DISCLAIMER:

- **Basato sulle dispense del corso 2023**
- **Questo è un riassunto, non contiene TUTTE le informazioni, le dispense del prof sono necessarie.**
- **Questi appunti non sono verificati, non assicuro la correttezza di niente.**
- **Molti esempi NON sono ufficiali del prof, quindi potrebbero essere sbagliati.**

in altre parole:

Non mi prendo alcuna responsabilità di informazioni incorrette o incomplete.

2. Significato del significato

Triangolo semiotico (p. 11):

1. **Concetto**

   > è un'idea astratta, un'entità mentale, un'idea che si ha in testa. Il concetto è indipendente dalla cultura/lingua di una persona. Esempio: l'idea "cane" è indipendente dalla lingua che si parla.

2. **Rappresentazione**

   > è la parola con cui si esprime un concetto. Nel nostro esempio la stringa "cane". Questa può variare in base alla cultura e/o lingua (es: dog).

3. **Referente**

   > ogni singola istanza del concetto è il referente. Nel nostro esempio ogni cane fisico è referente del concetto "cane".

Nel campo dell'NLP l'unico punto del triangolo da cui si può partire è la rappresentazione, possiamo spostarci verso il concetto tramite l'analisi semantica del testo o verso il referente tramite tecniche come la computer vision.

Granularità (p. 15)

Ogni step è un gruppo dello step precedente. es: un discorso è un insieme di frasi, una frase è un insieme di parole, etc...

parola	**chunk (composizione di parole, es: articolo + nome)**	**frase**	**discorso**	**documento**	**collezione di documenti**
WSD	Multiword epressions	Question Answering	Chatbot	Document summarization	Topic modeling

WSD e WSI (p.16):

metodi per identificare il significato di una parola polisemica in un contesto specifico.

Word Sense Disambiguation (WSD):

Abbiamo un insieme di significati predefiniti (es: synset di wordnet) e dobbiamo assegnarne uno specifico.

Problematiche:

- **Specificità**

  > per una certa parola esistono troppi significati specifici, quindi ognuno di essi verrà usato raramente. es: "*Suono la chitarra*" e "*Suono la chitarra ad un concerto*" hanno un senso di suonare diverso all'interno di WordNet

- **Copertura**

  > paradossalmente rispetto al punto precedente è possibile che possano non essere presenti alcuni significati necessari.

- **Soggettività**

  > il creatore dell'insieme dei significati dovrà decidere quali significati includere e quali no.

Word Sense Induction (WSI):

Non abbiamo un insieme di significati predefiniti, analizziamo molti testi e andiamo ad estrarre il contenuto (per esempio usando clustering).

Differenze:

> Nella WSD abbiamo un dizionario di significati predefiniti, nella WSI no; la WSI prova ad assegnare i significati senza uso di risorse esterne.

> Nella WSI si prova a trovare il significato di una parola basandosi solamente sull'uso di questa all'interno del dataset, senza usare risorse umane.

> La WSD è molto basata sulla grammatica, la WSI si basa sull'uso della parola.

> La valutazione nella WSD è molto semplice, nonostante ciò non è semplice a causa di soggettività e specificità. La valutaizone nella WSI è più complessa.

Valutazione WSI (metodo pseudoword):

Prendo due parole scollegate, ad esempio "cane" e "banana", e le sostituisco entrambe con una pseudoword (es: "bananacane"). Se il metodo funziona correttamente, sarà in grado di generare due cluster distinti per la pseudoword, uno per il significato originale "cane" e uno per "banana".

Ricerca Onomasiologica (p. 18) (lab2):

Ricerca di un termine a partire dalla sua definizione.

3. Costruzione del significato:

- **Pustejovsky**: `strutture qualia`
- **Hanks**: `valenze del verbo`

Pustejovsky: generative lexicon (p. 22):

Teoria del "generative lexicon", modello molto elegante e potente ma estremamente difficile da implementare. è composto da 4 strutture:

- **Argument structure**

  > descrive gli argomenti logici e la loro realizzazione sintattica di un predicato. Ad esempio, il verbo mangiare ha due argomenti: il soggetto (chi mangia) e l'oggetto (cosa si mangia).

- **Event structure**

  > definisce il tipo di evento, i suoi partecipanti e le sue proprietà temporali e aspettuali. Ad esempio, il verbo mangiare è un processo che coinvolge un agente e un paziente.

- **Qualia structure**

  > esprime le caratteristiche essenziali di un concetto, ispirate alle quattro cause aristoteliche: formale, costitutiva, telica e agentiva 3. Ad esempio, il concetto di mela ha una struttura dei qualia che specifica la sua categoria (frutto), le sue parti (buccia, polpa, semi), la sua funzione (essere mangiata) e la sua origine (crescere su un albero).

- **Inheritance Structure**

  > colloca il concetto all'interno di una tassonomia che ne determina le relazioni di sottotipo e sovratipo con altri concetti. Ad esempio, il concetto di mela eredita alcune proprietà dal suo sovratipo frutto, ma ne specifica altre che lo distinguono dai suoi sottotipi (mela verde, mela rossa, ecc.).

Qualia structure:

La qualia structure secondo pustejovsky è composta da 4 ruoli:

- **Costitutivo**

  > esprime la parte materiale/fisica di un concetto come peso, dimensione, forma, etc... o le parti che lo compongono. *es: il concetto di mela è composto da buccia, polpa e semi.*

- **Formale**

  > definisce le caratteristiche che distinguono un concetto da altri concetti dello stesso dominio. *es: il concetto di mela è diverso da quello di pera per la sua forma, il suo colore, il suo sapore, etc...*

- **Telico**

  > esprime la funzione/obiettivo di un concetto. *es: il concetto di mela ha come funzione quella di essere mangiata.*

- **Agentivo**

  > esprime l'origine di un concetto. *es: il concetto di mela ha come origine quella di crescere su un albero.*

Hanks: teoria delle valenze (p. 24) (lab3):

secondo Hanks il verbo è la radice del significato. In questo modello i sostantivi hanno l'unico scopo di specificare il significato del verbo attraverso gli slot.

definizioni:

- **Slot**

  > Ogni verbo ha una serie di slot che devono essere riempiti da un sostantivo. Il verbo "portare" ha 3 slot: agente, paziente e destinazione. Ad esempio, la frase "Marco porta la palla a casa" riempie i 3 slot con "Marco", "palla" e "casa".

- **Filler**

  > Ogni possibile valore che uno slot può assumere.

- **Valenza**

  > La valenza è il numero di slot che un verbo ha. I verbi intransitivi hanno valenza 0 (es: "piove"). Il significato di un verbo è dettato non solo da come vengono riempiti, ma anche dalla valenza di esso (in specifico se un verbo può avere più valenze).

- **Semantic-Type**

  > Sono delle macro-categorie che raggruppano i filler. Ad esempio, il filler "Marco" è di tipo "persona", il filler "palla" è di tipo "oggetto", etc...

- **Collocazione**

  > Combinazione di tutti i possibili filler.

Il significato di un verbo è definito dalla combinazione di filler (sematic-types) utilizzati

Problematiche:

Una parola può appartenere a diversi semantic-type più o meno generali. (*es: Studente può appartenere a "studente" -> "persona" -> "essere vivente" -> "entità"*). La difficoltà di questo modello sta nel trovare il semantic-type appropriato per ogni parola in base al contesto.

Affordance:

Capacità di intuire lo scopo di un concetto/oggetto basandosi esclusivamente sulle sue proprietà.

Applicato al linguaggio, l'affordance è la capacità di intuire il significato di una parola basandosi esclusivamente sul contesto in cui viene utilizzata. (*es: "grest" singolarmente non ha significato intuibile, ma con "Yesterday I saw a grest" riusciamo a intuire che sia qualcosa di visibile e quindi concreto.*)

Applicato ad Hanks questo serve per dire che il semantic-type non è sempre lo stesso per una parola, ma cambia in base al contesto.

Pattern generativi:

- **Pattern**: Definizamo un pattern come una frase che contiene dei jolly (*), ad esempio: " * *conosce* * *molto bene*"
- **Istanze linguistiche**: Tutte le occorrenze di un pattern in un corpus. *es: "Marco conosce Maria molto bene", "Luca conosce Roma molto bene", etc...* Per ogni * ci sono diverse parole valide che saranno i filler.

4. Text Mining (p. 34):

Approccio derivato dal campo del data mining, puramente statistico, che si occupa di estrarre informazioni da un testo. A differenza della linguistica computazionale classica che è top-down, il text mining è bottom-up.

Vector Space Model (p. 34):

Per il text mining le parole sono semplici token (o sequenze di caratteri) senza valenza lessicale (a differenza della semantica distribuzionale), un testo è quindi semplicemente una sequenza (vettore) di token con una certa frequenza.

Rappresentazione vettoriale:

per rapprensentare un testo in un vettore andiamo inizialmente a creare un dizionario che associ ad ogni parola un indice, in seguito (secondo l'approccio classico) andremo a creare un vettore V, di dimensione |d| (numero di parole nel dizionario) in cui ogni posizione V[i] conterrà la frequenza nel testo della parola con indice i nel dizionario.

Se possiamo rappresentare un documento in un vettore, possiamo rappresentare una collezione di documenti in una matrice, in cui ogni riga rappresenta un documento e ogni colonna una parola.

Cosine similarity:

Possiamo immaginare ogni documento come un vettore N-dimensionale (N numero di parole nel dizionario), possiamo quindi calcolare la similarità semantica tra due documenti come l'angolo tra di essi.

Definiamo la cosine similarity fra due documenti come il coseno dell'angolo tra i due vettori che li rappresentano.

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

Metodi statistici:

Nel campo del text-mining vengono utilizzate principalmente due statistiche:

- **Frequenza**: La frequenza di una parola è la sua dominanza all'interno del documento. Uno dei metodi più utilizzati per calcolare la frequenza è il TF-IDF (Term Frequency - Inverse Document Frequency). Il

TF-IDF è una frequenza normalizzata che tiene conto di quanto comune è una parola all'interno di una collezione di documenti.

$$\text{TF-IDF} = TF * IDF = \frac{n_{i,j}}{|d_j|} * \log \frac{|D|}{|d : i \in d|}$$

$$n_{i,j} = \text{numero di occorrenze del termine } i \text{ nel documento } j$$

$$|d_j| = \text{numero di termini nel documento } j$$

$$|D| = \text{numero di documenti nella collezione}$$

$$|d : i \in d| = \text{numero di documenti che contengono il termine } i$$

la IDF è un valore che tende a 0 per termini molto comuni, questo serve per tenere conto di quando una parola è molto comune a prescindere dal documento in cui si trova.

- **Co-occorrenza**: La co-occorrenza indica la similarità statistico-semantica fra due parole. Se due parole appaiono in contesti simili possiamo inserire il loro valore di co-occorrenza in una matrice di dimensioni |d| x |d| (|d| = dimensione del dizionario) in cui la cella i,j contiene il valore di co-occorrenza fra la parola i e la parola j.

Possiamo usare la co-occorrenza per calcolare un valore di similartià più accurato rispetto al metodo classico, questo terrà conto di parole che appaiono in contesti simili (es: gatto e micio).

Applicazioni Text Mining (p. 39):

- **Tag clouds**

  > "Nuvola" di parole nella quale ogni parola è rappresentata con una dimensione proporzionale alla sua frequenza all'interno del testo. Si può implementare una funzione che mappa il vettore di co-occorrenza sullo spazio bidimensionale dell'immagine in modo da avere termini correlati vicini fra loro.

- **Tag Flakes**

  > Rappresentazione grafica basata sulle co-occorrenze, ha l'obiettivo di estrarre una gerarchia di termini correlati. Si crea una struttura che parte da un nodo root e si espande in più rami rappresentanti i topic estratti.

- **Document clustering**

  > Tecnica del raggruppamento di documenti in base alla loro similarità. I gruppi generati non sono predefiniti ma vengono generati solamente in base ai dati.

- **Document classification**

  > Classificazione di documenti in base al loro contenuto. A differenza del clustering, la classificazione è supervisionata, ovvero i gruppi sono predefiniti. La differenza fra clustering e classificazione è simile a quella tra WSI e WSD (rispettivamente), nel clustering e WSI non abbiamo conoscenza a priori delle rispettive categorie, mentre nella classificazione e WSD abbiamo già a disposizione le categorie da utilizzare.

- **Document segmentation (lab4)**

  > Questo task consiste nel separare un documento in aree semanticamente omogenee. L'algoritmo più famoso per questo task è il TextTiling.

- **Document summarization**

  > Task che consiste nel riassumere un documento in un testo (generalmente) più corto.

  Esistono due approcci:

  1. **Estrattivi**

     > Consistono nel selezionare le frasi più importanti (usando un valore di "salience") del documento e riportarle nel testo riassuntivo, un'algoritmo molto famoso è il TextRank.

  2. **Astrattivi**

     > Questi metodi sono generalmente molto più complessi, si basano sulla generazione di testo nuovo. Ultimamente le reti neurali hanno ottenuto ottimi risultati in questo task.

- **Information retrieval**

  > Task che ha lo scopo di trovare i documenti più rilevanti per una determinata query (set di keyword). Recentemente questo task è stato migliorato con l'uso di analisi sofisticate che tengono conto della semantica contestualizzata dei documenti e delle query.`

5. Semantica Distribuzionale (p. 51):

Tecnica che di NLP che, a differenza del Text Mining (che segue un approccio puramente statistico), si basa sull'uso del linguaggio e della linguistica.

Citazioni (p. 51):

> Harris: "Le parole che si verificano negli stessi contesti tendono ad avere significati simili"

> Firth: "Una parola è caratterizzata dalla compagnia che mantiene"

> Furnas "L'uso congiunto delle parole serve principalmente a specificare più strettamente l'oggetto del discorso"

> Deerwester: "C'è una struttura semantica latente sottostante nei dati che è parzialmente oscurata dalla casualità della scelta delle parole rispetto al loro recupero"

> Blei: "L'argomento del documento influenzerà in modo probabilistico la scelta delle parole dell'autore durante la scrittura del documento"

> Turney: "Coppie di parole che si verificano in modelli simili tendono ad avere relazioni semantiche simili"

Rappresentazione matriciale (p.53+):

Le matrici si collocano a metà fra due rappresentazioni:

- **Simbolica**

  > In una rappresentazione puramente simbolica i simboli a se stanti hanno un basso contenuto informativo. Tramite la rappresentazione vettoriale/matriciale possiamo associare un valore numerico a ciascun simbolo.

- **Associazionistica / Connessionistica**

  > Una rappresentazione puramente connessionistica è basata solamente sui collegamenti fra concetti, l'uso di vettori ci permette di mantenere questo collegamento fra più concetti (matrice) senza perdere la loro rappresentazione simbolica (vettore).

Tecniche applicabili ai dati:

La rappresentazione matriciale introduce anche diverse tecniche matematiche applicabili ai dati:

- **Similarità**

  > Possiamo applicare tecniche come cosine similiarity e jaccard similarity o strategie di pesatura come la TF-IDF.

- **Trasformazioni matriciali**

  > Possiamo applicare tecniche come SVD, NNMF, etc...

- **Clustering**

  > Possiamo applicare tecniche di raggruppamento come K-Means, EM, etc...

Pre-processing e post-processing:

Per poter utilizzare al meglio la rappresentazione matriciale è però necessario utilizzare tecniche di pre/post-processing dei dati, come:

- **Normalizzazione**

  > Necessario per restringere la variabilità del linguaggio tramite tecniche come lemmatizzazione, stemming, tokenizzazione, etc...

- **Denormalizzazione**

  > Tecnica inversa alla normalizzazione, serve ad arricchire il linguaggio tramite tecniche come la WSD, le named entities, etc...

Svantaggi e problemi:

- **Word order**

  > Attraverso l'uso della rappresentazione matriciale classica si perde il concetto di ordine delle parole, senza l'ordine si può raggiungere un massimo di circa 80% di espressività del contenuto. Per ovviare a questo problema sono state pensate soluzioni come matrici pair-pattern (più

> sensibili all'ordine) o l'uso di vettori ausiliari detti vettori di ordinamento (con lo scopo di codificare informazioni riguardo l'ordine)

- **Rappresentazione non composizionale**:

> La rappresentazione matriciale di base non è composizionale, si basa principalmente sul significato di singole parole. Per ovviare a questo problema si possono usare costrutti che combinanno più vettori per comporne il significato.

Configurazioni matriciali:

1. **Term-Document Matrix**

> - **Righe**: Documenti
> - **Colonne**: Termini
> - **Valori**: Frequenza o altre misure associate a ciascun termine nel documento
> - **Utilizzo**: Viene generalmente utilizzata al livello di granularità del documento, per scopi come: Similarità fra documenti, clustering, classificazione di documenti, segmentazione di documenti e in parte question answering.

2. **Term-Content Matrix**

> - **Righe**: Contesto (può essere un documento (simile a 1), un >paragrafo, una frase, etc...)
> - **Colonne**: Termini
> - **Valori**: Frequenza o altre misure associate a ciascun termine >nel contenuto
> - **Utilizzo**: Viene generalmente utilizzata al livello di granularità delle parole, per scopi come: Similarità fra parole, clustering e classificazione di parole, WSD, information extraction, etc...

3. **Pair-Pattern Matrix**:

> - **Righe**: Coppie di parole
> - **Colonne**: Pattern (relazioni fra le parole es: X causa Y, X è risolto da Y, etc...)
> - **Valori**: Peso associato alla specifica relazione (pattern) fra le due parole
> - **Utilizzo**: Utillizato per scopi come:
>   - *Relational similarity*: similarità fra coppie di parole (es: similarità fra x-y e w-z),
>   - *Pattern similarity*: clustering su pattern associati a coppie simili.

Similarity (p. 58):

Esistono diverse definizioni di similarità:

1. **Semantic similarity**

> Concetti che hanno (quasi) lo stesso significato, sinonimi. (es: "gatto" e "micio")

2. **Semantic relatedness**

> Superclasse di semantic similarity, comprende concetti che condividono delle proprietà (affinità semantica). Possono essere meronimi (parte di un concetto) o antonimi (opposti) e anche sinonimi. Questa tecnica è poco utilizzata perchè restituisce relazioni troppo generiche. (es: "ruota" e "camion")

3. **Attributional similarity**

> Identica a semantic relatedness, meno utilizzata. (es: "ruota" e "camion")

4. **Taxonomical similarity**

> Concetti che condividono degli iperonimi. Facilmente calcolabile. (es: "gatto" e "struzzo", entrambi iponimi di animale)

5. **Relational similarity**

> Relazione fra coppie di concetti. (es: le coppie "gatto | miagolio" e "cane | abbaio" sono in relazione)

6. **Semantic association**

> Simile alla semantic relatedness ma basata sulle co-occorrenze delle parole. Possono esserci parole che sono semanticamente relazionate ma che non co-occorrono mai, viceversa parole che non sono semanticamente relazionae ma che co-occorrono spesso. (es: "culla" e "neonato")`

6. Semantica Documentale (p. 63):

La semantica documentale è tutto ciò che riguarda l'analisi di collezioni di documenti. Fanno parte di questo campo:

- topic modelling
- dynamic topic modelling
- text visualization
- etc...

Topic modelling (p. 63):

Topic model

> Un topic model è un modello statistico o probabilistico che individua automaticamente i topic in una collezione di documenti. Il topic model NON è supervisionato, quindi non è necessario fornire alcun tipo di annotazione manuale dei dati.

Topic

> Un topic è rappresentato semplicemente da una lista pesata di parole. Il problema principale di questa tecnica è che l'interpretazione del topic è spesso manuale e non sempre ovvia. I topic model estraggono semplicemente la co-occorrenza fra termini, un topic potrebbe quindi essere semplicemente una coincidenza statistica.

Latent Semantic Analysis (LSA) (p. 64) (lab5):

La LSA è una tecnica di topic modelling che si avvale dell'uso della SVD per estrarre i topic.

Concetto latente

> Un concetto latente è una caratteristica o tema nascosto in un insieme di dati o informazioni. è un concetto sottostante che non può essere osservato direttamente, ma solo tramite l'uso di analisi e modelli appropriati.

Singular Value Decomposition (SVD):

Nel caso della LSA l'input per la SVD è una matrice contenente le frequenze normalizzate dei termini nei documenti (Term-Document Matrix). La SVD scompone la matrice M in tre matrici:

$$M = U\Sigma V^*$$

- **U**: `Questa matrice contiene una nuova rappresentazione dei documenti (stesse righe), le colonne sono però sostituite con delle nuove features che vengono chiamate "concetti latenti".`
- Σ: `La diagonale di questa matrice contiene i valori singolari, i valori restanti sono tutti 0.`
- V^*: `Questa matrice (similmente a U) contiene la rappresentazione delle features latenti, ma trasposte.`

La SVD analizza l'input e identifica automaticamente le ridondanze in esso. Essa cattura l'informazione di co-occorrenza di parole X con parole Y creando nuove features che le rappresentano.

Un vantaggio di questo sistema è che i concetti latenti sono ordinati, sarà quindi possibile troncare le matrici considerando solo le prime dimensioni (concetti) più importanti.

Vantaggi:

- Riduzione della dimensionalità: riducendo la dimensionalità si riduce anche il rumore e si aumenta la velocità di computazione.
- Riduzione della sparsità: la SVD permette di passare da una matrice sparsa ad una densa.

Svantaggi:

- Valori negativi: la SVD può produrre valori negativi, questi possono essere difficili da interpretare.
- Vettori non interpretabili: i vettori prodotti dalla SVD non sono interpretabili, a differenza della matrice di input, non è possibile capire cosa rappresentano.

Utilizzo **(vedi esempio p. 65-66)**

L'uso della SVD ci permette di produrre matrici che tengono conto del contesto indiretto. Ad esempio se abbiamo una collezione di documenti, la similarity classica calcolata solamente fra i documenti i e j potrebbe risultare nulla (erroneamente); la SVD a differenza di questo tiene conto di tutta la collezione di documenti (matrice) per calcolare i concetti latenti, se poi effettuiamo la similarity fra i e j, questa potrebbe essere diversa da 0 in quanto tiene conto di tutti i documenti (contesto indiretto).

Svantaggi LSA:

1. **Non generalizza su documenti non visti**

> se vogliamo aggiungere documenti alla collezione, dobbiamo ricalcolare la SVD.

2. **(SVD) i valori negativi sono difficili da interpretare**

> questo può essere risolto utilizzando la **NNMF** (Non-Negative Matrix Factorization).

P-LSA (Probabilistic LSA) e LDA (Latent Dirichlet Allocation):

La LDA (generalizzazione della P-LSA) sfrutta la statistica Bayesiana e si basa sull'assunzione che un documento è un mix di topics e che una parola ha una certa probabilità di comparire in ogni singolo topic. Questo permette, dato un insieme di parole, di dedurre dia il topic di appartenenza che altre parole potrebbero comparire nei documenti topic-related.

Dynamic Topic Modelling:

Il dynamic topic modelling è un sistema dove non solo vengono estratti i topic, ma essi vengono proiettati nel tempo in modo da visualizzare la loro evoluzione.

Text Visualization (p. 68):

1. **Parallel coordinates**
2. **Radial visualization**
3. **HeatMap**
4. **Correlation circle**

7. Text2Everything (p. 74):

8. Basicness (p. 89) (lab6):

Teoria che si basa sul fatto che le parole possono avere un livello di complessità differente fra di loro.

Vocabolario di base (Ogden):

Ogden definì il concetto di "vocabolario di base", un insieme di ~500 parole che dovrebbero costituire la base della comunicazione quotidiana. Secondo Ogden le parole "di base" devono essere:

- brevi
- relative a concetti concreti
- facili da pronunciare
- usate frequentemente

Middle level:

Un'altra definizione di termini di base è quella di "middle level", ovvero termini che non sono troppo specifici o troppo generici *(es: forchetta è più basico di posate, ma cane è più basico di barboncino)*.

Definizione di basicness:

(vedi p. 91 per esempi)

- Il nome più comunemente usato per un concetto.
- Le parole che portano più informazione, e che siano le più differenziabili fra di loro.
- Parole che sono corte e morfolologicamente semplici.
- Parole facili da imparare.
- parole che massimizzano la similarità intra-classe e minimizzano la similarità inter-classe.
- Le prime parole che vengono in mente pensando ad un concetto.
- Le parole isate più comunemente nella quotidianità.

Alcuni concetti relativi a basicness:

- **Legame termine concetto**

  > La basicità di un termine può cambiare in base al concetto che descrive *(es: "cane" può intendere sia l'animale, basic, che il componente di un'arma da fuoco, non basic)*. Inoltre non tutti i concetti sono equamente descrivibili in ogni lingua.

- **Soggettività**

  > La basicità di un termine è soggettiva, dipende dal contesto e dall'utente.

- **Advanced**

  > Un termine è "advanced" se non è "basic".

- **Applicazioni**

  > L'uso di termini basici è fondamentale nel campo dell'apprendimento, termini basici facilitano l'accesso alle informazioni e la comprensione di esse.

9. Ontology Learning e Open Information Extraction (p. 95):

Ontology Learning (p. 95):

Processo di estrazione automatica di dati strutturati (ontologia) da dati non strutturati (testo).

La prima definizione fu quella di Phillip Cimiano, che la definì come un processo di "reverse engineering", ovvero un processo che analizza dati e conoscenze esistenti e cerca di estrarne i concetti e le relazioni sottostanti.

Questo approccio ha alcune problematiche:

- Dato che la conoscenza che vogliamo analizzare non è codificata, essa è soggettiva.
- Non tutte le informazioni riguardo al dominio sono necessarie, quando si concettualizza un sistema si tiene conto soprattutto il suo utilizzo.

Sottospecificazioni (p. 96):

- **Ontology population**

> Si possiede un'ontologia già costruita, il task è quello di popolarla con istanza trovate nel testo.

- **Ontology Annotation**

> Si possiede un'ontologia già costruita, il task è quello di annotare il testo con le informazioni concettuali.

- **Ontology Enrichment**

> Si possiede un'ontologia già costruita, il task è quello di popolarla con istanze (come in ontology population) e inoltre di (potenzialmente) ristrutturare l'ontologia a livello di concetti (aggiungo/rimuovo nodi) e relazioni (aggiungo/rimuovo archi).

Gradi di formalizzazione (p. 96):

1. testo libero, non strutturato. "Document repository".
2. formalizzazione che contiene una serie di termini con associati i domini di interesse
3. formalizzazione che contiene una serie di termini e le loro definizioni
4. Thesaurus, fornisce una serie di relazioni fra i termini (es: wordnet). Questa formalizzazione può a sua volta variare di complessità, da relazioni come sinonimi, iponimi, etc..., a relazioni più complesse come "parte-di", "utile-per", etc...

Con l'aumentare della formalizzazione, aumenta la complessità degli approcci automatici.

Task (p. 97):

- **Term extraction**

> trovare nomi per i concetti e per le relazioni fra di essi.

- **Synonym extraction**

> trovare termini che hanno lo stesso significato in un certo contesto.

- **Concept extraction**:
  - **intensionale (gloss learning)**

> si cerca di rappresentare in stringa tutto ciò che può descrivere un determinato concetto. (es: "cane" -> "animale domestico, mammifero, etc...")

  - **estensionale**

> enumerare tutti gli elementi che descrivono un determinato concetto (es: "cane" -> [elenco di cani])

- **Concept hierarchy induction**: strutturare concetti già noti attraverso una tassonomia
- **Relation extraction**

> creare relazioni fra concetti già noti (simile alle Pair-Pattern matrix che seguono però un approccio distribuzionale)

- **Population**:
  - **NER**

    > popolare l'ontologia con relazioni instance-of tramite l'uso di Named Entity Recognition.

  - **IE**

    > popolare l'ontologia con relazioni instance-of tramite l'uso di Information Extraction. (es: riconoscere i ruoli semantici di alcune persone all'interno del testo e inserirli nell'ontologia)

- **Notazione di sussunzione**

  > Costruzione automatica di gerarchie di concetti. ("sussunzione" è un'operazione che permette di stabilire se un concetto è più generale di un altro)

Metodi (p. 98):

esistono tre tecniche per svolgere i task sopra elencati:

- NLP
- Matematiche (FCA)
- Machine Learning

NLP:

Approcci che sfruttano informazioni come POS, NER, alberi di parsing, risorse lessicali, informazioni statistiche, pre-processing, etc...

Formal Concept Analysis (FCA):

La FCA è una tecnica che sfrutta un approccio derivato dalla matematica formale modificato per la costruzione di gerarchie.

La FCA si basa su 3 concetti principali:

1. **oggetti**: sono l'equivalente dei concetti, istanze.
2. **attributi**: features (attributi) associate agli oggetti.
3. **incidenza**: relazione che rappresenta il fatto che un oggetto possiede un certo attributo.

La relazione fra oggetti e attributi può essere rappresentata tramite una matrice (binaria) di incidenza, chiamata "formal context".

	C1	**C2**	**C3**	**C4**
A1	x		x	
A2		x		x
A3			x	

Su questa matrice possono essere applicati due operatori:

- **UP(C)**: viene applicato ai concetti (colonne) e restituisce gli attributi che possiede (*es: UP(C3) = {A1, A3}*).
- **DOWN(A1, A2, ...)**: viene applicato agli attributi (righe) e restituisce i concetti che possiedono quell'attributo (se più di un parametro, restituisce i concetti che li possiedono TUTTI) (*es: DOWN(A1) = {C1, C3}*) (*es: DOWN(A1, A3) = {C3}*).

Machine Learning:

Tutte le tecniche di ML che vengono applicate al task di costruzione di strutture concettuali e ontologie.

Open Information Extraction (p. 101):

"Open" information extraction si riferisce a una serie di tecniche che mirano ad estrarre informazioni semi-strutturate da una grande quantità di testo (che sarebbe normalmente troppo costosa computazionalmente).

Generalmente le informazioni prodotte da queste tecniche sono sotto forma di triple del tipo [argomento1]-[espressione verbale]-[argomento2].

Vantaggi:

- Diminuzione dello spazio testuale rispetto a quello originale.
- Generazione di informazioni semi-strutturate su cui è possibile effettuare query (es:"restituisci gli argomenti che hanno un verbo V all’interno delle verbal phrase").
- é possibile ridurre la quantità di rumore prodotta applicando vincoli agli argomenti o alle VP.

Svantaggi

- Presenza di grandi quantità di rumore nei risultati.
- Esistono svariati approcci per l'estrazione delle triple, quindi queste sono disallineate fra i diversi sistemi.
- é difficile valutare questi sistemi in quanto non esiste un approccio gold standard.
- Non sempre è semplice applicare l'uso delle triple in contesti reali.

Sistemi famosi:

- **ReVerb**: uso di vincoli sintattici
- **KrankeN**: uso di uno step di WSD
- **ClausIE**: estrea non solo triple
- **DefIE**: combina parsing con WSD, crea un grafo pesato

10. LLM e Prompting (p. 103):