# TLN - DI CARO

## 1) Lezione 2: Descrizione di un concetto

**Come descrivere un concetto?**

**Genus: Iperonimo più comune dell'oggetto che si pensa che l'interlocutore conosca al meglio**

**Ad esempio, per il concetto "Mela" salendo per il percorso iperonimico trovo "Frutto", "Cibo" (qualcosa di commestibile), "Cosa" (oggetto), ecc. Gli ultimi due sono molto restrittivi ("Cibo" e "Cosa") e si parla quindi di Middle Level, ossia una sorta di livello che più o meno tutti conosciamo.**

**Non sono elementi iper generici, neanche iper specifici, ma immaginando una sorta di gerarchia concettuale c'è una sorta di livello medio (Middle Level) che tutti utilizziamo e prendiamo come livello medio quando descriviamo qualsiasi cosa.**

**Di altre relazioni semantiche c'è ne sono tante altre.**

**Come definire la qualità di una definizione?**

**Se la mia definizione contiene un genus, allora contiene già un elemento cardine per la mia definizione perché ho qualcosa che in qualche modo restringe il campo semantico.**

**Uno dei problemi molto importanti è: Data una definizione, è difficile ricondursi al concetto di partenza.**

**Questo passaggio da definizione a concetto viene detto Onomasiologic Search, ed è esattamente l'opposto del dizionario.**

**Dizionari analogici: si rifanno all'onomasiologic search. Consistono in una concettualizzazione di un concetto che punta a tutti i suoi possibili termini, che è esattamente l'opposto del dizionario.**

**Tip of the tongue: E' un rompicapo che si basa sull'onomasiologic search. Esempio: "E' giallo, si mangia, ed ha la buccia cos'è? la banana, il limone, ecc.."**

**Genus differentia mechanism:**

**Abbiamo detto che il Genus è una restrizione semantica attraverso l'uso di un singolo termine che tipicamente è un iperonimo diretto o non, ma mai iper generico.**

**Una volta ristretto il campo semantico, tiro fuori degli elementi che discriminano altamente tutto ciò che ci può stare all'interno di un campo semantico rimanente, ossia le caratteristiche che differenziano quel concetto da tutti gli altri vicini.**

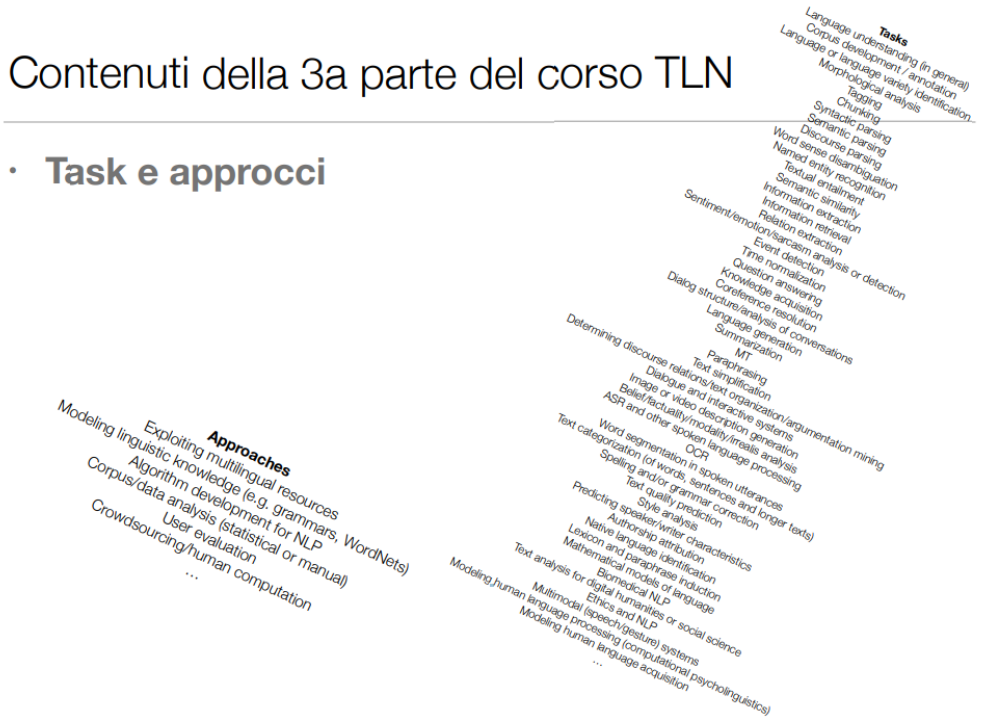
**Questo è il meccanismo classico per generare definizioni basato su Genus di elementi peculiari di quel concetto che lo differenzi dai vicini.**

**Circularity: Definizione di un termine tramite un termine che sto cercando di descrivere.**

**Ad esempio definisco X tramite XW e definisco W tramite X. Si crea una ricorsione circolare.**

## 2) Lezione 3: Teorie sul significato

### Di task di NLP ce ne sono tanti:




### Definizioni di base:

a) **Lessico:** insieme degli elementi linguistici a disposizione. Ad esempio, un vocabolario

b) **Sintassi:** regole per la costruzione di una frase.

c) **Semantica:** è l'interpretazione di una struttura lessico-sintattica. La semantica studia il significato delle parole

d) **Pragmatica:** È la capacità di mettere in relazione le parole che usiamo con il contesto comunicativo in cui ci troviamo nel momento in cui le utilizziamo. Ad esempio: l'espressione ti muovi come un elefante, per indicare che una persona si muove in maniera goffa

e) **Ambiguita':** Si può trovare sia nel lessico che nella sintassi. L'ambiguità è un livello che maschera il vero significato alle macchine. Ad esempio nella Polisemia/Omonimia.

f) **Comunicazione:** il linguaggio nasce come strumento per condividere significati presenti nella nostra mente. Serve quindi a comunicare.

g) **Convenzione:** la comunicazione è un concetto ampio, sia verbale che non verbale. Nel caso verbale si usa la convenzione di trasferire un contenuto semantico attraverso dei simboli (lettere).

h) **Granularità:** la struttura del linguaggio può essere più o meno profonda. Il contenuto semantico cambia se ci concentriamo su una parola, una frase o un testo.

i) **Soggettività:** tale livello dipende dalla persona, in base ad essa e agli altri elementi sottostanti può essere più o meno complesso.

j) **Cultura:** una parola (convenzione) cambia a seconda della cultura a cui si fa riferimento.

k) **Senso comune:** è legato alla cultura. È il significato che diamo alle convenzioni, in maniera condivisa all'interno della cerchia culturale, viene usato per esprimere concetti diversi.

l) **Esperienza personale:** dinamica che cambia il significato del significato

m) **Similarità:** è il meccanismo che ci permette di adeguarci a situazioni non previste e non note e ci permette di capire il significato di qualcosa di nuovo. Risulta fondamentale nel NLP perché i calcolatori devono avere a che fare con situazioni non note a priori. Il cervello lavora spesso con questo concetto (cosa abbiamo sentito dire, cosa abbiamo visto)

**Tramite queste definizioni abbiamo creato una ontologia di base, ossia concetti in cui il significato hanno un valore condiviso. Non ci interessa il significato singolo, ma quello condiviso tra i concetti. Un'ontologia è una rappresentazione formale, condivisa ed esplicita di una concettualizzazione di un dominio di interesse.**

## Teorie su "Word Meaning":

a) **Basate su primitive: per rappresentare il significato di una parola si sgretola in primitive universali che valgono per qualsiasi lingua. E'un modo arcaico di vedere la cosa.**

**Esempio: Parlare è legato a bocca, mano, ecc.**

**Esempio: per comprendere il significato di scrivania dobbiamo prima capire il concetto di tavolo e ancora prima il concetto di struttura piana con fondamenta. Il significato è dato dall'insieme di tali primitive.**

b) **Basata su relazioni: il significato che sta dietro una parola nasce dalla relazione con altre parole. Una parola di per sé non ha significato se non è impiegata in un contesto fatto di altre parole. Parliamo di una sorta di contestualizzazione. qui entra in gioco il concetto di Logic Forms e di inferenza: a determinate forme sintattiche posso associare poi un significato inferenziale, da questo deriva quest'altro.**

**Esempio: il telefono si relaziona in modo che gli venga attribuito significato, ossia il telefono si usa per comunicare piuttosto che per qualcos'altro.**

c) **Basate su composizioni: il significato si costruisce in maniera generativa e creativa nel momento in cui si utilizzano i concetti in forme strutturali. Si va a comporre significati attraverso composizioni lessico-sintattiche. Non solo una parola prende significato quando inserita in un contesto lessicale, ma la composizionalità stessa delle parole produce significati nuovi e le parole stesse prendono un significato individuale. In pratica è la composizione tra parole che dà il significato.**

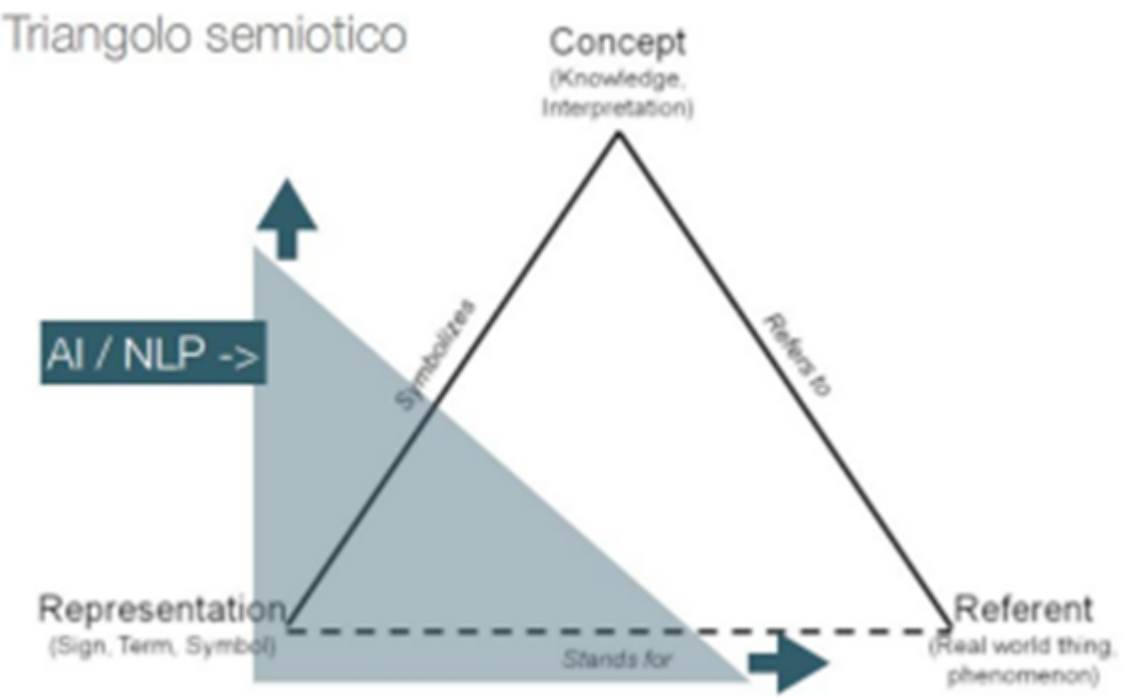
**Esempio: vino rosso. Rosso di per sé ha già un significato, tuttavia associato al vino il suo significato cambia.**

**Esempio: "Mi muovo come un elefante", attribuisco un modo goffo per esprimere un tipo di movimento buffo che si fa.**

## Triangolo Semiotico

**E'un modello utili per esprimere il significato.**

**Si tratta di una rappresentazione a triangolo tra tre concetti chiave di lettura di un concetto.**




**Si hanno 3 angoli:**

1) **Concept (knowledge o interpretation) : Immagine mentale dell'oggetto che si vuole descrivere;**
2) **Representation(Sign, Term, Symbol): simbolo usato per la rappresentazione. Ad esempio "gatto" viene utilizzato come simbolo per identificare l'oggetto gatto del mondo reale.**
3) **Referent (Real world thing phenomenon): Istanza concreta di quel concetto. Ad esempio "un gatto" oppure "tre gatti".**

**Partendo dallo spigolo del triangolo più a sinistra (Representation), la lettura del triangolo è: Il livello di rappresentazione simboleggia l'immagine mentale a cui fa riferimento uno o più referenti del mondo.**

**Tutti gli sforzi dell'AI e del NLP partono dalla convenzione => Partendo da un testo abbiamo solo simboli.**

**Si muove anche nella parte delle immagini perché non astrae solo gli oggetti dalle immagini e gli assegna un simbolo, ma va anche a capire il fenomeno nella sua naturalezza del mondo reale.**

**Ad esempio ho una detection di gatti e cerco di andare in questa direzione.**

## Teorie su "Multilingual Word Meaning":

**Lingue diverse creano problemi.**

**Ci sono sfide, ma anche opportunità: ad esempio l'analisi di altre lingue per migliorare la traduzione di altre.**

**Ad esempio in luoghi di mare ci sono più termini legati alla pesca rispetto a luoghi più distanti dal mare.**

**La lingua inglese è molto ambigua sui verbi di movimento. Se prendo il finlandese, su tale lingua ho verbi più specifici sul tipo di andamento di camminata che si può avere e mettendole in relazione riesco a disambiguare l'inglese grazie al materiale semantico del finlandese.**

**Lingue rare: Lingue pseudo morte di cui non esiste un wordnet e non si vuole disperdere il loro contenuto sintattico-semantico, per creare sistemi di NLP che funzionano anche su di esse.**

**Si hanno concetti diversi tra le lingue, ad esempio in italiano diciamo "spaghettata", "abbiocco", "boh" che esistono solo in tale lingua.**

**Granularitá: Analisi semantica a diversi livelli di granularitá, dove ogni livello ha i propri task. Possiamo citarli:**

a) **Word: complessità già elevata, word sense disambiguation**
b) **Chunk: multiword expression**
c) **Sentence: si utilizza il question answering**
d) **Discourse: chatbot, botta e risposta tra i partecipanti**
e) **Document: summarization, generare un estratto che contenga il più possibile della semantica del primo. Creazione di un riassunto**
f) **Collection of documents (topic): topic modelling, data una collezione di documenti, trovare i temi della collezione**

# 3) Lezione 5: Costruzione del significato e Text Mining

## Teoria di Pustejovsky e i qualia roles

**Per definire la costruzione del significato per far interfacciare lessico e sintassi-semantica, si hanno bisogno di 4 strutture:**

1) **Inheritance Structure: Nel momento in cui voglio costruire il significato, una parte dell'informazione da codificare è quella sulla struttura gerarchica. Ad esempio, definendo la struttura gerarchica del campo semantico "Mela" mi permette di lavorare anche su inferenze, perché sto parlando anche di concetti vicini a questa parola.**

2) **Event Structure:** sono tutte quelle informazioni che secondo Pustejovsky, bisogna codificare, relative a tutti gli eventi e a tutte le dinamiche temporali che possono essere interessate da quel concetto.

   Ad esempio: Transizioni, processi, stati attuali, fenomeni, ecc..

   Esempio: Il fatto che una mela possa marcire oppure che venga mangiata sono legate dagli eventi che possono manifestarsi.

3) **Argument Structure:** legame che c'è con la sintassi. Costruisco il significato attraverso delle frasi e di ogni parola mi serve la struttura gerarchica. Anche la grammatica (sintassi) mi aiuta a capire di cosa si sta parlando.
4) **Qualia Structure:** è una specifica struttura che associa proprietà ai concetti che vengono menzionati. In particolare si hanno 4 ruoli che sono di dettaglio su:
   a) **Constructive role:** informazione di struttura, essenziale (materiale, parti, peso, ..)
   b) **Telic role:** la funzione che serve, qual'e l'obiettivo
   c) **Agentive role:** da dove nasce questo fattore naturale o a chi è interessato a livello di agente. Esempio: Ho una lettera da consegnare, c'è il postino (quindi le persone coinvolte).
   d) **Formal role:** Tutte quelle proprietà che caratterizzano il concetto e lo distinguono da altri. Esempio: il colore della banana che viene spesso associato ad essa, diventa un ruolo formale e lo può essere anche la forma, la posizione, ecc.

**C'è una sovrapposizione tra la Formal role e gli altri ruoli perchè spesso se il concetto lo troviamo in Formal role, è possibile trovarlo anche in altri role perchè sono caratteristiche peculiari per quel concetto.**

**Pustejovsky crea una sorta di linguaggio formale logico dove lega tutte le variabili che arrivano da tutte queste strutture e riesce a modellare e formalizzare la semantica attribuibile a una frase.**

**Difetti:**

a) **I ruoli sono troppo specifici, non spiegati a un certo livello di dettaglio e crea ambiguità nel linguaggio.**
b) **Molto teorica e poco pratica, difficile da usare computazionalmente.**

## Teoria di Hanks

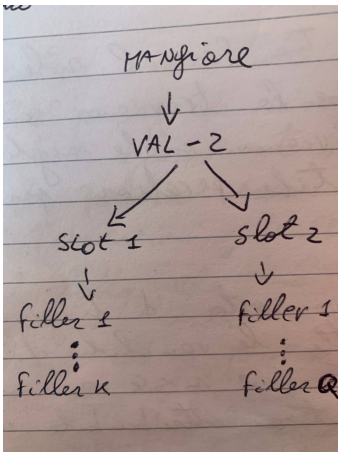
**Per Hanks, tutto parte dal verbo, dalla sua valenza.**

**La valenza è il numero di argomenti che prende in input (sarebbe la transitività).**

**Puó avere diversi livelli di transitività (valenza): Ad esempio “io mangio la mela” oppure “io mangio la mela al mare”. Nel primo caso ho valenza 2.**

**Secondo Hanks, la prima distinzione di significato avviene a livello di valenza.**

**Prendendo come esempio il verbo precedente, con valenza 2:**




a) **Gli slot sono il soggetto e complemento oggetto di tutte le cose possono essere prese in input a livello lessicale**
b) **I filler sono tutti gli elementi che possono riempire lo slot. Ad esempio tutto ció che può mangiare o puó essere mangiato.**

**Una volta analizzati tutti i filler di fianco al soggetto, cerchiamo di fare cluster types (semantic types).**

**I semantic types sono delle categorie che mettono insieme tutti questi filler. Ad esempio se siamo nei soggetti del verbo “mangiare”, ci possono essere persone, animali, o concetti astratti (“questa attività mi mangia tutto il tempo”).**

**Faccio una salto di clustering semantico anche per tutti gli slot.**

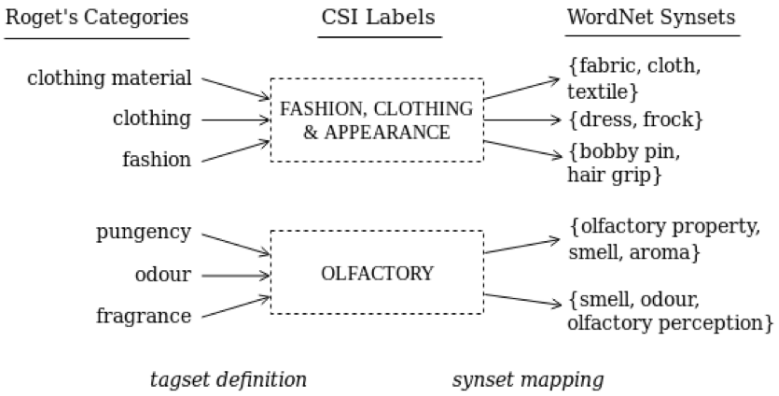
**Per Hanks, tutte le combinazioni possibili dei semantic types per quegli slot, sono tutti i possibili significati attribuibili a quel verbo.**

**Variazioni sintattiche: ci sono modi per riconoscere due combinazioni diverse che in realtà sono la stessa. Ad esempio, frasi passive e attive, perchè sono sintatticamente diverse ma semanticamente no.**

**Problematiche: Parliamo del concetto di "Scuola", quale proprietà serve per capire se il semantic type si riferisce a uno Studente, una persona o ad un essere vivente?**

**Wordnet Super Senses: Si ha una tassonomia in cui si hanno diversi alberi e le radici di tali alberi sono i supersensi divisi in PoS (Part to Speech).**

**CSI (altro modo per Semantic Types): Inventory di sensi grossolani più astratti e si hanno tutti i synset di wordnet con la categoria associata e possono essere usati come supersensi.**




## IL TESTO SECONDO LA STATISTICA:

**Linguistica Computazionale (NLP): studio di formalismi descrittivi del funzionamento del linguaggio naturale che permettano di essere trasformati in programmi eseguibili dai computer. Questo è un approccio top-down (studio regole ed eventuale codifica).**

**Statistico: Ha fatto mutare l'approccio tradizionale dell'AI che utilizzava solo delle regole condizionali, andando a studiare qualitativamente e quantitativamente particolare fenomeni. Questo è un approccio bottom-up.**

**Dal modello booleano si passa al modello extended boolean per arrivare al Vector Space Model.**

**Vector Space Model: codifica della conoscenza testuale tramite valori numerici.**

**Un testo è un insieme di token con una certa frequenza. La collezione di questi token formano un dizionario da quei testi o collezioni di testi (collezionati in un set, perché non devono esistere duplicati).**

**Questo dizionario veniva proiettato su un asse e ad ogni token veniva assegnato un identificatore univoco.**

**Questa metodologia creava molta sparsità.**

**Ogni parola veniva rappresentata nel vettore testuale con un numero che indicava le occorrenze di quella parola nel testo.**

**Una collezione di testi (n testi => n vettori) diventa una matrice.**

**Per confrontare testi codificati come vettori numerici è possibile utilizzare la Cosine Similarity:**

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

a) **Se un testo utilizza una data parola e l'altro no, la moltiplicazione viene annichilita (dal fatto che non ci sia nessun match);**
b) **Se invece è utilizzata, la misura restituisce un valore > 0 e concorre alla somma globale dei valori di similarità**
c) **Possono essere visti come punti in uno spazio multidimensionale, anche come puntatori (vettori direzionali, con verso e direzione) ed ogni vettore è un testo.**
d) **Per la comparazione dei testi, l'idea migliore è di andare a vedere l'angolo coseno, che mi dà un valore informativo semantico che pressapoco è il seguente:**

   **I due testi hanno una proporzione simile di parole utilizzate, perchè se sono vicino con l'angolo coseno significa che più o meno utilizzano le stesse parole (token).**

e) **Il numero di token crea un bias che non è importante nel momento in cui voglio confrontare due testi piccoli piuttosto che uno piccolo con uno molto grande. I valori numeri della moltiplicazione saranno molto alti e questo mi crea una sorta di bias che non mi permette di andare a comparare questi calcoli tra di loro perché questo dipenderebbe dalla norma(lunghezza) dei vettori in quello spazio multidimensionale. In questo modo con tale formula posso ignorare la lunghezza dei testi e vado a vedere solo la proporzionalità dell'utilizzo delle parole nel testo.**

**I metodi statistici applicati ai documenti di testo si basano essenzialmente su due tipi di informazioni statistiche:**

1) **Frequenza di una parola nel testo:**

a) **Term Frequency: frequenza di token(parole) all'interno di un testo**

$$TF = \frac{n_i}{N} \quad 0 \leq TF \leq 1$$

**n_i: numero occorrenze del token nel testo**

**N: numero token totale all'interno del testo**

**Non è il numero di occorrenze, ma il numero di occorrenze già normalizzato, ossia una percentuale di quanto quel termine occorre all'interno del testo.**

### b) Inverse Document Frequency

$$IDF = \log(\frac{nd}{ndt})$$

**nd: numero di documenti nella collezione in cui compare quel singolo termine**

**ndt: numero documenti totali**

**Se nd = ndt (ossia rapporto da 1), sappiamo che il log 1 = 0, ossia mi dà informazione dell'utilità entropica di quel termine.**

**Se lo trovo in tutti i documenti varrá 0 o quasi in tutti i documenti varrá poco piú di 0.**

**Se occorre sempre, significa che non è utile allo scopo di analisi di similarità, ossia non è rilevante.**

**Ad esempio le stopword, sono utili per la grammatica ma non per il calcolo delle similarità.**

### 2) Co-occorrenza di parole in un testo:

**Si ha una matrice come modello informativo, ma nelle righe e colonne ho sempre la stessa informazione (le parole).**

**All'interno delle celle inserisco valori di co-occorrenza.**

**Parsifico la mia collezione di testi e ogni volta che la mia parola i-esima viene usata nello stesso paragrafo (singola frase) allora vado a incrementare questo valore di co-occorrenza.**

**Alla fine si otterrà un valore alto per parole che co-occorrono insieme nel testo.**

**La matrice è diagonale (la co-occorrenza di "dog e cat" è la stessa di "cat e dog").**

**Questa matrice serve per altri tipi di analisi semantico del tipo "Text Mining" che permette di dire che se cat e dog hanno valori di concorrenza alti, vuol dire che compaiono negli stessi contesti e quindi condividono contenuto semantico ( non sono proprio distanti) e questo mi permette di calcolare degli score di similarità.**

**Tramite le co-occorrenze si può migliorare il risultato del calcolo della similarity andando a fare opportune moderazioni sullo string matching.**

**Il contesto puó variare all'interno di una singola frase, di un testo, di un documento ed è un concetto di intorno testuale**

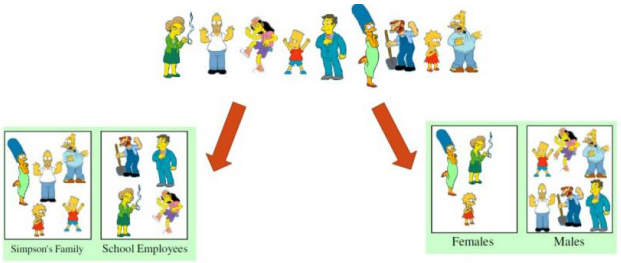
## Text Mining

**Tag clouds: uno dei primi utilizzi del tag clouds forse è stato fatto su piattaforme come flick, dove per esplorare contenuti di immagini si usavano dei tag che gli utenti inserivano insieme all'immagine. Oggi sono definiti come "hashtag" e vengono proiettati in tale modo ed erano anche interattivi. Si poteva cliccare su alcuni tag per sotto specificare di più la natura della sua ricerca. Piú il termine era grande e piú era dominante nella collezione (si faceva conto alla frequenza ma si poteva considerare anche la co-occorrenza).**


**Document Clustering: E' un metodo non supervisionato. Si tratta di calcolare la similarità dei punti 2 a 2 (3 a 3, o come ci piace) e ottengo la matrice di similarità basata su distanza.**

**Posso utilizzare algoritmi del data mining per raggruppare automaticamente questi dati. Quindi applico la Cosine Similarity e poi l'algoritmo di clustering che raggruppa autonomamente quelli piú simili tra loro e me li distanzia da altri che sono piú coesi.**

**Esempio: Invece di utilizzare il tag clouds, posso passare per un algoritmo di clustering e poi visualizzare tutto come tag clouds:**

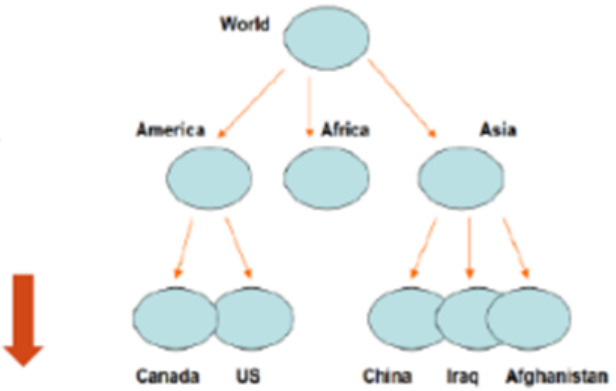


**Document Categorization/Classification:** **E'un metodo supervisionato. Raggruppamento in base documentale, ma i cluster esistono già e ci sono anche già delle etichette sui cluster.**

**Algoritmo:**

a) **Ho un set di documenti che trasferisco nel vector space model;**
b) **Ho una categoria/schema gerarchico di label (ad esempio per aree geografiche, per temi dei documenti, ecc…);**
c) **Come inserisco automaticamente questi documenti nei singoli nodi?**
d) **Si ha una gerarchia di tipo geografico:**

Esempio: Tassonomia geografica

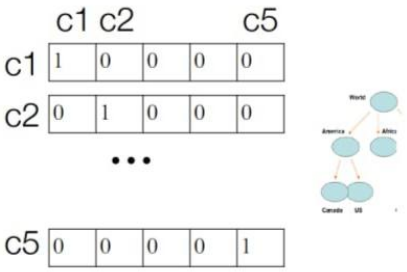



Concept vectors

| | world | Asia | Africa | America | Afghanistan | Iraq | China | Canada | US |
|---|---|---|---|---|---|---|---|---|---|
| $\vec{cv}_{world}$ | 0.450 | 0.169 | 0.141 | 0.158 | 0.018 | 0.018 | 0.018 | 0.021 | 0.021 |
| $\vec{cv}_{Asia}$ | 0.052 | 0.469 | 0.006 | 0.006 | 0.156 | 0.156 | 0.156 | 0.0003 | 0.0003 |
| $\vec{cv}_{Africa}$ | 0.100 | 0.012 | 0.873 | 0.012 | 0.0006 | 0.0006 | 0.0006 | 0.0007 | 0.0007 |
| $\vec{cv}_{America}$ | 0.057 | 0.007 | 0.007 | 0.520 | 0.0003 | 0.0003 | 0.0003 | 0.204 | 0.204 |
| $\vec{cv}_{Afghanistan}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.872 | 0.012 | 0.012 | 0 | 0 |
| $\vec{cv}_{Iraq}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.012 | 0.872 | 0.012 | 0 | 0 |
| $\vec{cv}_{China}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.012 | 0.012 | 0.872 | 0 | 0 |
| $\vec{cv}_{Canada}$ | 0.006 | 0.0003 | 0.0003 | 0.165 | 0 | 0 | 0 | 0.806 | 0.023 |
| $\vec{cv}_{US}$ | 0.006 | 0.0003 | 0.0003 | 0.165 | 0 | 0 | 0 | 0.023 | 0.806 |

**Si vuole associare ogni singolo testo al nodo della gerarchia. Si utilizza un modo chiamato CP/CV. E' basato su due fasi:**

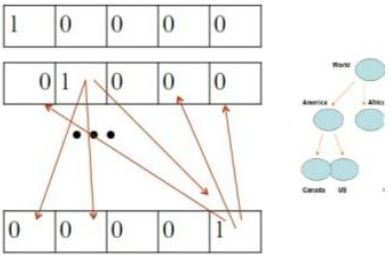
a) **Processo di inizializzazione:**



**Nella matrice si hanno concetti e nodi della gerarchia (ad esempio c1: World, c2: America, ecc..). Si crea un vettore per tutti i concetti e si danno 0 su tutti i concetti che non rappresentano quel concetto stesso.**

**b) Processo di Propagazione:**

**Questi nodi sono in relazione di "intorno" fra di loro e rispettano una certa topologia:**




**Propaga i valori delle sue dimensioni verso altre dimensioni che sono collegate tassonomicamente (Ad esempio "Africa" trasferisce un pó della sua dimensionalitá).**

**C'è un fattore α detto "fattore di propagazione" che indica quanto trasferire da un nodo all'altro. Questo vale perché trasferire da un World un pó di informazione all'Africa è necessario, in quanto Africa rappresenta un pezzo di mondo. Ad esempio ad ogni ciclo potrebbe distribuire il 10% ad Africa e il 15 % ad America (ipotesi esempio).**

**Il risultato finale è che Asia è il pezzo di mondo piú grande e viene propagata piú informazione, anche un bel pó per Africa e Afghanistan mentre il resto è tutto meno legato al nodo World.**

**Questo metodo viene utilizzato per trasferire i documenti nei singoli nodi perché qui dentro alcuni ID erano relativi a token che parlavano di entità geografiche e unendo lo spazio vettoriale della base documentale con lo spazio derivante dal CP/CV si poteva calcolare la similarità tra un testo e un nodo, e il testo poteva essere associato al nodo con la similarità piú alta.**

## Document Segmentation:

**E' una sorta di clustering sequenziale e intro documentale.**

**Non ha bisogno di etichette di partenza, si parte da un input che è un testo e l'output consiste nel trovare le linee di target, ossia il punto di interruzione di un discorso e l'inizio di un altro argomento.**

**Si possono usare diverse informazioni, non solo i token. In poche parole, si può utilizzare il cambio di dizionario che a un certo punto si avrà, ma esistono anche tecniche più avanzate che utilizzano le named entities. Ad esempio parlo di Putin e ad un certo punto non ne parlo più. Queste possono essere informazioni semantiche che mi indicano il cambio di argomento.**

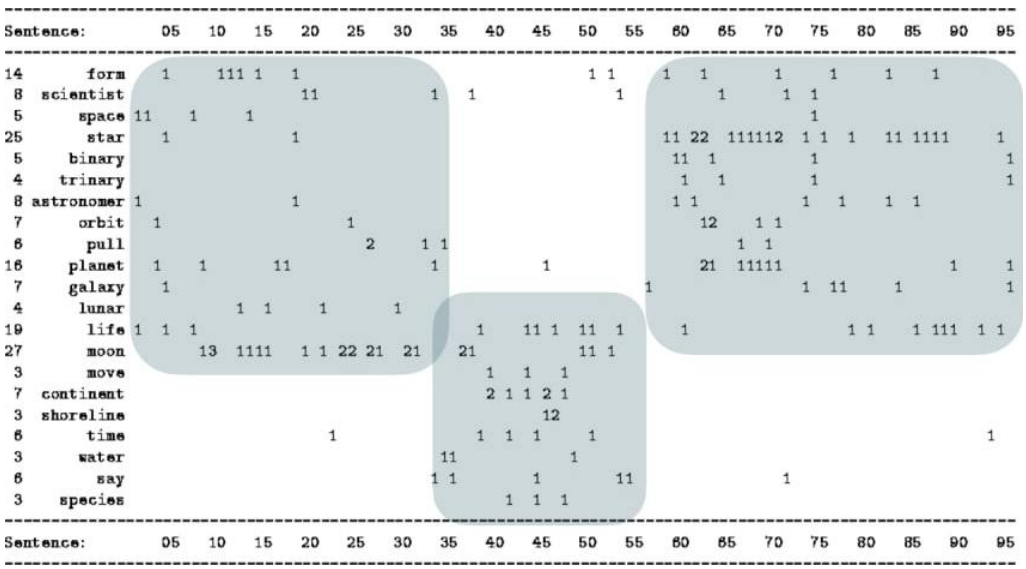
**Uno degli algoritmi piú famosi è il Text Tiling.**

**Algoritmo: Si costruisce una matrice in cui le colonne contengono il numero della frase (ossia dove inizia) e le righe i termini che vengono utilizzati.**

**Dentro questa matrice che ha carattere temporale/sequenziale di lettura (da sinistra a destra) sono contenuti dei numeri di frequenza di utilizzo di determinate parole.**

**L' ordine delle parole della matrice è scelto ad hoc solo per spiegare l'algoritmo.**

**L'output contiene il numero di inizio della frase dove cambia il discorso.**

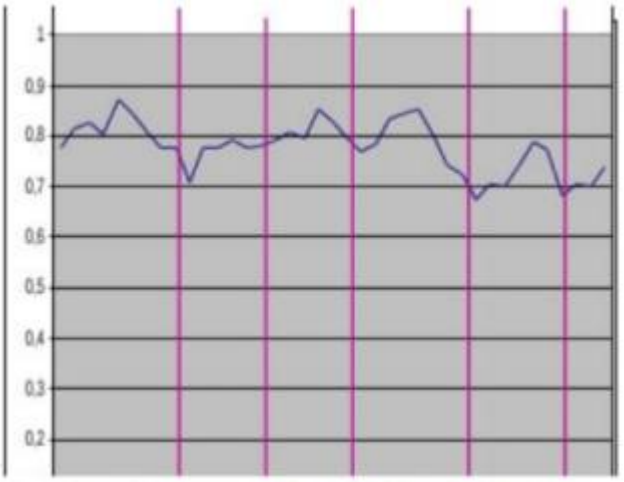



**Passi dell'algoritmo:**

a) **Separa il testo con delle finestre di lunghezza fissa (linee di taglio);**
b) **l'algoritmo le riposiziona in maniera iterativa;**
c) **all'inizio tali linee vengono date in maniera random (ne ho 3 in input e le mette a caso);**
d) **Prende queste finestre posizionate randomicamente e calcola una misura di sovrapposizione media dei termini nelle frasi in quel gruppo.**
e) **Se proietto su assi x,y, queste sovrapposizioni lessicali tra frasi sequenziali 2 a 2, ottengo la curva blu nel grafico (posso prenderle anche 3 a 3, o come voglio).**
f) **Poi calcolo un valore medio di coesione intragruppo.**
g) **Poi l'algoritmo riadatta le finestre cercando di spostarmi verso i picchi in basso, ossia i picchi dove la sovrapposizione è minore e la probabilità è piú alta che il discorso cambi. Lo fa in maniera iterativa, perché nel momento in**

**cui all'iterazione 1 posiziona le finestre poi devo ricalcolare la coesione intragruppo.**

**Termina inserendo il massimo di iterazioni dell'algoritmo in input o esistono altre varianti, ad esempio con la percentuale di cambiamento delle linee di taglio, l'algoritmo può terminare.**




**Al giorno d'oggi esistono algoritmi più avanzati.**

**Document Summarization:**

**Approccio puramente statistico per il text mining. Basato su token ed è il task che sta avendo piú attenzioni dalla comunità scientifica.**

**Metodi utilizzati:**

**Estrattivi:**

**Input di testo molto grandi**

**Output : creare una versione ristretta (riassunto) con stesso contenuto semantico**

**Analizzo l'input e cerco di estrarre le frasi piú rilevanti e le copio nell'output.**

**Ci sono diversi modi per farlo, uno ad esempio è quello di calcolare i vettori tf e idf delle singole parole e si va a prendere quello con score piú alti e poi utilizzare queste informazioni sulle frequenze e sulle dominanze delle parole per andare a estrarre solo le frasi che contengono tali parole.**

**Astrattivi: Implicano NLG. Si fa understanding (capisco di cosa si parla nel testo) e cerco di crearne uno da zero andando a collassare questa semantica.**

**É molto piú complicata, prima si faceva solo estrattiva mentre ora con le reti neurali si hanno meccanismi particolari. Ad esempio "Encoder - Decoder".**

**Si va inizialmente a codificare in una parte della rete neurale tutta l'informazione che mi deriva dal testo. Questa informazione vado poi a decodificarla e da questo punto avviene la compressione semantica. Si vá così a generare un altro testo, perció viene detto "Decoder".**

**La valutazione dei meccanismi di Summarization, che sia estrattiva o astrattiva, si fa con ROUGE (Recall Oriented Understudy for Gisting Evaluation) che preso un documento e il suo riassunto (summary) creato manualmente, per la valutazione considera quello generato automaticamente e lo confronta con quello creato da una persona tramite string matching (guarda quanti termini sono stati utilizzati nel riassunto generato automaticamente rispetto a quello manuale).**

**Molto debole come tecnica.**

**Orienteering & Browsing:**

**Nel momento in cui interagisco con una sorgente di interazioni comprendo anche quello che sto leggendo.**

**É anche una sorgente per sotto specificare meglio le query (riutilizzo piú strumenti e piú keyword per conoscere sempre di piú).**

**Information Retrieval:**

**Input: base documentale, bisogna fare una ricerca su base documentale tramite keywords (barra google)**

**Google costruisce il knowledge graph e non lo restituisce tutto.**

**Mappatura del knowledge graph sul meccanismo di ricerca (lista risultati)**

**Si sta lavorando ancora su information retrieval su agenti conversazionali (chatbot), ossia una ricerca che si protrae all'interno di una conversazione e non solo botta e risposta.**

**Integrazione di immagini, video e mappe**

**Navigazione aumentata attraverso link, snippet (parte di testo che senza cliccare si mostra un'anteprima dell'altra pagina e ci racconta una parte di testo di quello che c'è all'interno della pagina).**

## 4) Lezione 6: Semantica Documentale e Visualizzazione

**Task: Si ha una collezione di documenti, e consiste nel dire di che cosa parla questo insieme di documenti.**

**Per topic si identifica l'argomento, per modeling si identifica la modellazione.**

**Dynamic Topic Modeling: Dynamic sta per temporale.**

# Text Visualization

**Topic Modeling: Modello statistico e probabilistico che analizza l'uso del linguaggio e individua automaticamente gli argomenti in una collezione di testi (o base documentale).**

**Utilizza la co-occorrenza che è molto piú importante della frequenza.**

**É un modello non supervisionato.**

**Un topic è una lista pesata di parole.**

**Problemi:**

1) **L'interpretabilità non è cosí ovvia;**
2) **Topic estratti non sempre utili.**

**Gli algoritmi di topic modeling necessitano del parametro in input che è il numero di topic da estrarre.**

**Esistono due modi:**

1) **Latent Semantic Analysis (LSA):**

**Basata su una fattorizzazione matriciale chiamata SVD (Singular Value Decomposition)**

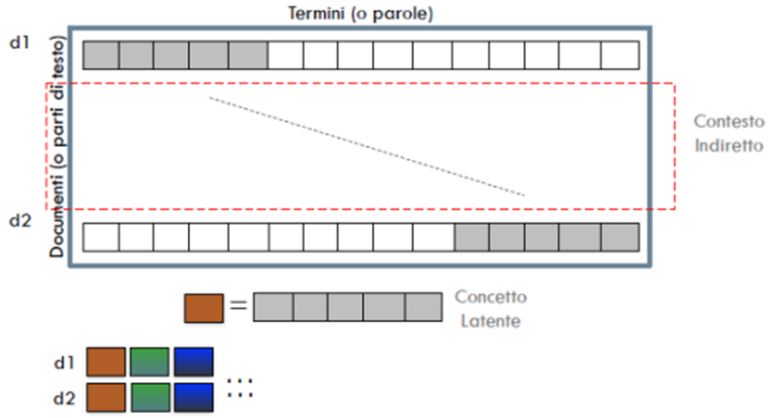
**SVD: Data una matrice n x n in input, approssima quella matrice tramite la combinazione di 3 matrici in moltiplicazione tra di loro.**

**Queste 3 matrici:**

1) **La prima rappresenta l'informazione delle righe della matrice di partenza;**
2) **La terza rappresenta l'informazione delle colonne della matrice di partenza;**
3) **La seconda è una matrice diagonale che ha solo 0 e tutti 1 sulla diagonale principale e rappresentano delle combinazioni lineari delle righe e colonne della matrice originale. Ci dice quanta entropia riusciamo a collassare dalle altre due matrici**

**Tale fattorizzazione è importante perché analizza la ridondanza della matrice di partenza e la prova a rappresentare come una combinazione lineare delle righe e colonne di partenza.**

**Presa una matrice di termini, l'algoritmo di SVD rappresenta tale matrice comprimendo l'informazione (occupa meno spazio) cercando di passare da una matrice sparsa a una densa, catturando dei concetti latenti.**




**Sulle righe abbiamo i vari documenti (d1,...,dn) e sulle colonne i termini del dizionario di quel documento.**

**Nelle celle abbiamo 0 se quel termine non viene utilizzato in quel documento e 1 altrimenti.**

**Evidenziamo le zone colorate di grigio come 1 e le zone bianche come 0.**

**Come vediamo, d1 e d2 non condividono nessun termine e c'è un overlap lessicale pari a 0.**

**Tra il d1 e d2, possiamo evidenziare il contesto indiretto, ossia tutti gli altri n - 2 documenti della collezione.**

**Se diamo in input tale matrice alla SVD, i termini che co-occorrono spesso vengono fusi in un nuovo spazio vettoriale ridotto.**

**In tali documenti ci possono essere co- occorrenze bidirezionali (indirette) e vengono collassate dalla fattorizzazione matriciale in concetti latenti, ossia tutti questi termini utilizzati assieme vengono uniti tutti in uniche features (colonne nuove della matrice).**

**Quello che si va a fare, è semplicemente un collasso delle informazioni ridondanti che si trovano spesso nella matrice di partenza.**

**In questo modo d1 e d2 esporranno un valore > 0 e se vado ad applicare la Cosine Similarity nel prodotto ottengo valore > 0.**

**Questo significa che latentemente incapsuliamo delle co-occorrenze in una matrice documento-termine.**

**Problemi: Non è un modello che generalizza su documento non visti. Se ho 1000 documenti e la mia matrice è già costruita su una paio di documenti e provo ad aggiungerne altri 1000, bisogna ricostruirla da zero altrimenti non li si può proiettare nel nuovo spazio.**

### 2) Latent Dirichlet Allocation (LDA):

**É un altro meccanismo che parte dalla probabilistic LSA basandosi su una distribuzione di probabilità di Dirichlet, che ha rivisitato il modello generalizzando e rendendolo piú performante ed efficace.**

**Si basa su statistica di Bayes**

**Si basa sull'assunto che un documento è un mix di topics e ogni parola ha una certa probabilità che compaia in ogni singolo topic.**

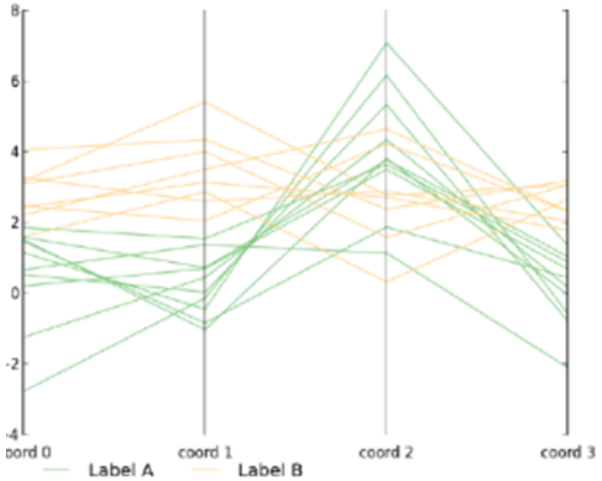
**É piú potente del LSA tradizionale perché se ho una probabilità che un termine compaia in un topic, ho un modello generativo che mi permetti di inferire dai topic, già estratti da un modello, anche da un documento nuovo.**

**Dynamic Topic Modeling: È un altra sottoarea di ricerca.**

**Ci si basa su una collezione di testi con un timestamp per ogni singolo documento e sono distribuiti su un asse temporale ed esistono tecniche di topic modeling che dividono in finestre temporali le collezioni di testi, effettuano diverse LDA, oppure vincolano alla prima LDA tutte le restanti, per catturare la modifica dei topic per proiettare un evoluzione dei topic nel tempo.**

## Tecniche di visualizzazione, approccio grafico:

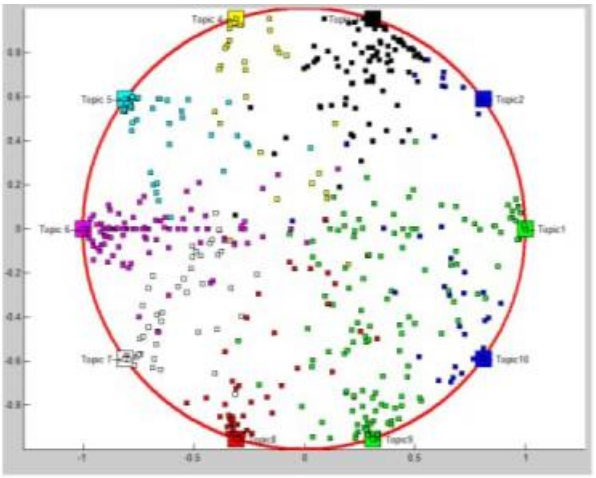
1) **Parallel Coordinates: Utilizzate quando la dimensionalitá va da 3 a 10. Queste 3 dimensioni vengono proiettate come coordinate parallele.**




**Un valore è rappresentato da una "spezzata" che collega i valori sulle 4 dimensioni. Otteniamo "spezzate" che si sovrappongono e facendo clustering visivo tra queste righe posso individuare pattern visivi che si comportano in maniera diversa da altri pattern.**

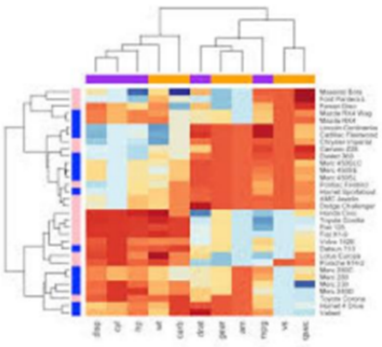
2) **Radviz (Radial Visualization): È un'altra strategia grafica che permette di visualizzare massimo 10/20 dimensioni organizzate su circonferenza. I punti**

**all’interno sono i punti nello spazio multidimensionale. Si può pensare che queste dimensioni siano dei magneti, piúè altro il valore per quelle coordinate e piú il punto viene attratto dalla dimensione posta sulla circonferenza.**

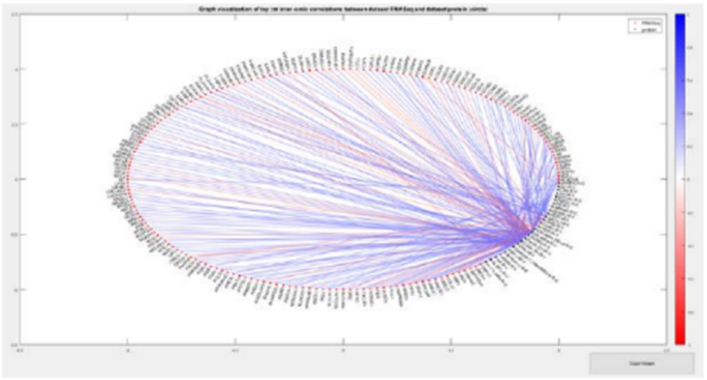



**Problemi: legati al fatto che se due dimensioni hanno relazione, ossia i punti di una dimensione hanno qualcosa in comune con l’altra, i punti finiscono in mezzo alle due classi e c'è una incidenza. Le attrazioni reciproche non permettono di stabilire bene dove mettere i punti.**

3) **Heatmap: Meccanismo per visualizzare matrici numeriche con N molto grande e vengono sostituite dal colore. Permette di evidenziare cluster di dati in maniera visiva.**



4) **Correlation circle: Si basano sul cerchio, ma gli elementi vengono inseriti direttamente lungo la circonferenza e all’interno si proiettano le correlazioni tra gli item.**



## 5) Lezione 8: Distributional Semantics:

**Rivisitazione in chiave linguistica del Text Mining.**

**Nasce tutto negli anni 54/57 con Harris e Firth che arrivano entrambe alla stessa conclusione.**

**Harris: Le parole che occorrono simili negli stessi contesti  tendono a tenere significati simili.**

**Firth: Una parola é caratterizzata dalla sua compagnia.**

**Negli anni 83, questi concetti sono stati rivisti nell'ambito dell'information retrieval.**

**Furnas: L'uso congiuntivo delle parole serve per specificare meglio gli elementi nel discorso.**

**Negli anni 90, con l'applicazione della matrice SVD al documento:**

**Deerwester: C'è una struttura latente semantica nei dati che è oscurata dalla casualità di scelta delle parole rispetto all'utilizzo delle parole durante l'interrogazione.**

**Nel 2003 con l'estrazione dal topic modeling:**

**Blei: Il tema dell'argomento influenza probabilisticamente la scelta delle parole.**

**Turney: Coppie di parole che co-occorrono in contesti simili tendono ad avere relazioni semantiche simili con il resto del mondo.**

**Nel Text Mining perché utilizzare le matrici? Perché sono strutture che si utilizzano bene e possono rappresentare qualsiasi dato che ha la forma X ○ Y**

**La distributional semantics funziona bene perché c'è una certa ambiguità tra la confusione di una matrice numerica con l'ambiguità del linguaggio.**

**Le matrici da un punto di vista teorico, si collocano tra i meccanismi simbolici e l'associazionismo. Quest'ultima dice che tutto è collegato con tutto, ma in realtà non è cosí, ma diciamo che molti oggetti condividono una regione di qualities features (qualcosa condivide qualcosa).**

**É una rappresentazione che facilita la conoscenza, perché avere lo spazio di colonne (qualities) che vengono condivise tra tutte (def di conceptual space).**

**Il meccanismo della matrice è anche collegabile con la prototype theory che dice che se abbiamo delle aree semantiche, dentro ci troviamo degli individui e in questi cluster c'è sempre un centroide.**

**lo stemming, la lemmatizzazione, ecc.. sono tutte tecniche che normalizzano e restringono la cardinalitá del dizionario. In realtá, si possono adottare anche tecniche di de-normalizzazione, ossia aumento con nuova sparsitá, ma l'aumento su carattere semantico, mentre la normalizzazione é linguistica.**

**Configurazione Matriciali:**

**Ci sono 3 utilizzi di queste matrici:**

1) **Term Document Matrix: Le righe sono i documenti e le colonne i termini. Contengono la frequenza di un termine in quel documento;**
2) **Term Context Matrix: Le righe sono parole/termini mentre le colonne è il contesto. Ci indica le frasi o dipendenze sintattiche in cui quei termini occorrono. Si usa per task come word similarity, word clustering, word classification, automatic thesaurus generation, word sense disambiguation, ecc.**
3) **Pair Pattern Matrix: Le righe sono coppie (word x, word y) e le colonne dei pattern.**

**Risolvono task che prima non si riusciva a risolvere.**

**Si hanno delle coppie perché non si analizza la semantica distribuzionale di una parola sola, ma di piú parole.**

**Per pattern, si intende dei pattern lessico-sintattici in mezzo a X e Y.**

**Se prendo (x,y) e (z,w), se risultano simili tramite cosine similarity, ossia condividono uno spazio comune, allora le loro relazioni semantiche sono simili.**

**Task: Clustering di parole in relazione fra di loro oppure clustering sulle colonne (pattern linguistici diversi, condivisi da tante parole) che sono modi diversi di legare i termini fra di loro, ma esprimendo la stessa semantica relazionale.**

**Similaritá:**

**Quinn: La similaritá è un concetto che ci serve per il pensiero, per l'apprendimento, ci serve anche per ordinare le cose in classi e che permette di stimolare sul significato.**

**Esistono diverse similaritá:**

a) **Semantic similarity;**
b) **Semantic Relatedness;**
c) **Attributional Similarity;**

d) **Taxonomical Similarity;**
e) **Relational Similarity;**
f) **Semantic Association.**

**Problemi:**

a) **Si perde l'ordine delle parole;**
b) **rappresentazione non composizionale;**

**Arricchimento matriciale:**

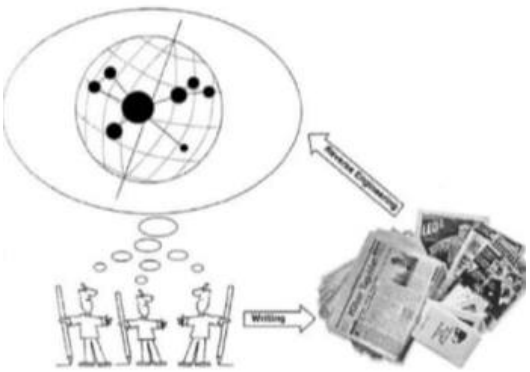
a) **Approcci utili arrivano dai Knowledge Graph codificati in un DB: semantica giá codificata in termini di nodi, azioni ed etichette. Dove arriva conoscenza statistica la uso, altrimenti utilizzo approcci piú moderno dati da NLP + Knowledge Graph.**

**Significato filamentare: studio che arriva dai word space model, la semantica delle parole contenute negli spazi vettoriali è da ritrovarsi in piccole proiezioni/filamenti relazionali.**

## 6) Lezione 9: Ontology Learning e Open Information Extraction

**Ontology Learning: Apprendimento in maniera automatica o semi automatica di un ontologia.**

**Descrizione del mondo e traduzione per la creazione di documentazione e distillazione della conoscenza scritta dalla quale si può fare Reverse Engineering per tornare alle concettualizzazioni originali.**



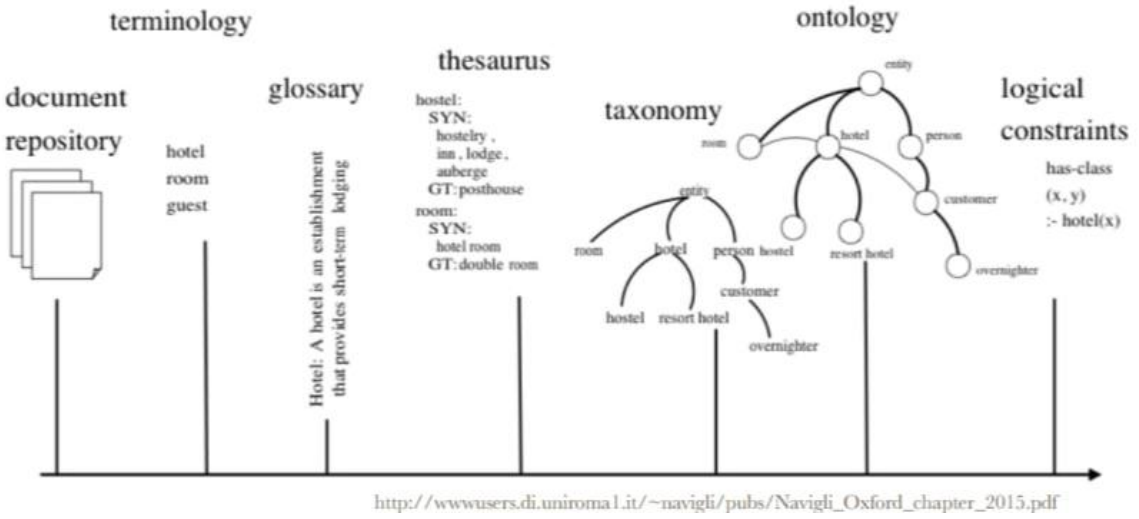
**Non si codifica tutta la conoscenza del dominio, dipende dal task.**

**Si cerca sempre il trade off tra effort nella costruzione dell'ontologia, usabilitá e task considerati.**

**Ontology population:** popolare un ontologia, ossia estrarre concetti da una base documentale e inserirli all'interno dei cluster (popolo l'ontologia).

**Ontology based annotation:** Creo una base documentale arricchita evidenziando simbolicamente nel testo dei termini che riflettono dei concetti di un'ontologia di riferimento.

**Ontology Enrichment:** Lo scopo non è solo popolarla, ma capire cosa manca in quell'ontologia.




http://wwwusers.di.uniroma1.it/~navigli/pubs/Navigli_Oxford_chapter_2015.pdf

**Livelli di profondità:** da sinistra a destra aumenta la complessità dell'ontology learning.

**Glossario:** in piú del dizionario ha anche la definizione ad esso associata.

**Thesaurus:** lega i termini in maniera orizzontale (lega relazioni sinonimiche, come ad esempio WordNet.

**Taxonomy:** codifica delle relazioni tassonomiche.

**Ontology:** é la cosa piú generica che si puó avere, in cui ci sono concetti legati da relazioni semantiche di ogni tipo.

**Logical Constraints:** rappresentazione logico-formale dove si specificano regole di inferenza.

**Task:**

1) **Term Extraction:** Trovare nomi per concetti e relazioni
2) **Synonym Extraction:** Trovare sinonimi
3) **Concept Extraction:**
    a) **Intension (Intensionale):** trovare la def del concetto (gloss learning)
    b) **Extension (Estensionale):** tutta la terminologia utilizzabile per riferirsi a determinati concetti.

**4) Concept Hierarchies induction: induco direttamente gerarchie;**

**5) Relation Extraction: estraggo relazioni semantiche generiche;**

**6) Population (NER, IE):**

a) **Scatta la regola per fare information extraction;**
b) **pattern matching nel testo con BERT;**
c) **NER per identificare persone;**
d) **NEL per associarle a uno specifico contesto;**

## Concept Hierarchies Extraction

**Metodi:**

1) **Natural Language Processing: approcci di NLP semplici per fare Ontology Learning:**
   a) **Ruolo delle Part of Speech (POS) che identificano specifiche istanze;**
   b) **Preprocessing: Normalizzazione, Lemmatizzazione, Stemming;**
   c) **Analisi sintattica;**
   d) **Similaritá;**
   e) **Risorse Linguistiche.**

2) **Formal Concept Analysis (FCA): metodo che arriva dalla matematica usato per fare estrazione di tassonomie a livello automatico.**

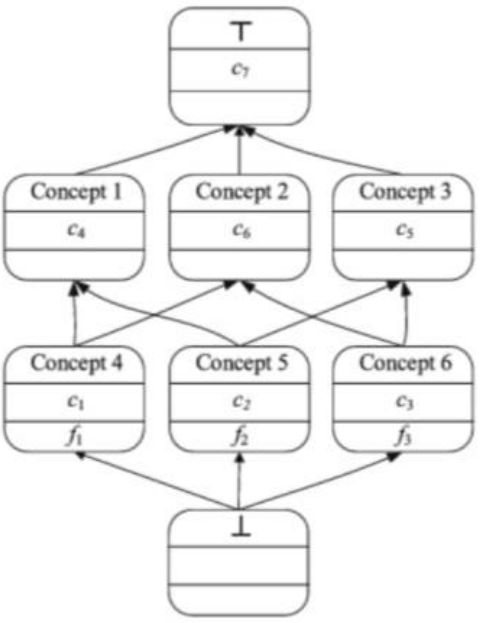
| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |
|---|---|---|---|---|---|---|---|
| $f_1$ | x | | | x | | x | x |
| $f_2$ | | x | | x | x | | x |
| $f_3$ | | | x | | x | x | x |

(a)

| | |
|---|---|
| ⊤ | $\{f_1, f_2, f_3\}$ , $\{c_7\}$ |
| Concept 1 | $\{f_1, f_2\}$ , $\{c_4, c_7\}$ |
| Concept 2 | $\{f_1, f_3\}$ , $\{c_6, c_7\}$ |
| Concept 3 | $\{f_2, f_3\}$ , $\{c_5, c_7\}$ |
| Concept 4 | $\{f_1\}$ , $\{c_1, c_4, c_6, c_7\}$ |
| Concept 5 | $\{f_2\}$ , $\{c_2, c_4, c_5, c_7\}$ |
| Concept 6 | $\{f_3\}$ , $\{c_3, c_5, c_6, c_7\}$ |
| ⊥ | $\{\emptyset\}$ , $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ |

(b)

(c)




**Esprimiamo un concetto (istanza del nostro dominio) attraverso un insieme di features.**

**Nelle celle delle matrice (a) segnamo con "x" se quel concetto ha quella determinata caratteristica.**

**Passando dalla matrice (a) alla matrice (b) diciamo che questa matrice (b) si chiama Formal Context.**

**Parto eliminando una delle tre features, ossia scelgo f_3 e cerco quei concetti che hanno entrambe {f_1,f_2}, ossia {c_4,c_7}. Dopodiché facendolo tra tutte le combinazioni ed eliminando le simmetrie costruisco il reticolo.**

**La FCA è un metodo formale deterministico che prende la matrice (a) e costruisce il Formal Context (b) da cui ci ricavo anche il reticolo (lattice).**

**I concetti (c_1,c_2,...,c_n) nella tabella (a) sono concetti latenti, ossia concetti estratti autonomamente che hanno in sé un aspetto intensionale e uno estensionale.**

**L'intensionale sono le features.**

**Le estensionali sono gli elementi della base di dati iniziale che soddisfano e rientrano all'interno di quel sottospazio di features.**

**Il reticolo può essere visto come una tassonomia perché si hanno da una parte un insieme di features condivise in modo sequenziale. Ad esempio mela e pera hanno features in comune, ma anche altre diverse, quindi si sotto specificano.**

**Open Information Extraction: É una specie di Ontology Learning**

**L'ontology learning è definito come il passaggio da un dato non strutturato nel testo a un dato strutturato in maniera automatica. Mentre l'open information extraction è detto anche shallow parser/shallow ontology learning perché creo informazione semi strutturata.**

**Mi focalizzo sulla granularitá delle frasi e tiro fuori triplette (Hanks theory).**

**Viene detta "Open" perché si tiene aperta a informazioni su larga scala.**

**Ci sono fasi di estrazione e di filtro.**

**Problematiche:**

a) **Chiunque puó creare un sistema di OIE;**
b) **Molto difficile valutare l'OIE perché non c'è uno standard per valutare le qualità delle triplette.**

## 7) Lezione 10: Knowledge Graph e Integrazione con NLP

**Arrivano dai database a grafo piuttosto che dai classici.**

**L'importanza nel KG è quella relazionale e non dei record.**

**Hanno molta potenza in scalabilitá.**

**Matcha con le strutture dati per la forma che ha, mentre nei DB classici comporterebbe dei join, che sono operazioni costose.**

**La matrice di concorrenza non è altro che un grafo bipartito e può essere usato per trasformare il testo in un grafo con delle relazioni.**

**Perché trasformarli? Se ad esempio devo fare un'analisi, lo trasformo in un grafo bipartito ed utilizzo la teoria dei grafi per manipolare tale struttura.**

**Ogni nodo di ogni relazione puó avere metadati associati.**

**Nel caso di Neo4J esistono delle proprietà che possono essere associate a dei nodi, delle etichette e dei tipi e si può inserire la direzione di una relazione ecc.**

**Quali tipi di nodi si puó inserire dipende dal framework utilizzato.**

**Vantaggi:**

1) **La codifica dei KB/KG è self-explanatory: relazioni codificate tramite label e i nodi possono essere cose diverse (parole,...) e quindi ha natura iper leggibile;**
2) **NLP: approcci neurali recenti raggiungono lo stato dell'arte in termini di risultati in molteplici task;**
3) **Approcci Neurali eccellenti nel catturare relazioni (nei primi livelli c'è BERT).**

**Svantaggi:**

1) **Porta a una rappresentazione dei dati che non è sempre banale e puó introdurre rumore e approssimazione dei dati;**
2) **a volte servono quantitá riportate in input dal corpus in grandi quantitá e qualitá e interpretabilitá dei modelli;**
3) **sono importanti anche alcuni difetti degli approcci knowledge-based e statistico-neurali.**

**Non sempre serve fare NLP avanzato o almeno quando della conoscenza è proiettata nella KB/KG.**

**Cosa ci si puó fare con le KG:**

**Analisi con KG: con il KG posso farci determinate analisi.**

**Centralitá: Analisi riguardo alla loro topologia strutturale (PageRank, ecc..)**

**Similaritá**

**Percorsi: lunghezza, ottimizzazione funzioni, search, ecc..**

**Embeddings: Reti Neurali (Node2Vec)**

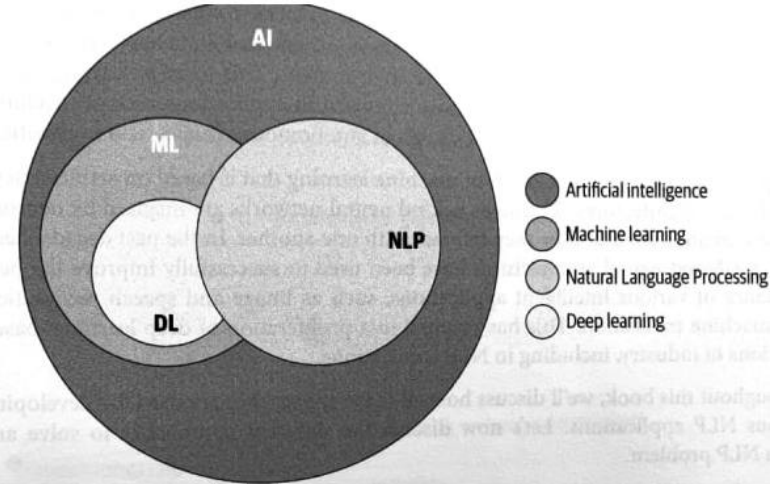
**Predizione**

**Esistono 2 approcci metodologici diversi:**

1) **Data driven:** viste, selezioni, ecc..
2) **Goal oriented:** riservato alle aziende che utilizzano piú la struttura a grafo.

**Viene utilizzato per altri tasks: Sense Disambiguation, Question Answering, Semantic Search, Sistemi di Raccomandazione, KG Completion, Entity Resolution**

## 8) Lezione 11: Overview NLP, Machine Learning, Deep Learning, AI




**Il testo costituisce i dati. Questi dati e le features (caratteristiche) dei dati li diamo in input a un modello di apprendimento che crea un nuovo modello che utilizziamo per classificare e valutare le performance.**

**Alcuni algoritmi famosi:**

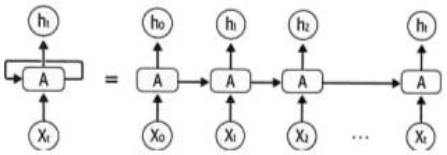
a) **Naive Bayes (Bayes,...)**
b) **Support Vector Machine (SVM)**
c) **Hidden Markov Model (HMM)**
d) **Conditional Random Fields (CRF)**

### NLP e Machine e Deep Learning

**Recurrent Neural Networks (RNN):**

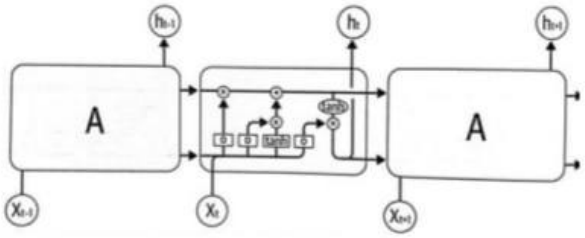
**Fondamentali perché c'è un mapping lineare tra il testo e quello che è una RNN, ossia una rete che non usa solo l'informazione in input. É una rete che piano piano va avanti nella lettura e cerca di ricordare la traccia di quello che si è detto per dare un output.**

**Il problema delle RNN è che non scalano molto bene perché se il testo è molto lungo, dal punto di vista computazionale è molto difficile utilizzarle (esplode il problema).**



**Long Short Term Memory Neural Networks (LSTMNN): Si basano sul concetto che mentre legge, alcune cose sono piú rilevanti di altre, e queste altre le si puó dimenticare.**

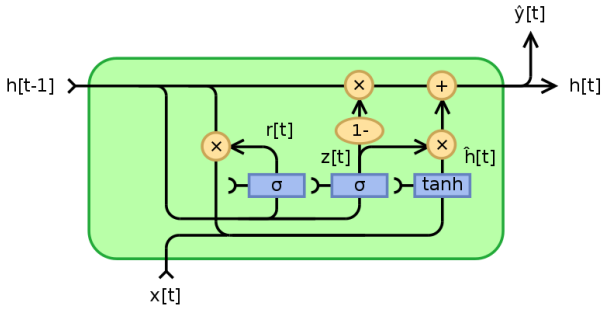
**Risolve il problema delle RNN perché tante cose che si sono dette, le perde e quindi riesce a gestire sequenze lunghe di testo e tanti dati in input e alla fine si ottiene un modello che ricorda sia long term (cose che si sono lette molto tempo prima), ma anche le short term( anche quelle che si sono lette poco prima).**




**Gated Recurrent Unit Explained Neural Networks (GRUENN):**

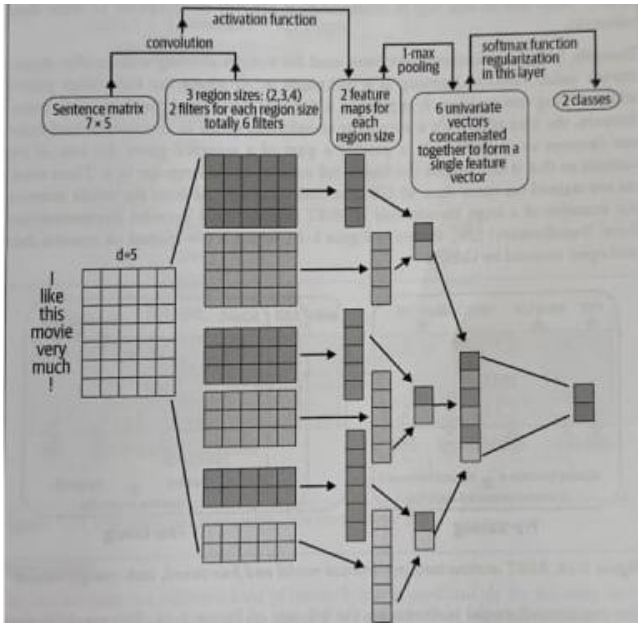
**Risolvono alcuni problemi del LSTM.**

**É un approccio piú semplice delle LSTM e seppur semplificando molto l'architettura, su pochi dati funzionano al pari del LSTM, mentre se sono molti dati sono superiori alle LSTM.**




**Convolutional Neural Networks (CNN): Si parte dal numero di filtri (finestre) che scorrono sopra la matrice e fanno una sorta di parsing della matrice di partenza, ossia ogni volta mi danno una proiezione della matrice che è una sottomatrice di quello che vedono. Ogni sottomatrice viene traslata su un vettore in maniera iterativa, ossia vado a fare tramite dei meccanismi chiamati di pooling, delle somme che contribuiscono ai valori medi su tutta la sottomatrice. Vado quindi a creare un vettore finale dove ho**

**ogni singola cella che collassa un pó di tutta l'informazione che arriva da una sottosegmentazione dell'input con il meccanismo di pooling, e con questo vettore si arriva a un modello classico, dova da un set di features vado a decidere qual'è la classe finale.**




**Transformer (Encoder - Decoder):**

**Basati su Encoder-Decoder**

**Encoder: Un set di features viene mappato su una dimensione molto piú piccola, simile alla latent semantics che cerca di comprimere una dimensione ridotta.**

**Decoder: Presa tale semantica collassata, cerca di ridescriverla e riproiettarla in uno spazio multidimensionale.**

**L'attention è un architettura che permette alle reti neurali di focalizzarsi piú su alcune parole che su altre (danno peso alle singole parole rispetto al task).**

**BERT si basa su tale meccanismo che è anche il task di masked language model.**

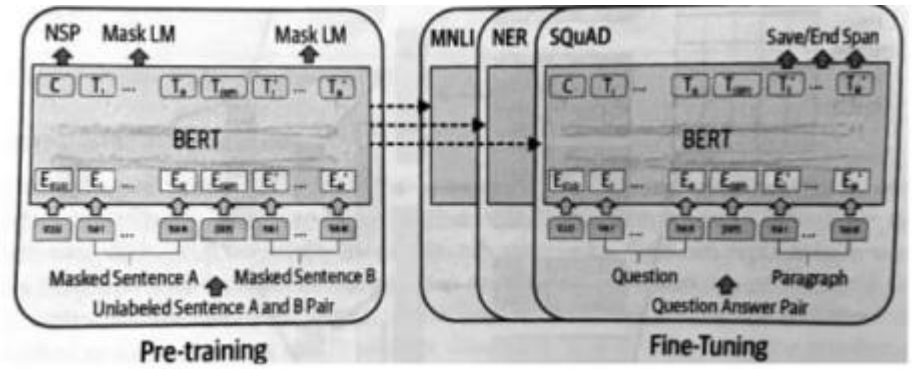
**Questo ha creato un sistema per apprendere in maniera completamente non supervisionata, senza utilizzare etichette, ma solo il testo nella sua forma non strutturata.**

**Esempio: Abbiamo tante frasi (corpus molto grande) . Ad un certo punto la rete maschera delle parole e chiede alla rete di inferire qual'e la parola mascherata, quindi apprende in maniera autonoma senza delle etichette perché dai semplici dati fa questo gioco con se stesso. Se sbaglia continua finché non apprende del tutto.**

**Questo meccanismo prende il nome di semi-supervised learning, perché non abbiamo piú bisogno di classi, delle label per costruire un training set. Se lo costruisce da solo e si crea tali task per apprendere.**

**Questi task sono gli embeddings sulle parole che riescono da soli a discriminare e a lavorare su task, come ad esempio le maschere.**

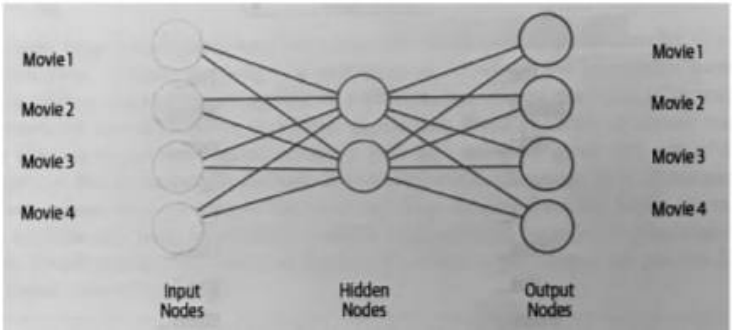
**Output: Embeddings (vettori) allenati in maniera autonoma e non pronti all'uso. Dopo avere acquisito questa conoscenza (detta di pretraining) facciamo fine-tuning, ossia contestualizziamo quegli embeddings su task reale.**




**Auto Encoders(AE): Collassiamo l'input su pochi neuroni e cerchiamo di ricostruire l'originale.**

**Nel momento in cui la rete si stabilizza (buon rapporto tra input e output ricostruito), prendiamo i livelli di mezzo (hidden nodes) che sono piccoli e questi sono i concetti latenti del LSA (ossia una bassa dimensionalitá che spiega il nostro input).**

**Qui si va oltre il mapping lineare, è un mapping non lineare e l'espressività e le performance aumentano rispetto a LSA.**




**Problemi Aperti:**

1) **Overfitting;**
2) **Few-shot Learning: learning con pochissimi dati di esempio;**
3) **Domain Adaptation: adattare degli embeddings e darli in input ad un altra rete che fá ancora dei cicli di learning su alcuni dati specifici su un dominio;**

4) **Interpretabilità: le reti neurali sono delle black box e non si sá come trasferire milioni di parametri appresi in qualcosa di interpretabile;**
5) **Common Sense:**
    a) **Costo: costo nell'uso di reti neurali sia fisici che energetici;**
    b) **Sviluppo di reti possibilmente inseribili su dispositivi mobile**