

Riassunto a opera di Gabriele Naretto e Persona Anonima 🙈

Questo è un riassunto e come tale prendetelo con le pinze, ci possono essere possibili errori di ortografia e informazioni sbagliate.

Introduzione

Vediamo i diversi tipi di semantica

- **Lessicale:** studio di *cosa e come denotano le parole* → quindi il lessico
- **Formale:** *modelli logico matematici che definiscono il linguaggio*
- **Statistica:** prettamente *numerica*
- **Linguistica-distribuzionale:** *studio della distribuzioni semantiche delle parole, il tutto guidato da un approccio statistico/linguistico*

Ogni argomento del corso fa parte di una delle seguenti semantiche.

In origine lo scopo del trattamento del linguaggio naturale era fare capire uomo e macchina, il tutto con il task del question answering (QA). Oggi non riguarda più il QA ma tutto quello che ci sta dietro. Ovvero molti argomenti che vedremo in questo corso.

ogni domanda ha uno specifico significato e va associata a diverse informazioni semantiche. Le informazioni semantiche possono essere:

- parole
- punteggiatura
- lemmi
- sinonimi
- sintassi
- senso della parola
- named entities
- ruolo semantico
- Iperonimo
- Antonimo
- Meronimo
- gloss
- relazione semantica specifica
- anaphora
- frequenza
- contesto

Significato del significato

Definizioni per capire i vari argomenti che tratteremo:

- **Lessico:** Elementi che servono per costruire una frase e fanno parte del dizionario
- **Sintassi:** studio delle relazioni tra i pezzi del dizionario
- **Semantica:** Corrisponde all'interpretazione di una struttura lessico-sintattica a cui si attribuisce un significato.

- **Pragmatica:** È una disciplina della linguistica che si occupa dell'uso contestuale della lingua ovvero di come il contesto influisca sull'interpretazione dei significati prescindendo dall'uso di lessico e sintassi.
- **Ambiguità:** Proprietà del linguaggio, permette una comunicazione efficiente
- **Polisemia:** Una parola può esprimere più significati
- **Omonimia:** Parole con la stessa forma ma significati diversi Comunicazione
Condivisione di significati che risiedono nella mente
- **Convenzione:** Veicolo del contenuto semantico tramite simboli (es. suoni)
- **Granularità:** Livello di dettaglio che si vuole descrivere
- **Soggettività:** Essendo il linguaggio un'approssimazione delle immagini della nostra mente si possono commettere errori durante l'esposizione
- **Similarità:** Meccanismo che permette l'inferenza di significato da altri termini conosciuti
- **Esperienza personale:** Insieme di eventi che forma il nostro modo di comunicare
- **Senso comune:** Convenzioni a livello collettivo
- **Cultura:** Convenzione storica/senso comune storici

Definendo tutti questi concetti/parole abbiamo creato un'ontologia, ovvero creato una base di conoscenza in cui si conosce in cui si condivide il significato.

Significato delle parole

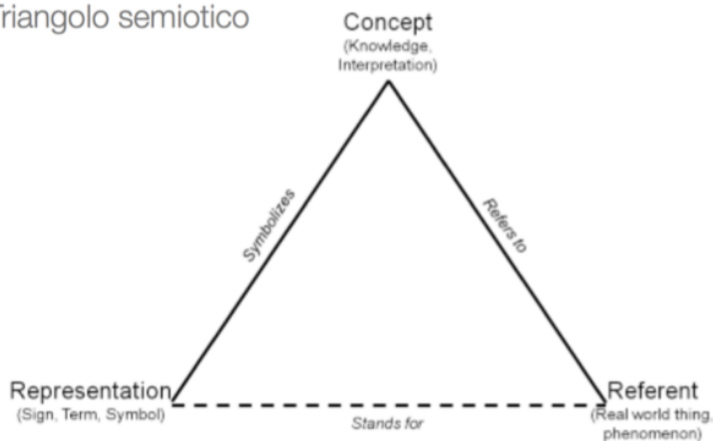
Teorie del significato

Abbiamo tre teorie sul possibile significato delle parole

- Basate su **primitive**: si divide la parola in concetti più piccoli e atomici, la parola è l'insieme di questi significati. La parola "Scrivania" è composta dal concetto "tavolo" che ci porta al concetto di "piano".
- Basate su **relazioni**: La parola ha un significato nel suo contesto, ovvero l'insieme di parole con cui ha una relazione. → questo implica che una parola non abbia in sé significato, ma sia il contesto in cui è posta e le parole con cui è relazionata che gliene danno.
- Basate su **composizione**: oltre al contesto, se mettiamo insieme più parole (le componiamo) queste cambieranno significato. Quindi composizioni della stessa parola gli danno un senso diverso.

Triangolo semiotico

Triangolo semiotico



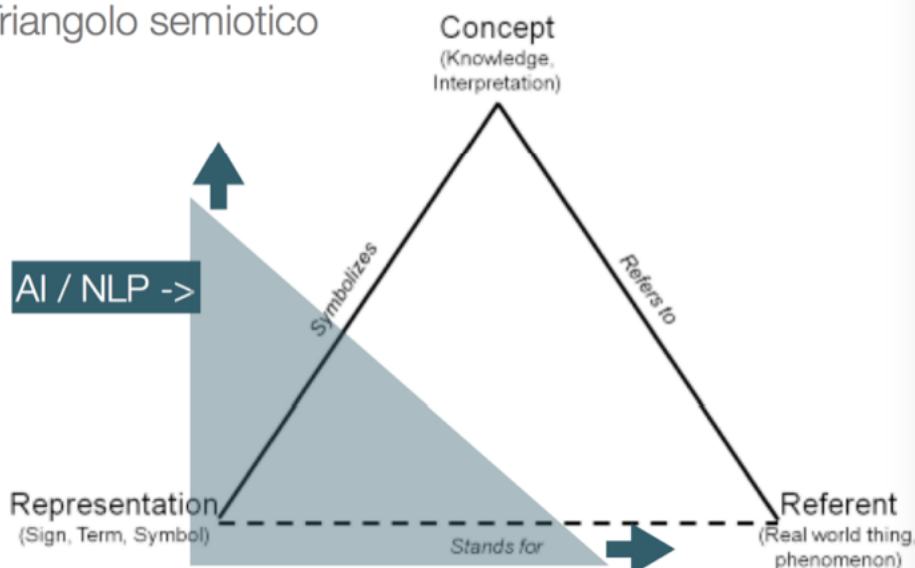
La parola prende il suo significato da tra componenti diverse:

- Il **concetto**: ovvero il concetto che abbiamo nella mente quando pensiamo alla parola, se pensiamo al gatto ci viene in mente un gatto, quello è il concetto.
- La **rappresentazione**: ovvero come viene rappresentato il concetto a cui pensiamo? un gatto in italiano viene rappresentato dalla parola “gatto” in inglese dalla parola “cat”
- Il **referente**: un'istanza del concetto

Le AI possono usare il triangolo semiotico?

Le AI non possono avere un concetto come noi lo intendiamo quindi partiranno per forza dalla rappresentazione → le macchine possono rappresentare a modo loro i dati in base al task. Quando parliamo invece del referente ci viene in aiuto la computer vision → ovvero con le recenti tecniche le macchine possono interpretare immagini riconoscendo le istanze di una classe (e quindi il referente)

Triangolo semiotico



Riflessione: Lavorando con questo triangolo semiotico ci accorgiamo che le IA non possiamo rappresentare i concetti (o almeno i concetti sono più vicini a delle astrazioni/concettualizzazioni come quelle viste come radici) → quindi la risorsa

ConceptNet ha un nome “sbagliato” perché appunto le AI non possono rappresentare i concetti. ma rappresentano la conoscenza di senso comune

Multilinguismo

La sfida che ha l’NLP nel trattare più lingue è vista come una sfida ma anche come un’opportunità: un testo in dieci lingue, invece di essere un problema, può essere studiato semanticamente in maniera più ricca attraverso ad es. strumenti indipendenti dal linguaggio, o attraverso l’allineamento delle varie versioni testuali.

Analizzare semanticamente un testo in più lingue in maniera indipendente permette di allineare ed integrare i risultati estrattivi tra di loro per:

- capire quali sono le informazioni semantiche più importanti, più certe o maggiormente condivise;
- migrare informazioni semantiche ritrovate su una lingua su di un’altra lingua.

Altro aspetto rilevante è la rarità di una lingua. Quando si ha un testo in una lingua rara il processamento può diventare molto complesso e costituisce da sempre un problema. Un ultimo problema da tenere in considerazione raccoglie tutte quelle sfumature di una lingua comprendenti modi di dire, usanze e convenzioni. Si tratta di concetti diversi che non sono lessicalizzati in tutte le lingue: come per esempio i concetti italiani di “Boh”, “Mamma mia” che non hanno diretti referenti in inglese.

Granularità

Possiamo vedere il testo sotto diverse profondità:

- **Parola** → task word sense disambiguation
- **Chunk** → composizione di parole, task di multi word expressions
- **Frase** → task di question answering
- **Discorso** → task conversazionali come i chatbot
- **Documento** → task di summarization
- **Collezione di documenti** → task di topic modelling, text classification, text clustering

Noteremo a breve che in base al tipo di profondità avremo dei task differenti che si collegano con i prossimi argomenti

Word sense disambiguation (WSD)

Abbiamo una parola con molti significati(polisemica), il task è capire a quale senso ci riferiamo. Per questo task si possono usare dizionari online come wordnet.

Le problematiche del task sono:

- **Specificità:** spesso risorse come wordnet hanno delle parole in cui ci sono troppi sensi, sensi che per noi sono lo stesso vengono divisi a causa di piccole differenze.
- **Copertura:** wordnet non è omogeneo, ci sono delle zone in cui si hanno meno termini mappati, o addirittura zone non coperte
- **Soggettività:** per quanto wordnet sia opera di più persone usa definizioni non oggettive

Nota: in verità queste non sono proprio problematiche del task ma problematiche di wordnet, quindi della risorsa, il punto forse è che le problematiche delle risorse per svolgere questo task sono sempre le stesse ?

Word sense Induction (WSI)

La Word Sense Induction (WSI) riguarda l'identificazione automatica *dei sensi* di una parola dato il suo uso in tantissimi testi, catturando tutti quelli che sono i contesti che attribuiscono un senso preciso e differente dagli altri alla parola.

Quindi si va a identificare i diversi sensi/definizioni della parola dai testi.

Le principali differenze con il word sense disambiguation sono:

- WSD deve disambiguare e per farlo usa un dizionario (abbiamo tanti senti e dobbiamo capire qual'è quello giusto), mentre in WSI non usa un dizionario però deve riuscire a distinguere sensi diversi da un testo (abbiamo uno o più testi e vogliamo trovare tutti i sensi di una parola) → in un certo senso è come se in WSI volessimo costruire un dizionario di sensi di una parola
- WSD si basa sulla grammatica, WSI si basa sulle parole (anche sgrammaticate)
- WSD è più facile capire se il task è andato a buon fine, ma a causa del problema della specificità potremmo avere degli errori. WSI non ha un dizionario su cui basarsi quindi non è possibile verificare con cura se il task ha dato esito negativo.

Pseudo Word

Metodo di valutazione di WSI, prendi due parole (**a** e **b**) e si uniscono (creando una parola nuova **c**), nel corpus si sostituisce **c** con tutte le occorrenze di **a** e **b**, e si valutano i cluster che ne escono. (dovrebbero uscire due cluster, uno per la parola **a** e uno per la parola **b**)

Il procedimento in passi è il seguente:

- Merging delle parole → concatenazione delle parole
- Clustering → meccanismi della WSI per identificazione dei cluster
- cluster-to-class → confronto tra i cluster

Riflessione: ma se una delle parole è polisemica ? non si rischierebbe di creare più cluster semantici ? già solo il task di WSI può creare molti cluster per una sola parola




Definizione delle definizioni, ricerca onomasiologica e genus

Una domanda che ci poniamo è “come si definisce una definizione?” oppure “come si descrive un concetto?” o anche “Quali caratteristiche sono più importanti di un concetto?”

Esistono delle risorse online che provano a dare delle definizioni e lavorano su queste domande, vediamo alcune:

- **Dizionari elettronici:** Wordnet e Babble Net
- **Risorse linguistico cognitive:** Property Norms → questionari compilati da più persone in cui vengono salvate le risposte/descrizioni. Le risposte vengono poi aggregate e raccolte in grandi dataset, *queste risorse sono molto promettenti e forniscono statistiche di quali sono le parole più frequenti che sono comparse nelle varie descrizioni → riusciamo a catturare la percezione e l'immediatezza*
- **Common-sense knowledge:** ConceptNet → *descrivono conoscenze di senso comune*, spesso contengono informazioni non disambiguate.
- **Visual Attributes:** si basano sulle caratteristiche/fisionomia degli oggetti → contengono variabilità linguistica, raccolgono tutte le caratteristiche di oggetti

osservabili e possono essere rappresentati come una tassonomia

	behavior diet shape_size anatomy color_patterns	eats, walks, climbs, swims, runs drinks_water, eats_anything is_tall, is_large has_mouth, has_head, has_nose, has_tail, has_claws, has_jaws, has_neck, has_snout, has_feet, has_tongue is_black, is_brown, is_white
	botany color_patterns shape_size texture_material	has_skin, has_seeds, has_stem, has_leaves, has_pulp purple, white, green, has_green_top is_oval, is_long is_shiny
	behavior parts texture_material color_patterns	rolls has_step_through_frame, has_fork, has_2_wheels, has_chain, has_pedals has_gears, has_handlebar, has_bell, has_breaks, has_seat, has_spokes made_of_metal different_colors, is_black, is_red, is_grey, is_silver

- **Corpus manager:** gestiscono corpus a scopo di ricerca un esempio è sketch engine

Una cosa comica di queste risorse è che usano il concetto di definizione, ovvero usare delle parole per definire altre parole. Ma questo fatto evidenzia delle domande come “come si descrive un concetto?” oppure “quali sono le caratteristiche più importanti?” ecc

La ricerca onomasiologica si cerca dalla definizione, il termine. → *Quanto è “complesso” identificare un concetto data una sua definizione?* (quindi il concetto è il senso della parola visto questa frase?)

C'è da fare attenzione al problema della **circolarità indiretta**: quando vogliamo definire X che ha come attributi (proprietà) y, w e z, e w ha altri attributi nella sua definizione può essere un problema. Si cerca di usare ricorsivamente il termine per spiegare il termine.

Il **genus differentia** è la categoria più vicina/generica che racchiude il concetto (come iperonimo).

Riflessione: a questo punto sottolineo che il professore non ha dato proprio una definizione di concetto.

Il triangolo semiotico vuole costruire il significato d'una parola, e per farlo usa il concetto, ovvero una il concetto mentale che ci viene in mente quando pensiamo a una parola → se pensiamo alla parola gatto mi viene in mente un gatto → probabilmente concettualizziamo il gatto ovvero lo generalizziamo, aggregando gli aspetti più caratteristici di tutti i gatti e creiamo un'immagine di gatto (come la teoria dei prototipi o esemplari vista con radici).

Ma quando parliamo di ricerca onomasiologica lui usa il termine “concetto” come se intendesse il senso della parola → quindi il senso di una parola per il professore è il concetto che vuole rappresentare? → il tutto rappresentato da una definizione?

Costruzione del significato

Si parlerà di teorie per la costruzione del significato → questa parte deriva dalla **semantica lessicale**.

Abbiamo principalmente due teorie:

Pustejovsky

La prima teoria che tratteremo, si basa sulla struttura “generative lexicon”. E sull’uso di diverse componenti/parti:

- **Argument Structure:** struttura che esprime il legame tra sintassi e semantica del concetto ovvero come si può mappare quello che si vuole esprimere su quel concetto attraverso l’uso di lettere, parole e grammatica.
- **Event Structure:** struttura che esprime tutti i tipi di eventi che coinvolgono quel concetto come lo stato, il processo o la transizione.
- **Qualia Structure:** esprime la struttura del concetto ovvero come sono definite le sue caratteristiche. Queste caratteristiche vengono chiamate da Pustejovsky qualia.
- **Inheritance (eredità) Structure:** struttura che colloca il concetto all’interno di una tassonomia per poterne discernere a grandi linee già il significato (es. Se ho la parola “mango” devo sapere che è un frutto).

Qualia Structure

Ci concentriamo adesso maggiormente sulla struttura più interessante che è data dalla Qualia Structure. Secondo Pustejovsky esistono infatti quattro ruoli Qualia:

- **Ruolo costitutivo** → esprime la parte di composizione del concetto quindi è il ruolo più materiale che riguarda il peso, la dimensione e le parti che compongono il concetto.
- **Ruolo formale** → esprime tutte quelle caratteristiche che definiscono il concetto e lo distinguono dagli altri all’interno dello stesso dominio (es. Se parlo del mango il suo ruolo formale è dato da tutte le caratteristiche che lo contraddistinguono dagli altri frutti)
- **Ruolo telico** → ciò che rappresenta l’obiettivo o la funzione del concetto, sul ruolo comportamentale del concetto (es. Il cane che abbaia)
- **Ruolo ‘agentive’** → composto da tutta quella serie di entità spesso umane, ma che possono essere anche artificiali o eventi naturali che rappresentano l’origine del concetto.

Si assegna quindi a ogni elemento un ruolo e una struttura in base ai concetti definiti da Pustejovsky → Successivamente, ogni frase o asserzione che contiene uno di questi concetti può essere analizzata in modo formale attraverso un modello semantico basato sul lessico, utilizzando ragionamenti condizionali o relazionali rispetto ad altri concetti. E però una teoria molto complessa da applicare e implementare.

Hanks

Ha creato la **teoria delle valenze** per la costruzione del significato.

Questa teoria si basa sul fatto che il verbo sia la radice del significato, non esistono quindi delle espressioni di significato senza verbo. I sostantivi non hanno in sé significato ma servono da incastro per dare significato al verbo.

Ogni verbo ha una valenza, ovvero la cardinalità degli argomenti che compongono la struttura di cui il verbo è la radice, possiamo rappresentarlo un po come una n-upla in cui avremo n argomenti in base al verbo, “arg1 verb arg2” → in base al tipo di verbo (transitivo,

intransitivo ecc) il numero di argomenti aumenta o diminuisce. Il verbo *piove* non ha argomenti, mentre *portare* ha 3 argomenti.

Hanks afferma che in base al numero di argomenti utilizzati si differenzia il significato.

La valenza ovvero gli argomenti possono essere anche detti/visti come slot.

Ci sono due concetti che si legano a quello di valenza:

- **Collocazione:** vengono rappresentate tutte le possibili combinazioni dei filler. quindi dato un verbo con due filler, prendiamo tutti le parole del primo filler e tutte le parole del secondo filler e creiamo tutte le possibili combinazioni.
- **Semantic Type:** si vanno a raggruppare i filler nelle loro macro categorie. le macro categorie possono essere *persona, luogo, oggetto*, ecc

A questo punto, una volta che si collezionano tutti i filler e si raggruppano per Semantic Type, Hanks dice che ogni significato costruito dipende dalla combinazione dei Semantic Type degli argomenti. Quindi se ho il Semantic Type di tipo 1 per il primo slot e quello di tipo 3 per il secondo slot, allora questa combinazione per quel verbo rappresenta un significato preciso → quindi ripetendo **Secondo hanks le possibili combinazioni di semantic types rappresentano un senso preciso.**

Le problematiche di questa teoria sono molteplici:

- Quali sono i semantics type ?
- Qual è il loro grado di generalizzazione ?
- Ci sono parole poco frequenti e quindi per queste abbiamo pochi dati → questo pone una difficoltà di analisi di queste parole
- **I termini si riferiscono a concetti ad un certo livello di generalizzazione dipendentemente dal contesto** → spesso la generalizzazione dipende dal contesto → Vediamo un esempio e consideriamo la frase: "The student went to school". Quale può essere il Semantic Type per "student" con il verbo "went"? Quali proprietà di quel soggetto sono attivate da quel contesto? È uno Student? È un Person? O è un Living Entity? Molto probabilmente il contesto non è così astratto da considerare una tassonomia così alta come l'essere vivente, magari potrebbe bastare il supersense Person o magari in un contesto puramente scolastico basterebbe Student.

Affordance Linguistiche

Un modo per risolvere le problematiche che si hanno nelle teorie di hanks sono le affordance linguistiche.

Iniziamo però spiegando cos'è un affordance linguistica → L'affordance è un termine che serve a indicare un oggetto che anche se non si è mai visto prima è possibile capire come utilizzarlo in quanto l'oggetto in sé fornisce dei suggerimenti per l'uso.

Questa teoria può essere usata in ambito del linguaggio → ovvero ogni parola ha un suo utilizzo di base che può essere intuito senza doverlo spiegare, come per gli oggetti.

un esempio è "io ho visto un chiora nel bosco" (*la parola chiora non esiste l'ho scelta a caso*) nonostante non sappiamo che cosa sia la chiora sappiamo che si può vedere, quindi è un oggetto concreto, se aggiungiamo alla frase "ho visto una chiora nel bosco,era molto veloce" capiamo che oltre a essere un oggetto concreto capiamo che si può muovere.

Questo serve per dire che il contesto descrive le proprietà dei concetti che non conosciamo e ci permette, almeno parzialmente, di capirli. Il contesto crea un'associazione tra le parole e le proprietà. questo ci porta potenzialmente a usare una parola anche senza conoscerne il significato. **Quindi il significato che diamo al concetto è dato dalle proprietà che queste hanno e le proprietà possono essere conosciute o assunte (come nel caso dell'esempio di prima).** Questo concetto si lega alla teoria di Hanks e dei semantics type, poiché il semantics type non può essere lo stesso, ma cambia in base al contesto in cui ci troviamo.

Nota/Svarione personale: quello che fa Hanks è quindi trovare dei pattern linguistici utilizzando il verbo come centro ? → nella sezione dell'Open information extraction viene vengono estratte delle tuple formate da *arg1 verb arg2*, quindi una sorta di pattern linguistico del testo. Alla fine quelli non sono dei pattern per estrarre informazione dal testo ? Alla fine anche le affordance non fanno la stessa cosa ? Mettiamo di avere una frase "ho visto un lupo nella foresta" questa sarebbe traducibile in un Pattern/Tuple "Soggetto vede Lupo" e "Lupo in Foresta" alla fine queste tuple si basano su argomenti e verbi e sono anche queste riconducibili all'open information extraction e alle matrici Pair Pattern e quindi ci si può fare inferenza che è quello che l'uomo fa → avendo queste patter io possono trovare delle similarità e quindi andare a fare inferenza su parole sconosciute.

La potenza generativa dei pattern

Supponiamo di avere a disposizione un corpus e dei pattern, dove per pattern si intendono delle frasi che all'interno contengono dei jolly (*).

Es → "Lorem * dolor sit *, consectetur adipiscing *"

Per linguistic instances si intendono tutte le occorrenze di quel pattern in un corpus, in quanto le parole all'interno del corpus faranno match con il pattern creando delle istanze complete → le parole andranno al posto degli asterischi e creeranno una frase completa.

Ovviamente possono esserci più parole che fanno match con un solo asterisco e quelle saranno dunque i filler per quel jolly. (un po' come la teoria di hanks, in cui abbiamo più filler per lo slot/argomento/valenza di un verbo)

A questo punto, se si potessero associare tutte le proprietà a quei concetti che ricoprono l'asterisco, allora sarebbe possibile raggrupparle attraverso concettualizzazioni: questo significa che se si andrebbero a legare le co-occorrenze degli elementi per ciascun jolly, creando cluster. Quindi andiamo a creare dei cluster concettuali in cui gli elementi all'interno del cluster hanno le stesse proprietà.

Questo è un po' quello che fa Hanks, ma questa volta non viene fatto sulle parole ma sulle loro proprietà.

Questi cluster semantici non sono lessicalizzati, ma esprimono le proprietà che quel jolly deve avere per ottenere una frase di senso compiuto in quel contesto. **Quindi il potere espressivo e generativo di un modello del genere diventa più potente di quello presentato da Hanks perché è possibile inserire qualsiasi parola.**

Se si hanno degli overlap tra le proprietà in due frasi diverse, allora è possibile dedurre una similarità semantica tra le due frasi. → (ci possiamo collegare alla similarità relatedness)

I dati e le proprietà possono essere ricavati da risorse linguistiche come WordNet, FrameNet o VerbNet.

Altre fonti per le proprietà possono essere:

- Questionari, indagini, studi cognitivi
- Proprietà latenti derivanti da similarità
- Ulteriori risorse linguistiche per i dati possono essere:
- Corpora (Wikipedia, etc.)
- Annotazioni manuali
- Open Information Extraction
- Hackathons
- Machine Learning

Oltre alla potenza generativa, un altro vantaggio dei pattern non lessicalizzati è costituito dal fatto che, a differenza dell'approccio lessicalizzato di Hanks, non c'è bisogno di grandi quantità di dati in input perché l'andamento delle proprietà sui termini segue una forma logaritmica. Quindi all'inizio ci sarà l'aggiunta di molte proprietà, ma arrivati ad un certo punto, anche se si continuano ad aggiungere termini non ci saranno più nuove proprietà.

Text Mining

Si tratta di tecniche di data mining su testo.

Approccio lessicale statistico → ovvero un approccio bottom up che si concentra sull'analisi qualitativa e quantitativa di fenomeni specifici per effettuare inferenze in modo automatico

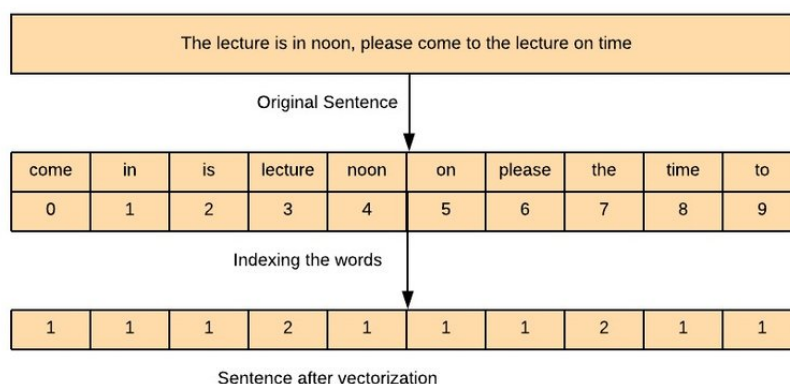
Rappresentazione Vettoriale

Quando parliamo di rappresentazione vettoriale intendiamo che le parole sono token ovvero una sequenza di caratteri senza nessuna particolare valenza a livello lessicale.

Un testo è quindi un insieme di token che possono avere una frequenza all'interno di un testo.

Gli approcci per trattare questi token sono diversi:

- Count Vectorized → vettori grandi quanto il numero di parole del documento che contano evidenziano la frequenza delle parole nel testo.



Il più importante lato positivo di questi vettori è che riescono a rappresentare interi insiemi di documenti con pochi vettori. Inoltre questi vettori possono essere confrontati tramite la

funzione "**cosine similarity**" → ovvero il prodotto degli elementi di una coppia di vettori diviso le norme dei due vettori. La formula della cosine similarity è:

$$\frac{A * B}{||A|| * ||B||}$$

quindi al numeratore avremo $(a_1 * b_1) + ... + (a_n * b_n)$
e al denominatore $norma(A) * norma(B)$

Metodi Statistici

- **TFIDF** → vettore che moltiplica Term Frequency (ovvero la frequenza nel testo) con L'inverse Document Frequency (che valuta quanto una parola appare nei vari documenti). La formula è la seguente:

$$w_{x,y} = tf_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

- **Co-Occorrenze** → assumendo che due parole con simile significato siano presenti negli stessi contesti, le co-occorrenze vengono rappresentate come una matrice diagonale, che corrisponde alla matrice per la matrice stessa, visivamente la rappresentiamo come:

-	apples	are	green	and	red	sweet	oranges	sour
apples	2	2	1	1	2	1	0	0
are	2	3	1	1	2	1	1	1
green	1	1	2	1	1	0	1	1
and	1	1	1	1	1	0	0	0
red	2	2	1	1	2	1	0	0
sweet	1	1	0	0	1	1	0	0
oranges	0	1	1	0	0	0	1	1
sour	0	1	1	0	0	0	1	1

Questo metodo permette di calcolare una cosine similarity molto più performante in termini di risultati. Si può per esempio cogliere la correlazione fra gatto e micio cosa che in un tf-idf o count vec non si potrebbe fare.

ordinamento sulla frequenza e poi si estrae una denominazione del topic a cui si riferiscono, creando una sorta di raggio di diversità semantica basata sulla similarità reciproca dei termini.

Potrebbe quindi essere possibile creare delle tag cloud disposte in maniera gerarchica per organizzare le informazioni su dei topic: una volta estratti automaticamente i topic, si organizza la tag cloud in base ad essi, creando di fatto una struttura a sub-topic

Da Google bard

Una tag flake è una tecnica di visualizzazione utilizzata nell'estrazione di testo per rappresentare le relazioni gerarchiche tra tag. È un tipo di nuvola di tag, ma invece di visualizzare semplicemente i tag in una nuvola, le tag flake mostrano i tag come una gerarchia, con i tag più importanti in alto e i tag meno importanti in basso.

Le tag flake possono essere utilizzate per aiutare gli utenti a comprendere il contenuto di una raccolta di testi. Mostrando le relazioni gerarchiche tra i tag, le tag flake possono aiutare gli utenti a vedere come i diversi tag sono correlati tra loro e come contribuiscono al significato complessivo della raccolta.

Le tag flake possono essere utilizzate anche per aiutare gli utenti a esplorare una raccolta di testi. Facendo clic su un tag in una tag flake, gli utenti possono vedere tutti i documenti che sono stati taggati con quel tag. Questo può aiutare gli utenti a trovare documenti pertinenti ai loro interessi.

Document Clustering

Metodo di machine learning che consiste in apprendimento NON supervisionato. il tutto svolto su documenti. I concetti base dati dal prof sono:

- Non esiste il clustering perfetto
- Non esiste una buona misura per il clustering

Document Classification

Metodo di machine learning SUPERVISIONATO.

Si vuole assegnare la giusta etichetta/classe al documento.

Document Segmentation

Una sorta di **Topic Modeling Intra documento**. Si vanno quindi a cercare in un documento dei punti in cui il contesto cambia e si può dividere il testo, lo scopo dell'esercitazione data dal prof è quello di prendere tre parti di testi unite in uno solo e trovare i punti di taglio che dividono i documenti.

Text Tilling

Un algoritmo molto usato è quello del Text Tilling, i passaggi dell'algoritmo sono i seguenti:

- **Separazione**: divisione del testo generando i tagli
- **Calcolo** della coesione intra gruppo
- **Ricerca** di punti di testo a bassa coesione circondati da punti di testo ad alta coesione, detti breaks point
- **Riadattamento** dei punti di taglio

Questi punti vanno eseguiti in modo iterativo finché non si trovano gli stessi punti due volte

Document Summarization

Task che comporta la riduzione del testo ma mantenendo la semantica contenuta al suo interno, quindi stesso contenuto ma in meno parole.

Abbiamo due tipi di summarization:

- **Estrattivo** → si estraggono le frasi che hanno più salience (misura che calcola l'importanza) e si riassume il testo
- **Astrattivo** → si riassume il testo generando nuovo testo che ha lo stesso significato. Questo metodo è migliorato molto grazie a reti neurali e transformers

Per misurare la qualità di un riassunto si usa la **ROUGE** → che usa bi-grammi e trigrammi del riassunto per paragonarli ai bi-grammi e trigrammi del testo originale.

Information Retrieval

Task che si basa sul recupero di documenti interessanti usando delle query (set di keyword). Possono essere relazionati a concetti ontologici o altri tipi di metadato.

Si effettua un'analisi sofisticata che cattura la semantica contestualizzata delle parole nella query e la semantica generale dei documenti, prescindendo dalla sovrapposizione lessicale.

I possibili sviluppi sono:

- Navigazione aumentata
- Integrazioni di Immagini, video e mappe
- Utilizzo di modelli avanzati di interazione come per esempio chatbot e risposte pre-generate

Semantica distribuzionale

Questo tema fa parte della linguistica computazionale legata alla linguistica distribuzionale.

Di base stiamo usando tecniche di text mining ma in chiave linguistica

Perché usare le matrici

Nella semantica distribuzionale come in molte tecniche di Text Mining vengono usate delle matrici per diverse rappresentazioni e compiti. La potenza delle matrici deriva dal fatto che stanno a metà fra due rappresentazioni:

- **Rappresentazione simbolica:** Rappresentazione che usa simboli, questi simboli ci permettono di fare inferenza, il problema è che i simboli sono poveri di significato. Ma se invece usiamo i vettori riusciamo a dare un significato numerico a questi simboli.
- **Rappresentazione associazionistica/connessionista:** teoria che pensa che tutti siano associati con tutto, come una rete semantica. Il learning si effettua con l'apprendere i pesi delle connessioni.

La matrice si mette a metà di queste rappresentazioni, le matrici facilitano la condivisione della conoscenza ed il significato diventa una regione geometrica. Si collegano inoltre alle teorie del conceptual space e del prototype model (entrambe viste con radici)

Che tecniche, misure si usano

Usando rappresentazioni matriciali e vettoriali abbiamo diverse tecniche che possiamo usare

- **Similarità** → capire quanto sono simili due vettori → cosine similarity
- **Trasformazioni matriciali**

- **Metodi che raggruppano i dati** → clustering (come K-means)
- **Tecniche di pre-Processing** → tra cui abbiamo:
 - **Normalizzazione** → tokenizzazione, stemming, lemmatizzazione, **restringiamo la variabilità del linguaggio**
 - **Denormalizzazione** → arricchimento semantico usando names entities, semantic roles e tecniche che associano un etichetta semantica.

Configurazione matriciale

Secondo Turney esistono 3 configurazioni matriciali:

- **Term Document Matrix** (termine-documento): matrice in cui avremo i documenti nelle righe e le parole nelle colonne, questa la usiamo per:
 - Similarità fra documenti
 - Clustering di documenti
 - Classificazione di documenti
 - Segmentazione di documenti
 - Parzialmente Question Answering, divisa in
 - analisi della domanda
 - recupero documento che contiene risposta
 - recupero frase che contiene risposta
 - estrazione risposta
- **Term Content Matrix** (termine-contesto): su ogni riga abbiamo un contesto e su ogni colonna un termine, un contesto non è per forza un documento ma può essere una frase, un paragrafo o una dipendenza sintattica. Si usa per
 - similarità fra parole
 - clustering tra parole
 - classificazione di parole
 - Word sense disambiguation
 - Information extraction
- **Pair Pattern Matrix** (coppie-pattern): su ogni riga abbiamo le coppie di parole mentre sulle colonne si ha un pattern. I pattern mettono insieme le parole in base a “X risolve Y” o “X causa Y” ecc che vengono enumerate in base al peso associato alla specifica relazione → che viene salvato nello spazio vettoriale della matrice. Usiamo le coppie di parole per motivi computazionali, se usassimo le triple diventerebbe intrattabile. Si usa per:
 - Relation similarity → similarità tra coppie
 - Pattern similarity → cluster su pattern che legano coppie simili
 - Relational Clustering
 - Relational Classification
 - Relational search → “individua la lista di tutte le X tali che X causa il cancro” → non si cerca un documento che parla di X ma tutti i concetti che sono in relazione con X

Ruolo della similarità

La similarità è fondamentale, tanto che spesso la semantica distribuzionale viene anche chiamata informalmente semantica della similarità.

Le scienze cognitive credono che l'uomo usi stime di similarità ogni giorno, **poiché ogni cosa che viene usata, imparata o vista viene ricondotta a concetti visti in precedenza per le loro similitudini.** → ci si può ricollegare alle teorie di hanks e agli affordance linguistici, ovvero capire il significato di oggetti in base ai verbi e a concetti pregressi.

Ricapitolando: similarità usata dall'uomo per ricondurre pratiche sconosciute ad attività conosciute → questo si collega agli affordance linguistici ovvero il derivare proprietà di oggetti in base a esperienze pregresse → questo si riconduce a hanks e al fatto che i semantic type dipendono dal contesto

Sono stati definiti diversi tipi di similarità:

- **Semantic Similarity:** si lavora sul significato, **simili a livello di significato ovvero sinonimi**
- **Semantic Relatedness** (Correlazione): **Intende concetti che condividono delle proprietà, che hanno una qualche affinità semantica.** Possono essere meronimi, antonimi ma anche sinonimi. *Ciò che è semanticamente simile è anche semanticamente relazionato ma non il contrario.* Questo tipo di semantica è quasi inutilizzabile perché restituisce come output il fatto che due concetti siano relazionati ma non specifica il motivo per cui lo sono
- **Attributional similarity:** Questo tipo di similarità intende concetti che condividono degli attributi (appunto delle proprietà). In letteratura comunque si preferisce utilizzare il nome semantic relatedness;
- **Taxonomic similarity:** riguarda i concetti che condividono gli iperonimi
- **Relational Similarity:** lavora su coppie tuple dei concetti
- **Semantic Association:** utilizzato nelle scienze cognitive piuttosto che NLP. tratta concetti che co-occorrono frequentemente. E' molto simile alla relatedness ma la differenza sostanziale sta nel fatto che la semantic association è relativa alle co-occorrenze di concetti. è orientata alla corpus analysis.

Ripetiamo quindi che il cardine di questa Semantica distribuzionale è strettamente correlata alla similarità e che però questo è il suo grande problema, visto che la similarità in molti ambiti tende essere soggettiva e quindi un punteggio può variare in base alla persona.

Problemi di Ordinamento

Le matrici non tengono conto dell'ordine e quindi recenti studi mostrano che il massimo di accuratezza raggiungibile è 80%.

Da un punto di vista di information retrieval (reperimento delle informazioni) questo però è un buon traguardo perché mostra che questa rappresentazione funziona indipendentemente dall'ordine. Ovviamente non si capirebbero tutte le parole o le frasi ma basterebbe capire la maggior parte di esse per riuscire ad estrapolare il concetto.

Per risolvere questo problema si pensava per esempio a delle matrici pair-pattern che sono più sensibili all'ordine

Problema di Rappresentazione matriciale non compositazionale

La rappresentazione matriciale non è compositazionale → le rappresentazioni matriciali sono piuttosto orientate al significato di singole parole.

Il linguaggio o la semantica sono invece composizionali → la composizione di parole crea nuovo significato che può essere anche diverso dal significato singolo delle parole.

La possibile soluzione che viene spesso adottata è la combinazione di vettori per creare costrutti più complessi e quindi per generare nuovo significato. Ad esempio, si usa il vettore di "vino" e lo si combina con quello di "rosso" per creare un nuovo vettore che approssimi il significato di "vino rosso".

Semantica documentale

La semantica documentale riguarda tutto quel tipo di analisi e di ricerca che si effettua al livello di collezione di documenti.

Topic Modelling

Modello statistico o probabilistico che analizza l'uso del linguaggio ed individua automaticamente gli argomenti (Topic) di una collezione di testi.

Modello **NON Supervisionato**.

un topic è rappresentato come una lista di parole che appartengono a quella categoria, abbiamo inoltre una grande problematica → ovvero che non sempre l'interpretabilità di questi topic è ovvia. I topic che vengono estratti, dipendentemente dalla tecnica di topic modeling utilizzata, misurano quanto i termini vengono usati negli stessi contesti. Quindi è possibile che i topic estratti non siano utili, e che rappresentino una semplice coincidenza statisticamente significativa.

Latent Semantic Analysis

La prima tecnica di Topic Modeling è stata la Latent Semantic Analysis.

Che **usa una fattorizzazione matriciale detta Singular Value Decomposition (SVD)**.

Questa tecnica prende in input le matrici contenenti le frequenze normalizzate (quindi i count vectorized normalizzati ? oppure matrici term document ?) e crea in output tre matrici.

1. **Matrice 1** → rappresentazione multidimensionale dei testi (stesse righe) ma con più features che vengono chiamate *concetti latenti*.
2. **Matrice 2** → contiene 0 in tutte le celle eccetto la diagonale, che contiene i *singular value*.
3. **Matrice 3** → nuova rappresentazione multidimensionale delle features latenti ma trasposta

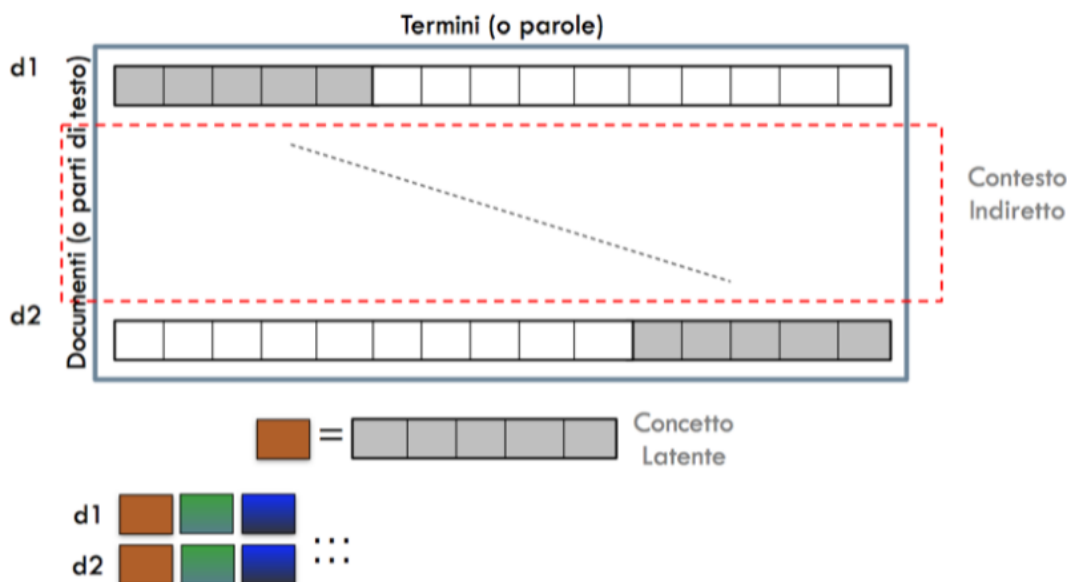
Questo metodo quindi partendo dalle matrici delle frequenze evidenzia automaticamente le ridondanze.

Avendo **matrici Term-Documento** (quindi matrici che sulle righe hanno le parole e sulle colonne i documenti) viene catturata l'informazione delle co-occorrenze di parole X con parole Y, creando così nuove dimensioni/features che accorpano tali occorrenze, dando vita a concetti latenti. **Questi concetti latenti hanno la qualità di essere ordinati, dal concetto latente più importante a quello meno importante.**

L'SVD permette quindi di approssimare la matrice di partenza in altre matrici, molto più piccole, che rappresentano il contenuto numerico espresso nella matrice di partenza con minor uso di dimensioni.

I vantaggi principali di questo metodo sono due:

- Riduce le dimensioni e quindi la scarsità di dati
- I nuovi vettori sono molto potenti per valutare una similarità lessicale indiretta
 - Questi vettori però sono meno interpretabili



trasformando d1 e d2 si potranno cogliere i concetti latenti e quindi se questi sono accomunati o meno.

I problemi di questo metodo però sono molteplici:

- Non generalizziamo i modelli non visti → se ci sono nuovi documenti si deve rifare la trasformazione
- Valori negativi dopo la trasformazione sono difficili da interpretare
 - Sono state sviluppate varianti come la non-negative matrix factorization

Un'evoluzione di questo metodo sono la probabilistic LSA (pLSA) oppure la Latent Dirichlet Allocation (LDA) (che il prof suggerisce nell'esercitazione)

Quindi riassumendo: Un sistema di topic modeling cattura cluster di parole affini che molto probabilmente esprimono un argomento all'interno di una collezione di testi usando metodi statistici che sfruttano le trasformazioni matriciali per trovare le features latenti.

Esiste una variante che è detta Dynamic Topic Modeling in cui i topic vengono proiettati nel tempo per poter catturare la loro evoluzione.

Text Visualization

Spesso nel text mining andiamo a ottenere risultati che sono su n dimensioni, questo non ci permette di mostrare delle rappresentazioni grafiche.

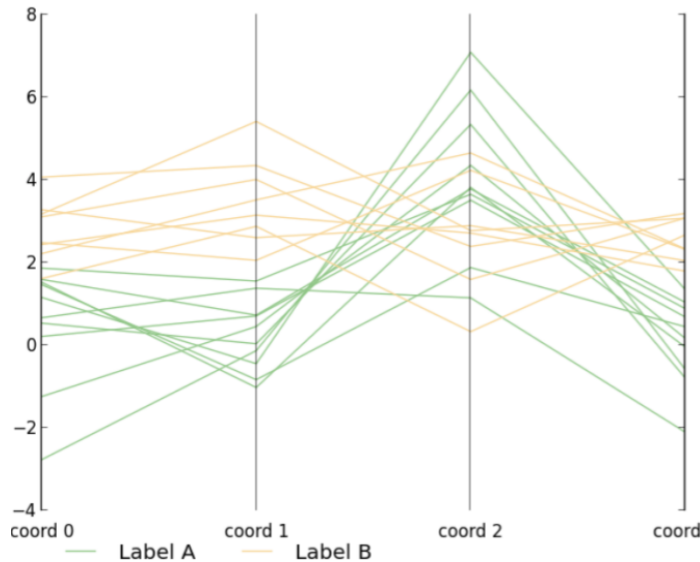
Dobbiamo quindi rappresentare il tutto in modo bi-dimensionale.

IL Text Visualization utilizza diverse strategie di mapping multidimensionale (fattorizzazioni matriciali, Multi Dimensional Scaling (MDS), e approcci grafici come Parallel Coordinates, RadViz, HeatMap, Correlation Circle, ecc.).

Di seguito vediamo alcuni approcci grafici.

Parralel Coordinates

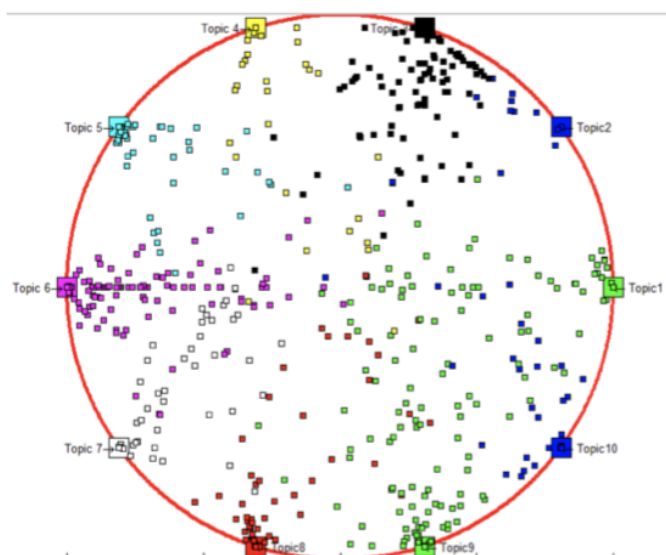
Ogni dimensione viene associata ad una coordinata parallela alle altre. Un punto nello spazio multidimensionale diventa una retta che congiunge i valori per quelle dimensioni. Se si colorano differemente le linee in base alla loro posizione/trend si possono identificare dei cluster semantici che rappresentano gli argomenti di un testo.



RadViz

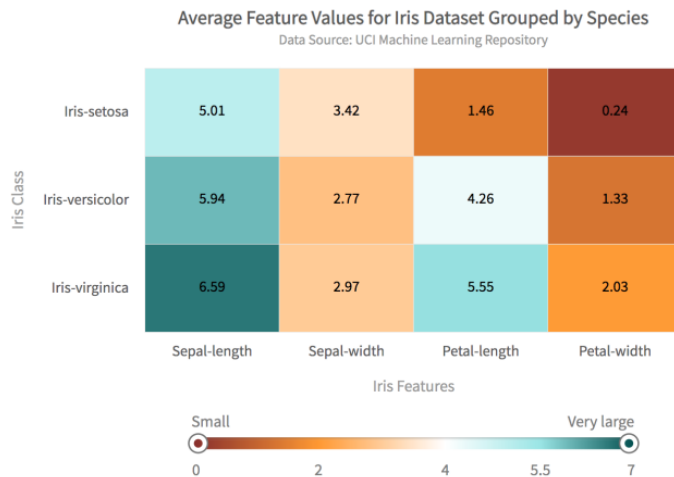
Le dimensioni vengono inserite all'interno di una circonferenza, i punti all'interno sono delle istanze(documenti) e la loro posizione è calcolata in base all'attrazione delle dimensioni/features poste sulla circonferenza.

Quindi di base abbiamo un cerchio in cui nei vari "angoli" ci sono le classi/topic, ogni punto del grafico è un documento che viene "attratto" dal topic/classe .



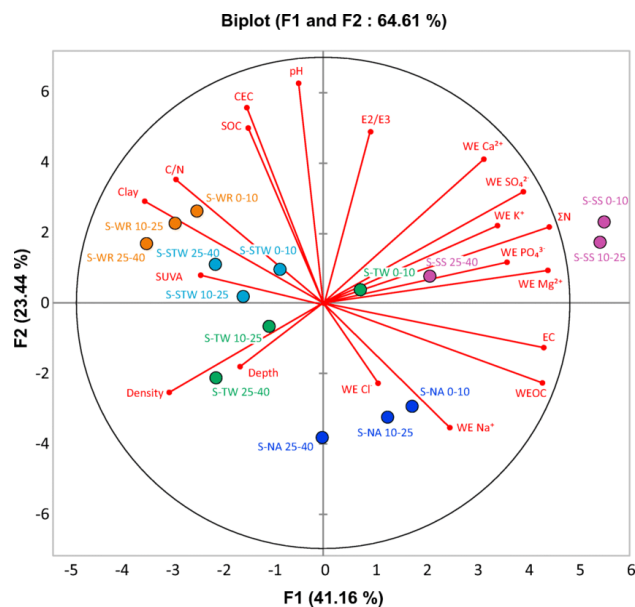
HeatMap

Nelle HeatMap i dati vengono espressi attraverso una matrice e il colore o l'intensità del colore esprime il valore numerico. Viene per esempio usata in machine learning per vedere quanto sono correlate delle variabili.



Correlation Circle

I Correlation Circle sono semplicemente varianti di cerchi in cui sulla circonferenza vengono posti degli elementi (ad esempio termini, o topic) e i collegamenti fra questi rappresentano la loro correlazione. Questo è quindi un modo per visualizzare correlazioni, co-occorrenze, ecc. I termini possono anche essere organizzati gerarchicamente, come nel caso della figura sottostante.



Text2Everything: Non Solo Testo

Un questa parte parleremo dei vari modelli di NLP che vanno oltre il linguaggio, come i modelli che generano immagini da testo, o voci ecc

I modelli possono apprendere in diversi modi:

- **Apprendimento supervisionato** → si danno dati etichettati così da poter correggere i pesi del modello

- **Apprendimento debolmente supervisionato** → vengono date etichette imprecise o con rumore per generalizzare meglio
- **Apprendimento semi-supervisionato** → alcuni dati sono etichettati e altri no, quelli etichettati per apprendere e quelli non etichettati per generalizzare
- **Apprendimento auto-supervisionato** → il modello genera la propria supervisione prendendo parti dall'input. non si usano etichette (BERT usa questo metodo)
- **Apprendimento per rinforzo** → Si impara in base agli errori commessi ricevendo un feedback. Il modello ottimizza le proprie azioni per massimizzare una ricompensa.
- **Fine tuning Pre Training** → verranno approfonditi in seguito

Pre Training e Fine tuning

- Pre Training → i pesi della rete addestrata vengono salvati per poi essere usati come punto di partenza per l'allenamento di una rete diversa su un nuovo compito.
 - **ci si risparmia l'addestramento iniziale della rete che spesso è molto oneroso**
- Fine Tuning → usando come partenza il pre training la rete viene addestrata su compiti più specifici utilizzando un set di dati più piccolo e spesso etichettato.

In altre parole, il pre-training insegna al modello le regole generali dei dati, mentre il fine-tuning lo addestra su un compito specifico, consentendogli di applicare efficacemente queste regole generali in un contesto specifico.

Tecniche di Pre Training

- **Masked Language Modeling (MLM)** → alcune parole vengono cancellate randomicamente dalla sequenza di input, il modello viene così allenato a prevedere le parole cancellate sulla base del contesto fornito e dalle parole rimanenti → **il modello può apprendere la struttura e la sintassi del linguaggio**
- **Denoising Auto Encoder (DAE)** → inserisce rumore nei dati del modello, il modello deve rimuovere il rumore dai dati in input → **questo permette di apprendere le caratteristiche di dati esposti a distorsione**
- **Replaced Token Detection (RTD)** → task discriminativo, il modello deve determinare se il token è stato sostituito da un altro token generato da un altro modello → **aiuta il modello ad apprendere contesto e coerenza del linguaggio**
- **Next Sentence Prediction (NSP)** → vengono aggiunte nuove frasi da documenti diversi e il modello deve determinare se l'ordine è corretto → **aiuta a catturare la rappresentazione a livello di frase e a comprendere le relazioni semantiche tra le frasi.**
- **Sentence Order Prediction (SOP)** → variante di NSP, invece di usare frasi da documenti diversi, si usano campioni dello stesso documento, due frammenti contigui come positivi e due frammenti invertiti come negativi → **aiuta a comprendere la sequenzialità all'interno di un singolo documento**

Task di modelli Pre Trained

Questi modelli possono essere multi modali quindi non considerare solo testo, vediamo quindi quali sono questi task:

- Classificazione di testo

- Generazione di testo
- Riassunto
- Classificazione delle immagini
- Image Segmentation → assegnare etichetta a ogni pixel dell'immagine
- Rilevamento degli oggetti → le caselle delimitatrici e le classi degli oggetti in un'immagine
- Classificazione Audio
- Automatic Speech Recognition → trascrizione del testo
- Risposta a domande Visive
- Document Question Answering
- Didascalia delle immagini → generare didascalia per un immagine

Modelli Open Source (Aperti) e Chiusi

Esistono due tipi di accesso ai modelli:

- **Modelli Aperti** (open source) → si ha accesso al codice e si può modificare e scaricare liberamente
 - **Personalizzazione** → si può personalizzare il modello a piacimento, modificare i metodi di addestramento e integrare il modello in contesti più complessi
 - **Privacy** → non si condivideranno informazioni complesse tramite API, questo permette alle aziende una sicurezza in più
 - **Costi** → avendo un modello aperto si evitano di pagare tariffe che un modello chiuso potrebbe richiedere, abbiamo però costi relativi alla potenza di calcolo e storage
 - **Indipendenza dai fornitori** → con un modello aperto non si hanno dipendenze legate a un fornitore. Se un fornitore di API cambia politiche potremmo essere soggetti a aumenti del prezzo
 - **Ricerca e sviluppo** → I modelli aperti sono fondamentali per la ricerca scientifica
- **Modelli chiusi** → si possono usare solo attraverso delle API

RLHF: Reinforcement Learning from Human Feedback

Il modello viene addestrato usando delle ricompense.

La ricompensa si usa per ottimizzare la politica di un agente attraverso l'apprendimento per rinforzo (RL). Il modello di ricompensa viene addestrato in anticipo rispetto alla politica da ottimizzare per prevedere se un dato output è buono (alta ricompensa) o cattivo (bassa ricompensa). Si può migliorare la robustezza del modello se la funzione è scarsa o rumorosa.

Il feedback umano viene raccolto chiedendo agli umani di classificare gli esempi di comportamento dell'agente. Queste classificazioni possono quindi essere utilizzate per valutare gli output, ad esempio con il sistema di valutazione Elo.

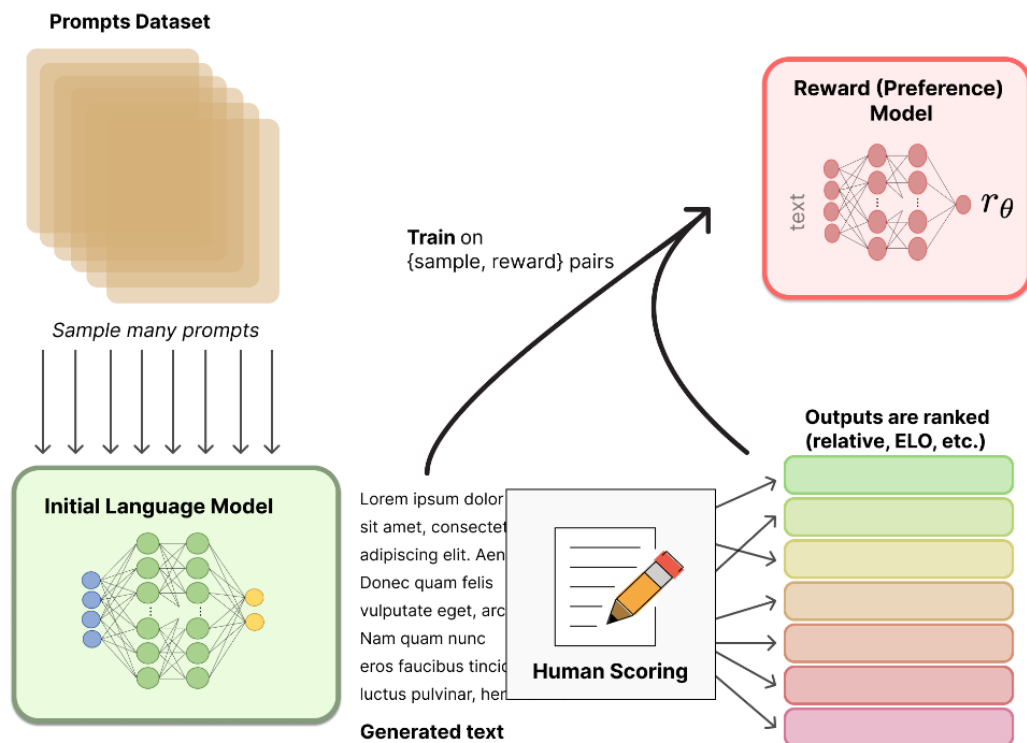
Usare algoritmi di rinforzo in contesti di NLP è difficile perché non è facile definire dove dare le ricompense, soprattutto per compiti che coinvolgono preferenze umane e valori.

Un esempio di modello addestrato con RLHF è ChatGPT.

Il Funzionamento è il seguente:

- **Si parte da un LLM pre-trained**, senza fine-tuning
- **Si genera un secondo modello, detto "reward model"**, che data una sequenza di testo genera uno scalare di ricompensa che rappresenta in qualche modo una preferenza umana. Per far ciò, si utilizzano umani per confrontare differenti output dallo stesso prompt. Questi output vengono poi classificati con un sistema, tipo Elo, e normalizzati in punteggi numerici.
- **Poi, utilizziamo il RL per fare fine-tuning del LLM sul reward model.**

In un'immagine:



Per fare fine-tuning con RL bisogna però impostare il problema come un classico problema di Reinforcement Learning:

- **Policy**: un LM che prende un prompt e ritorna una sequenza di testo
- **Spazio d'azione**: tutti i token del vocabolario del LM
- **Spazio d'osservazione**: la distribuzione delle possibili sequenze in input
- **Funzione di ricompensa**: la combinazione del preference model e un constraint sul cambiamento della policy

Andiamo a preferire la distribuzione dei token che piace di più all'umano rispetto ad altre, calcolando la Kullback-Leibler (KL) divergence tra quello proposto e quello voluto.

RLHF ha diverse problematiche, vediamole:

- RLHF ha bisogno di tanti crowdworkers (gente che valuta la predizione?) per poter essere efficace
- RLHF mette **l'etica del modello in mano a chi lo deve giudicare**: le compagnie addestrano questi crowdworkers sulla base delle risposte volute ma il risultato

dell'operazione è sempre limitato all'abilità del singolo di giudicare eticamente o meno una risposta.

- RLHF **ha un costo operativo** (per pagare gli schiavi che valutano le domande) oltre a quello computazionale

Constitutional AI

Metodo di apprendimento auto supervisionato, i passaggi sono i seguenti

- Si prende un modello già con un fine-tune RLHF
- Il modello risponde a molte domande, alcune delle quali potenzialmente dannose, e genera risposte in bozza
- Il sistema mostra al modello la sua risposta in bozza, insieme ad un prompt che dice qualcosa del tipo "riscrivi questo in modo più etico"
- L'AI lo riscrive per renderlo più etico
- Il sistema ripete questo processo fino a raccogliere un ampio dataset di risposte "in bozza" e risposte rivisitate (più eticamente corrette)
- Il sistema addestra il modello a riscrivere risposte meno simili alle bozze e più simili alle rivisitate

Si possono confrontare l'Elo (sistema di valutazione/classificazione dell'output) dell'efficacia e l'Elo dell'innocuità con l'approccio RLHF standard e quello RL Costituzionale.

La pratica convenzionale suddivide l'IA etica in due categorie: "efficacia" e "innocuità".

- Efficace → Un'IA è definita efficace se risponde alle domande in maniera appropriata
- Innocua → È definita innocua se non compie azioni dannose o offensive

Questi fattori possono però entrare in conflitto, un AI può essere innocua evitando di rispondere a tutte le domande come potrebbe essere estremamente efficiente che però risponde alle domande su come si costruisce una bomba.

Le società che producono questi modelli ambiscono a creare sistemi che bilancino questi due obiettivi, collocandosi lungo una sorta di frontiera di Pareto: non possono incrementare l'efficacia senza compromettere l'innocuità, e viceversa.

Text2Text

Alternativa al Fine-Tuning : Lora

- Molto spesso non è semplice adattare un modello ad un certo downstream task
- Quasi sempre vengono utilizzate tecniche come il fine-tuning, che, però, prevedono il re-training di tutti i parametri del modello
- Poco tempo fa è nata LoRa (Low-Rank Adaptation), una tecnica utilizzata in NLG che migliora nettamente il processo di adattamento ai task
- LoRa prevede il freeze dei pesi del modello e l'iniezione di matrici trainable rank decomposition nel modello

Matrice trainable rank

Una rete neurale contiene molti strati densi che non fanno altro che moltiplicazioni matriciali. Le matrici peso in questi layers sono full-rank (ovvero, colonne e righe sono linearmente indipendenti)

L'idea di LoRa è che se i pesi di modello possono avere una loro dimensione intrinseca, ovvero possiamo comprimere modelli ed ottenere prestazioni simili, allora possiamo comprimere e sostituire matrici full-rank con corrispettivi low-rank

Quantizzazione

La quantization è una tecnica per ridurre il costo computazionale (e di memoria) dell'inferenza → Consiste nel rappresentare i pesi e le attivazioni con datatype a minor precisione, Ad es. da `float32` a `int8` oppure, il più usato, da `float32` a `bfloat16`.

L'idea è banale: ridurre il numero di bit significa, in questo modo il modello utilizzerà meno memoria e consumerà meno energia dato che operazioni come le moltiplicazioni matriciali saranno più veloci, grazie all'aritmetica su interi.

QLoRa, ad es. utilizza una quantization a 4bit (la più piccola mai utilizzata!)

Basicness

Passiamo dal concentrarci sulla grammatica alle parole.

Perché anche le parole possono avere diversi livelli di complessità. A questo si collega il continuo evolversi del vocabolario umano.

Brown studia come ci siano diverse parole che spesso siano comuni e usate dai bambini, semplici da capire, da imparare e soprattutto riferiscono concetti comuni a tutti gli esseri umani → queste sono parole Base. Ma segue subito una domanda, come posso contraddistinguere queste parole ? → risolvere questa domanda aiuterebbe a spiegare e descrivere oggetti, situazioni e concetti.

Questa domanda è particolarmente rilevante per i concetti complessi, per i quali è difficile trovare un termine "definitivo"/facile per spiegarli.

Ogden sostiene che dovrebbe esistere un insieme di parole base che *dovrebbe costituire la base della comunicazione quotidiana, pensato per essere la base lessicale per una comunicazione efficace, chiara e precisa*. **Vogliamo quindi trovare un dizionario di base.** Le caratteristiche che rendono una parola basica devono soddisfare i seguenti vincoli (secondo Brown):

- **Essere brevi**
- **Essere relative a concetti concreti**
- **Facili da pronunciare**
- **Molto utilizzate**

Gerarchia dei termini

È stato poi introdotto l'idea di una gerarchia di termini dai più generici a quelli più specifici.

(Un po' come in wordnet, in cui in cima abbiamo Essere vivente e come foglia abbiamo dalmata).

Il livello **Basico** si colloca più o meno a metà di questa gerarchia e sono quindi detti **Middle Level** → di solito si apprendono prima i concetti come “forchetta” per poi passare “posata” (esempio dal basso verso l’alto). In un caso opposto si apprende prima “cane” e poi “bassotto” (dall’alto in basso)

Viene anche ipotizzato il concetto che queste **livelli basici fungano per la sopravvivenza sociale** ovvero l’acquisizione di questi vocaboli diventa essenziale per imparare una nuova lingua, questo per soddisfare i bisogni primari e la comunicazione quotidiana.

Vediamo però i problemi di questa teoria

- **Legame termine concetto**
 - Non tutti i concetti hanno una parola che li descrive facilmente.
 - Molti termini basic sono usati per concetti complessi (**cane** della pistola)
- **Soggettività** → per ogni lingua i concetti base vengono definiti in modo soggettivo
- **Non esiste una definizione per i termini “advance”** → tutto quello che non è basic è advanced.
- Qual è lo scopo dei concetti base ?
 - Quando si apprende una lingua questi concetti sono fondamentali
 - Il loro sfruttamento facilita l’apprendimento dei concetti complessi e avanzati

Ontology Learning e Open Information Extraction

Lavoreremo sulla Semantica Lessicale ma in modo più matematico → come costruire un’ontologia in modo automatico usando dati non strutturati. Questo procedimento è detto chiamato ontology learning.

Alcune delle operazioni che effettuiamo sono legati al Text Mining o altre cose che abbiamo visto.

Definizione: “**data una conoscenza di un certo dominio, con la sua rappresentazione e la sua codifica si cerca di ritornare alla concettualizzazione di partenza**” → si tratta di reverse engineering.

Abbiamo principalmente due problematiche legate a questo processo:

- la conoscenza del mondo non è uguale per tutti e quindi esistono versioni differenti
- la concettualizzazione lavora principalmente sull’utilizzo dell’oggetto e quindi ignora molte caratteristiche dell’oggetto

L’Ontology learning comprende molte sottosezioni:

- **Ontology Population**: avendo un’ontologia preesistente si vuole estrarre dal testo le informazioni per popolarla (**Popolare Ontologia**)
- **Ontology Annotation**: avendo un’ontologia preesistente l’obiettivo è quello di taggare il testo con delle informazioni concettuali → concettualizzazione come annotazione semantica (**Annotare ontologia**)

- **Ontology Enrichment:** avendo un'ontologia preesistente si vuole andare a trovare nuove istanze ma anche arricchire l'ontologia generando nuovi concetti e relazioni (**Arricchire ontologia**)

Il grado di formalizzazione dei dati si può rappresentare in modo crescente:

1. Testo non strutturato
2. Terminologia con termini del dominio che ci interessa
3. Rappresentazione a glossario che hanno riferimenti alle definizioni
4. Rappresentazione su un thesaurus (wordnet) → che ha relazioni tra parole, sinonimi ecc
5. Tassonomia
6. Ontologia

In genere passando dai metodi meno sofisticati a quelli più avanzati la complessità degli approcci automatici aumenta.

Task

I differenti task sono:

- **Term Extraction** → trovare nomi per concetti e relazioni tra concetti
- **Synonym Extraction** → estrazione di parole che hanno lo stesso significato in determinati contesti
- **Concept Extraction**
 - **Intensionale** → anche detta gloss learning → si cerca di astrarre e di rappresentare attraverso un formato stringato tutto quello che descrive un concetto
 - **Estensionale** → Enumerare gli elementi che descrivono un concetto
- **Concept hierarchies Induction** → si cerca di strutturare concetti già noti attraverso una tassonomia
- **Relation Extraction** → si cerca di strutturare dei concetti già noti attraverso delle relazioni (simile a quello che si fa con le Pair-Pair Matrix)
- **Population** → fatto attraverso
 - Named Entity Recognition (NER) → popolare ontologia attraverso relazioni "instance-of"
 - Information extraction (IE) → più generale rispetto al NER
- **Notazione di Sussunzione:** meccanismi che permettono di costruire automaticamente gerarchie come la FCA

Metodi

Per risolvere i task si hanno tre metodi:

- **Natural Language Processing** come:
 - estrazione di informazione quali Part-of-Speech o named entities
 - pre-processing
 - regole di alberi di parsing
 - informazioni statistiche
 - risorse lessicali

- **Matematica** (Formal Concept Analysis) → approcci matematici formali
 - costruzione automatica di gerarchie
- **Machine Learning**

Formal Concept Analysis (FCA)

Si compone di 3 elementi,

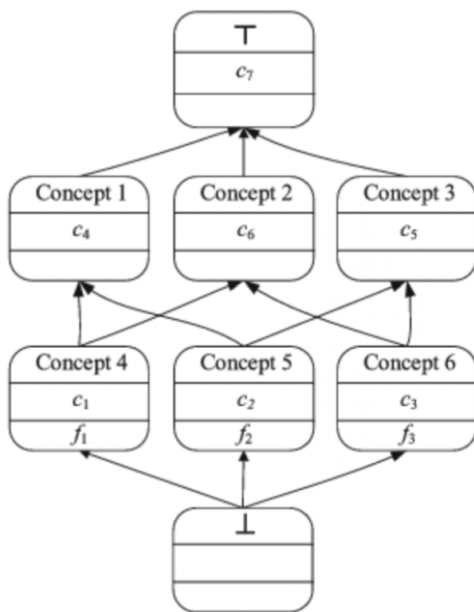
- *oggetti* → equivalenti dei concetti (o istanze) che sono associate a attributi
- *attributi* → caratteristiche degli oggetti
- *incidenza*. → indica se un oggetto possiede o meno un attributo

Queste relazioni vengono unite in una formal context (ovvero delle matrici di adiacenza) → che ha su ogni riga un attributo e su ogni colonna un oggetto.

Sugli elementi di questa matrice si possono applicare degli operatori:

- Up → viene chiamato sulle colonne della matrice (sugli oggetti) e riporta gli attributi degli oggetti
- Down → viene applicato sulle righe e specifica quali oggetti possiedono quel'attributo

Esiste un metodo che data una formal context restituisce un lattice di concetti, il tutto grazie a questi due operatori. **Un lattice è una struttura a grafo** risultante dal formal context, risulta come segue:



Questo algoritmo per creare il lattice tenta sostanzialmente di creare un ordine tra gli oggetti in base alla loro appartenenza a determinati attributi: sfruttando il fatto che esistono oggetti più generali di altri. Viene costruito un lattice (reticolo) in cui **innanzitutto vengono considerati gli insiemi di elementi che non possiedono attributi, poi gli insiemi degli elementi che ne possiedono uno, poi due, e così via**. Attraverso quindi la navigazione di questa struttura, è possibile costruire eventualmente anche una gerarchia di oggetti (e dedurre un'ontologia).

Open information Extraction

Estrazione di informazioni da corpora di grandi dimensioni, si tenta di analizzare il testo sotto tutti i punti di vista (sensi, gloss, sintassi ecc)

Queste operazioni sono però costose da un punto di vista computazionale.

I tipi di estrazioni sono:

- **“Open” (OIE)** → indica l'estrazione allo scopo di costruzione di conoscenza semi strutturata, nella maggior parte dei casi sono triple con informazioni relazionali.
 - Le triple sono composte da: (Argomento,Espressione verbale,Argomento) → ci si può collegare alla teoria di Hanks ?
 - **Rischia però di estrarre molto rumore**
 - si possono inserire vincoli su argomenti e verbal phrase per ridurre il rumore
 - Riduce lo spazio testuale e permette l'esecuzione di query → ritornami le triple che hanno un determinato verbo nel verbal Phrase, oppure un determinata espressione verbale
 - Approccio data oriented che permette estrazione pseudo semantica
 - Usato per question answering (che alla fine è rispondere alle query)
 - **Non esiste un approccio preciso per l'estrazione di triple → triple disallineate nei diversi sistemi**
 - **Non avendo uno standard è difficile valutare questi sistemi**
 - **Non è sempre semplice applicare l'uso di triple estratte in contesti reali**

Large Language Models e Prompting

Introduzione al Prompting

I LLM sono modelli allenati a predire il prossimo token in un testo. Ma il tipo di frase che viene posta a un LLM può cambiare molto il testo generato. Il prompting è lo studio di come formulare questi input/frasi.

Vediamo quali sono le **linee guida del prompting**:

- **Istruzioni chiare e precise** → che dicano esattamente quello che deve fare
- **Uso dei segni e punteggiatura** → per specificare meglio le richieste
- **Richiesta di formato output** → per avere i risultati in uno specifico formato
- **Iterative Prompt Development** → iterativamente si cerca di migliorare il prompt per affinare il risultato

Vediamo anche dei principi generici per ottenere risultati di qualità:

- **Prompt che segue una sequenza di istruzioni**
- **Prompting iterativo** → chiedere al modello una risoluzione in step in modo da capire meglio la risposta (per problemi matematici per esempio)
- **Prompting basato su dialoghi** → inserire un dialogo e chiedere al modello di continuare
- **Forzare il modello a ragionare piano piano** → chiedere al modello una lista di semplici step di piccoli step di ragionamento
- **Richiesta di feedback** → chiedere al modello feedback sul lavoro

Summarization con LLM

Usare LLM per sintetizzare testi. Si inserisce un testo e viene chiesto di generare un testo che contenga tutte le informazioni rilevanti.

Le caratteristiche sono:

- Si **condensano le informazioni più rilevanti** fornendo un output di qualità
- Si può chiedere che la **risposta abbia un numero limitato di token(parole)**
- Su può chiedere che l'**output si concentri su un aspetto particolare del testo**
- Risparmio di tempo

Inferenza di LLM

Gli LLM possono essere usati per fare inferenza su testi.

Esempi di inferenza sono:

- sentiment analysis
 - In modo binario → questo documento ha un sentimento positivo o negativo ?
 - Oppure chiedere una lista di emozioni
- Information extraction dai testi
 - estrarre informazioni specifiche come il nome di un'azienda da un testo.
 - Chiedere a quali topic fa riferimento il testo
- Generazione di dati sintetici

Trasformazione con LLM

Trasformare i testi in altri formati, lingue o stili di scrittura.

I task possono essere:

- Trovare gli errori in un testo
- Traduzione e rilevamento della lingua di un testo
- Cambiamento del tipo di scrittura
- Comunicazione interna di un sistema in più lingue
- Trasformazione di formati → da JSON e XML

Espansione con LLM

Task di Espansione di un testo → ovvero partendo da un materiale di partenza (relativamente piccolo) si va ad aumentare il contenuto generandone di nuovo.

Concetto di **Temperatura** → La temperatura è un parametro che può essere regolato per controllare la variabilità dell'output del modello.

Temperatura bassa = generazioni più prevedibili

Temperatura alta = generazione imprevedibile

Un esempio di temperatura è il valore Chaos che abbiamo su midjourney, per esempio questo è il prompt "una mela rossa".



Search/IR con LLM

Usare gli LLM come sistemi di ricerca dati al pari dei sistemi di ricerca come Google.

Esistono diverse tecniche:

- **Formulare i prompt come una domanda normale/generica** → “Come posso riparare un rubinetto che perde?”
- **Formulare i prompt come domande più specifiche e dettagliate che forniscono più contesto e informazioni per la ricerca** → Usare filtri temporali o per rilevanza
- **Extra:** Esistono plugin che permette a ChatGpt di cercare in rete oppure Bing e Bard che hanno accesso a internet e sono consultabili con chat testuali

Bisogna in ogni caso ricordarsi che possono dire cose sbagliate

Aspetti pratici

Il prompting riguarda diversi fattori:

- **Istruzioni** → si riferisce ad una specifica attività o direzione che il modello deve eseguire.
- **Contesto** → può includere informazioni esterne o contesti aggiuntivi che possono guidare il modello verso risposte migliori
- **Input dati** - si riferisce all'input o alla domanda per la quale vogliamo trovare una risposta.
- **Indicatore di Output** - indica il tipo o il formato dell'output.

Istruzioni e consigli

Si possono usare comandi come:

- Scrivi
- Traduci
- Ordina

- Riassumi
- Classifica

Si consiglia inoltre di:

- Evitare la negazione → evitare di dire “non voglio questi valori” ma concentrarsi sul “voglio i valori che soddisfano questi valori”
- Se si pone una richiesta al modello siamo nel caso (**zero shot**)
- Si possono usare degli esempi per dire al modello cosa si vuole (**caso few-shot**)
- Si può aggiungere al prompt una richiesta esplicita in più passaggi (caso **chain-of-thought**)