occorrenza: ogni volta che la parola i viene utilizzata nella stessa frase (contesto, documento), allora vado ad incrementarne il valore nella matrice di co-occorrenza. Avrò valori alti per parole che spesso co-occorrono insieme spesso.

La matrice serve per altri tipi di analisi semantica (derivati dal text mining) che ci dice che **se due parole hanno alta co-occorrenza, compaiono in contesti simili** e condividono un po' di semantica. Potrò calcolare degli score di similarità proprio basati su questa matrice.

Document Clustering

Meccanismo per raggruppare per similarità diversi documenti e rappresentarli nelle sue dimensioni su un piano. Apprendimento non-supervisionato.

Categorizzazione/Classificazione

- Sono sinonimi in NLP
- Sempre un raggruppamento ma i cluster sono già pre-fatti. Tipico apprendimento supervisionato.
- Come associare documenti di testi ad una tassonomia? Ad ogni nodo, con la sua etichetta, viene associato un vettore numerico e vengono fatte due cose:
  - **Inizializzazione**: creiamo un vettore dove abbiamo tutti 0, tranne nella dimensione che rappresenta il concetto stesso (è una matrice quadrata). Se abbiamo 5 concetti allora abbiamo una matrice 5x5 con 5 concetti e 5 features.
  - **Propagazione**: essendo che i nodi hanno un intorno topologico, propaghiamo i suoi valori verso altre dimensioni vicine (dall'alto verso il basso, dal basso verso l'alto). Viene utilizzato un fattore alpha per decidere quanto trasferire da una dimensione all'altra. (esempio: Africa-mondo si trasferiscono nella gerarchia solo alpha-percentuale informazioni).

Document Segmentation

- Input: un documento, testo
- Output: **linee di taglio**, dove si evidenzia tipicamente un cambio di argomento nel testo.
  Uno dei più famosi è il **text-tiling**

Text tiling

- Su asse x abbiamo il numero della sentence, sequenzialità dell'intero documento
- Su asse y abbiamo i termini utilizzati: in ordine di apparizione
- Dentro la matrice quindi abbiamo dei numeri, che esprimono la frequenza della parola nella sentence, vediamo che sono evidenti dei clusters.

All'inizio dell'algoritmo, le linee di taglio sono messe random e dopo un po' di cicli (tipo k-means) andranno a riposizionarsi nella parte corretta. Le frasi potranno essere separate a due-due o tre-tre... e andiamo effettivamente a spostare le nostre colonne dove abbiamo bassa coesione tra frasi (coesione intra-gruppo).

TODO: x-means per trovare il numero dei segmenti

Document summarization

- Metodi **estrattivi**: estrarre delle frasi importanti, non si genera nessun testo
- Metodi **astrattivi**: generare del testo semplificato da un testo originale (natural language understanding + generation)
- Valutazione: ROUGE

Orienteering browsing

Task creato per aiutare le persone a orientarsi in una sorgente di informazione.

Semantica Documentale e Visualizzazione

Analisi semantica a livello più alto possibile: siamo a livello di documenti interi, collezioni di materiale (non un singolo termine, paragrafo o documento, ma una collezione di essi).

Topic Modelling

Tecnica che meglio traduce il concetto di semantica documentale.
Quando ho una collezione di documenti o:

- Faccio una search (information retrieval)
- Modello un topic (argomento) in maniera automatica

Def: modello statistico/probabilistico che si basa sull'utilizzo del linguaggio per trovare l'argomento di un documento. La radice informativa più importante sono le **co-occorrenze**. Nei modelli di t.m. vengono quindi utilizzate queste ultime.

Per una macchina un **topic** è semplicemente un set di termini messi assieme, dove ogni termine ha un certo peso probabilistico.
Dato in input una collezione di testi, vogliamo un insieme di insiemi di termini pesati (diversi topic).

Questo è un modello **non supervisionato**.

Problemi del topic modelling

- **Bassa interpretabilità**: basandosi sulle co-occorrenze, termini tendono a comunicare un argomento a se stante, ma a volte esse non hanno tanto senso assieme (statisticamente rilevanti ma di difficile interpretazione)
- **Alcuni dati inutili estratti**: non sempre quello che estraiamo non è utile (function words, eventi anomali, numeri e combinazioni lessicali che *sporcano* i topic)
- **Parametri statici**: quasi sempre gli algoritmi di t.m. necessitano come parametro il numero di topic da tirare fuori *staticamente*, è ovviamente scomodo: per questo proviamo e riproviamo, analizzando

Latent Semantic Analysis (LSA)

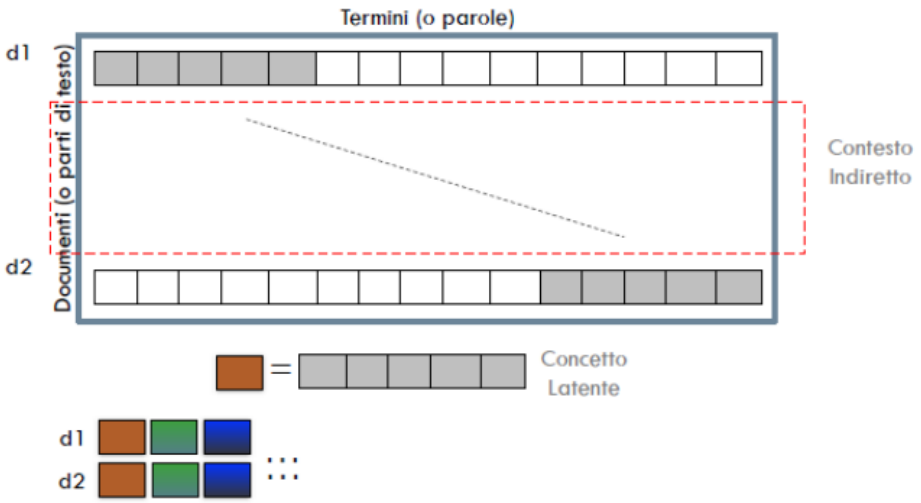

Figure 3: LSA

- **Input**: Raw text data
- **Processo**: Document-term matrix $\Longrightarrow$ SVD
- **Output**: Topic-encoded data
- Basata sulla **Singular value decomposition**: fattorizzazione che data una matrice numerica $n \cdot m$, approssima la matrice attraverso una combinazione di tre matrici ($X = U\Sigma V^T$, dove Σ è la matrice diagonale e U, V sono le matrici ortogonali), il cui prodotto approssima la matrice di partenza: la prima matrice da l'informazione sulle righe, la terza rappresenta le colonne e la seconda, che è diagonale, rappresenta le combinazioni lineari delle righe e delle colonne in input della matrice originale. Analizziamo la ridondanza della matrice originale, provando a discernerla attraverso la

sua combinazione lineare delle righe e delle colonne di partenza. Possiamo trattare la prima e l'ultima matrice come uno **spazio semantico** che possiamo comparare con metodi come la *cosine similarity*.

- Analizza concetti **latenti** (*nascosti*) e concetti **indiretti**. Termini che co-occorrono spesso vengono identificati da questa fattorizzazione matriciale e fusi in uno spazio matriciale ridotto.
- La matrice diagonale (seconda) da l'informazione su quanta entropia siamo riusciti a dare della singola dimensione. Le due matrici esterne sono documenti x termini (un termine = freq. del termine nei documenti).

Problemi

- Modello che **non generalizza** su documenti che non ha mai visto
- Presenza di **valori negativi** dopo *SVD* che non sono facilmente interpretabili:

Soluzione 1: Latent Dirichlet Allocation

- Altro meccanismo, successore della probabilistic LSA, basato su una distribuzione di probabilità **Dirichlet**, sfruttando la statistica Bayesiana.
- Basato sull'assunto che un documento è un mix di topics e ogni parola si porta dietro una distribuzione di probabilità che compaia un certo topic.

Soluzione 2: Dynamic Topic Modelling

- Si divide in finestre temporali il corpus e poi si guarda come il topic si modifica nel tempo.

Text Visualisation

- Approcci grafici:
  - **Parallel Coordinates**
  - **RadViz**: cerchio con x coordinate, che rappresentano le dimensioni, pensate come magneti che attraggono i diversi valori sullo spazio
  - **Heat map**
  - **Correlation Circle**: si proiettano le correlazioni tra topic, mettendoli in una circonferenza

Distributional Semantics

NLP + Text Mining + Statistica

- Nuovi nomi, nuove forme ad approcci vecchi con elementi in più: aspetti che arrivano dal NLP simbolico (legato alla grammatica e allo studio della linguistica): nuova **letteratura** che prende i contributi del text mining e li ha uniti alla letteratura sul linguaggio naturale (linguistica)
- Basata sulla **ipotesi distribuzionale**: oggetti lingustici con distribuzioni similari hanno anche significati similari.

Citazioni storiche

- *Harris* e *Firth*, anni '50: parlano di **distributional hypothesis** e **distributional analysis**: "*le parole che occorrono negli stessi contesti tendono ad avere significati simili*" e "*una parola è caratterizzata dalla sua compagnia*"
- *Furnas*, anni '80: "*l'uso congiunto delle parole serve per specificare meglio l'oggetto del discorso*" $\Longrightarrow$ Se voglio cercare su Amazon una cuccia per gatti non basta mettere *gatto* ma dovrò mettere *cuccia per gatti*.
- *Deerwester*, anni '90: "*Esiste una struttura semantica nascosta e latente nei dati che è parzialmente oscurata dalla randomicità della scelta della parola rispetto a come viene prelevata*"
- *Blei*, 2003: "*Il tema del documento (ciò che mi accingo a scrivere) influenza probabilisticamente la scelta delle parole*"
- *Turney*, 2003: "*Coppie di parole che co-occorrono in pattern similari tendono ad avere le stesse relazioni semantiche*"

Usare matrici

- Comprensione umana basata su un'**approssimazione**
  - L'ordine delle parole è importante ma non così tanto come si pensa: con le matrici perdiamo l'ordine.
  - Un 20% della semantica di un testo inglese è dato dall'ordine delle parole (se tolgo l'ordine approssimo la semantica, perdo dei pezzi ma non è troppo problematico)
- **Significato** espresso come una **regione geometrica**
  - Rappresentazione che facilita la condivisione della conoscenza: avere uno spazio di quality features è facile da analizzare
  - **Prototipo**: *centroide*

Pre-Processing

- **Normalization**
  - Tokens to types

  - Basic english
  - Stemming
  - Lemmatization
- **De-normalization** (*semantica*)
  - Named entities
  - Semantic roles
  - Word Sense

Uso delle matrici nella D.S. (Peter Turney)

Esistono 3 macro-categorie di matrici nella distributional semantics, secondo Turney:

- **Term-Document matrix**: **documenti / termini** sulle due dimensioni (document similarity, document clustering, document classification, document segmentation, partial question answering)
- **Term-Context matrix**: **contesti / termini** sulle due dimensioni (word similarity, word clustering, word classification, WSD, spelling correction, semantic role labelling, NER...)
- **Pair-Pattern matrix**: $pattern/wordX : wordY$ (pair) sulle due dimensioni.
  - Si analizza la distribuzione semantica di una serie di parole (nel caso pair sono due)
  - **Pattern** è il pattern lessico-sintattico tra le due parole, cioè tutto ciò che lega una parola all'altra (*X is solved by Y, X solves Y, X is resolved by Y...*)
  - Questa informazione serve per:
    * Relational similarity
    * Pattern similarity
    * Relational clustering
    * Relational classification
    * Relational search: "*lista tutte le X tale che X causa il cancro*"
    * ...

Ruolo della similarity

- La similarità ha un ruolo fondamentale anche nella nostra sopravvivenza

- Similarità in NLP:
  - **Semantica** (sinonimi e quasi-sinonimi)
  - **Semantic Relatedness**: riguarda concetti che condividono proprietà. Il significato di questa quantità è generico.
  - **Attribuzionale**: simile alla *SR*.
  - **Tassonomica**: condivisione di iperonimi *etc etc.*.
  - **Relazionale**
  - **Associazione semantica** (culla e bambino)

Ontology Learning & Open Information Extraction

Ontology Learning

- Processo **reverse engeneered** che si tratta dell'estrazione dei concetti dalle ontologie esistenti e codificate dagli umani.
- La conoscenza di dominio non è utilizzata completamente: se devo creare una tassonomia degli alimenti, se i miei task sono di classificazione dei prodotti in un supermercato non mi interessa di andare nel dettaglio dei singoli ingredienti nei prodotti $\implies$ serve un livello che dipende dal task! La **domain knowledge** dipende perciò dal dominio ma non comprende tutta la conoscenza *intera* del dominio!

Differenze con altri tipi di task basati sulle ontologie

- **Ontology Population**: fa parte dell'ontology learning ma serve per popolare un'ontologia pre-esistente. Dati dei nuovi documenti riempio uno scheletro di ontologia (tipo riempire un frame).
- **Ontology-based Annotation**: l'attenzione è sui *documenti* $\implies$ voglio una base documentale arricchita, evidenziando simbolicamente parti del testo che riflette i concetti essenziali del testo.
- **Ontology enrichment**: cerco di arricchire un'ontologia dal punto di vista di concetti e relazioni mancanti.

Livelli di profondità delle ontologie

- Livello dei **documenti**
- Livello della **terminologia**
- Livello del **glossario**: il glossario, in più del dizionario, possiede una glossa (definizioni + esempi)

- Livello del **thesaurus**: tipo **WordNet**, abbiamo le prime relazioni
- Livello **tassonomia**: relazioni tassonomiche
- Livello **ontologia**: livello di senso comune, soggettività...
- Livello **logico**: regole simbolico-formali dove si specificano delle regole di inferenza

Ontology Learning: cosa vogliamo estrarre

- **Term Extraction**: trovare nomi dei concetti e delle relazioni (termini ricorrenti, tag clouds)
- **Synonym Extraction**
- **Concept Extraction**:
  - **Intensioni**: concetto $\Longrightarrow$ glossa
  - **Estensioni**: glossa $\Longrightarrow$ concetto
- **Concept Hierarchies Induction**
- **Relation Extraction**: estraggo relazioni semantiche generiche
- **Population**: trovare delle informazioni con information extraction, pattern learning, reti neurali e legarla all'ontologia

Ontology Learning: come vogliamo estrarre (NLP Classico)

- **POS-tags**
- **Preprocessing**
- **Analisi Sintattica**
- **Alberi sintattici**
- **Similarità con vector space model**
- **Risorse Linguistiche pre-esistenti**

FCA: Formal Context Analysis

- Metodo che proviene dall'ambito matematico: utilizzato per fare ontology learning
- Serve per estrarre delle tassonomie da delle informazioni strutturato: esso parte dalla matrice concetto-feature e costruisce il formal context, da cui si può ricavare il lattice.
- **Figure**:
  1. (a) **matrice concetto-feature** (dati in input)

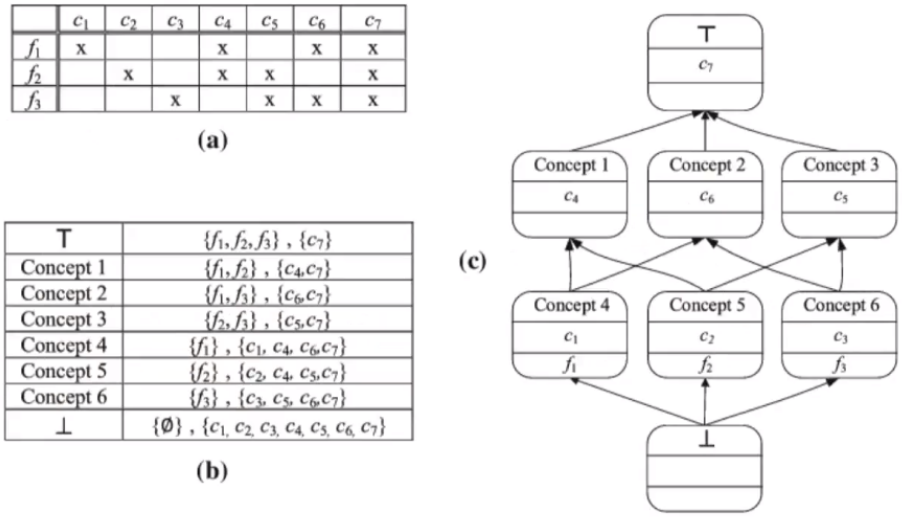

Figure 4: FCA: tutti i passaggi

2. (b) **formal context**: lista dei concetti che condividono una certa serie di features, in tutte le loro combinazioni, queste vanno a formare un lattice. $\top$ è la testa, ovvero dove ho tutte le feature e vado verso $\bot$, che non ne contiene alcuna.
3. (c) **lattice**: formal context sotto forma di lattice

Open Information Extraction

- Chiamato anche **shallow parser** oppure **shallow ontology learning** $\implies$ informazione semi-strutturata
- Si concentra sulla granularità delle frasi e tira fuori **relational phrases**, *triplette* con argomenti (propbank ad esempio $\implies$ *arg1 - VP - arg2*)
- Nato per rendere scalabili i sistemi di information extraction, perché se io ho un meccanismo di question answering che mi strutturi un testo, riesco ad interrogare una base documentale di qualsiasi tipo per avere un'informazione del tipo soggetto-verbo-oggetto.
- Si può utilizzare POS o dependency parsing per estrarre queste informazioni (se non usiamo reti neurali eh...)
- **Problematiche**: esistono troppi sistemi diversi, rendendo così difficile da valutare l'estrazione! L'OIE non ha un meccanismo standard per valutare la bontà delle triplette.

	KB	NLP
Positivi	Informazione codificata e autoesplicativa	Approcci neurali recenti raggiungono lo stato dell'arte
Negativi	Non tutta la conoscenza ha la forma di un grafo	Servono tanti dati e di buona qualità, possibilmente interpretabili dai modelli

Knowledge Graph

- Grafi dove i legami sono molto importanti
- **Altamente scalabili**: usati in grafi come nei social network graphs
- **Operazioni poco costose**: mettendoli a confronto con dati SQL vediamo come sia molto più semplice collegare i dati
- **Facili trasformazioni dei dati**

Modello

Nodi + Relazioni

- Nodi: dipende dal tipo di ontologia specificata, spesso sono tipizzati
- Relazioni: dipende dal tipo di ontologia specificata, possono avere direzioni

KG vs NLP

- Alle volte non serve estrarre le relazioni con reti neurali, quando sono già presenti su KG
- In questi ultimi anni vengono coniugati gli sforzi nella ricerca di reti neurali e KG

Approcci Metodologici KG

- Guidati **dai dati**
- Guidati **dagli obiettivi** (tipo minimizzare il tempo, le risorse...)

Utilizzi KG

- Sense disambiguation
- Question answering
- Semantic search
- Recommender Systems
- Knowledge graph completion
- Entity resolution

Machine Learning e NLP

Qua sono state saltate tutte le descrizioni dato che sapevo già tutto

- Grazie al paradigma **text as data** riusciamo a far passare: *features* $\Longrightarrow$ *model* $\Longrightarrow$ *evaluation*

Tecniche di ML-NLP

- Naive Bayes (antica)
- Support Vector Machines (antica)
- Hidden Markov Model (datato)
- Conditional Random Fields (datato)
- Neural Networks (*SUIIIIIIII*)

Neural Networks per NLP

- Recurrent Neural Networks e LSTM
- CNN
- Transformers (state-of-the-art)
- Autoencoders (state-of-the-art)

Problemi Aperti

- Overfitting
- Few-shot learning: training set contiene troppe poche info
- Domain-adaptation
- Interpretability
- Common-sense
- Costo computazionale
- Sviluppo su piccoli devices