
ANNO ACCADEMICO 2024/2025

Tecnologie del Linguaggio Naturale

Teoria - Knowledge Base

Altair's Notes



**UNIVERSITÀ
DI TORINO**

DIPARTIMENTO DI INFORMATICA

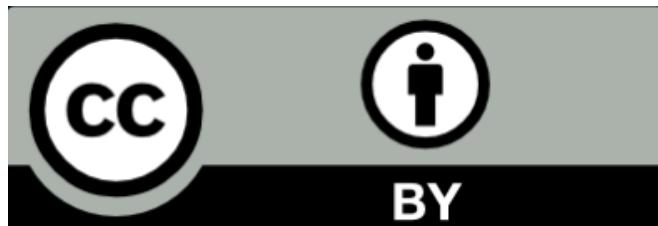
CAPITOLO 1	SEMANTICA LESSICALE	PAGINA 5
1.1	Introduzione Programma della Seconda Parte del Corso — 5 • Che cos'è la semantica lessicale? — 5 • Alcune Definizioni di Ontologia — 6	5
1.2	Design di Ontologie DOLCE — 8	8
CAPITOLO 2	KNOWLEDGE REPRESENTATION	PAGINA 11
2.1	Knowledge Bases Rappresentazioni Strutturate — 11	11
2.2	Rappresentazioni Gerarchiche Eredità delle Proprietà — 14	13
2.3	Frame I Frame per Rappresentare la Conoscenza — 15 • Strutture dei Frame — 16	15
2.4	Teorie del Significato Il Significato delle Parole — 17 • Calcolo Sintagmatico del Significato — 19	16
CAPITOLO 3	WORDNET E FRAMENET	PAGINA 21
3.1	WordNet La Matrice Lessicale — 22 • Relazioni Semantiche — 22 • Sostantivi — 23 • Verbi — 24 • Conceptual Similarity e Word Sense Disambiguation — 25	21
3.2	FrameNet La Teoria dei Frame — 26 • Il Frame "Revenge" — 27 • Frame Semantics e Text Understanding — 29	26
CAPITOLO 4	NATURAL LANGUAGE TOOLKIT (NLTK)	PAGINA 31
4.1	Introduzione ai Tasks di NLP Tokenizzazione — 31 • Stop-Words Filtering — 31 • Stemming — 32 • PoS Tagging — 33 • Lemmatizzazione — 34 • NER — 34	31
4.2	SpaCy La Pipeline — 35	34
CAPITOLO 5	RAPPRESENTAZIONI DI SIGNIFICATO	PAGINA 38
5.1	Modelli Probabilistici Ngrams — 38 • Valutare un LM — 40 • Smoothing — 40	38
5.2	Rappresentazioni Vettoriali Modello di Spazio Vettoriale — 41 • Confrontare Vettori — 42 • Il Metodo di Rocchio — 43 • WordToVec — 44 • Altri Embeddings — 44	41

CAPITOLO 6	UN'INTRODUZIONE AI TRANSFORMERS	PAGINA 46
6.1	Contextualized Embeddings Architettura — 46 • Strati — 48	46
6.2	Training Positional Embeddings — 50 • Sub-word Tokenization — 51 • Attention — 52 • Training — 53	49

Premessa

Licenza

Questi appunti sono rilasciati sotto licenza Creative Commons Attribuzione 4.0 Internazionale (per maggiori informazioni consultare il link: <https://creativecommons.org/licenses/by/4.0/>).



Formato utilizzato

Box di "Concetto sbagliato":

Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contenente il concetto giusto.

Box di "Corollario":

Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 0.0.1: Nome delle definizioni

Testo della definizione.

Box di "Domanda":

Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Osservazioni":

Osservazioni 0.0.1

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.

1

Semantica Lessicale

1.1 Introduzione

1.1.1 Programma della Seconda Parte del Corso

Per quanto riguarda questa parte del corso ci si concentrerà sulla semantica lessicale, in particolare sui seguenti punti:

- Problema della Knowledge Representation.
- WordNet e BabelNet.
- FrameNet.
- N-grams.
- Word/Sense embeddings.
- Encoder/Decoder.

1.1.2 Che cos'è la semantica lessicale?

Definizione 1.1.1: Semantica Lessicale

La semantica lessicale è lo studio del significato delle parole e delle loro relazioni. Lo studio di cosa i singoli oggetti lessicali significano, cosa fanno, come possono essere rappresentati e combinati.

Sono scelte le ontologie per:

- *Parte metodologica*: adottare un piano molto interdisciplinare a un alto livello di generalità.
- *Parte architetturale*: centralità del ruolo che un'ontologia può giocare in un sistema informativo.

Le parole sono flessibili:

- Capacità di ripetere le parole nei *sensi* già noti, combinarle in enunciati anche nuovi, capacità di estendere una parola o una frase per esprimere nuovi sensi:
 - Queste capacità sono ereditate dagli esseri umani.
 - Non si sviluppano in ogni circostanza.
- La cooperazione tra le capacità del punto precedente è importante per sviluppare la *metalinguistica*:
 - Che cosa significa questa parola? Che cosa vuol dire? Si può dire così? Si scrive goccie o gocce?
 - Le parole hanno funzione riflessiva: si usa la *lingua per riflettere sulla lingua*.

Rigidità vs. Deformabilità:

- La chimica e la matematica parlano di determinati aspetti dell'esperienza (sono *rigide*).
- Le lingue invece sono *deformabili*: se ne può alterare o dilatare il significato.

1.1.3 Alcune Definizioni di Ontologia

Domanda 1.1

Cosa significa il termine ontologia?

Definizione 1.1.2: Ontologia Filosofica

In filosofia il termine ontologia indica lo studio dell'essere e delle sue categorie fondamentali. Un'ontologia definisce un insieme di primitive rappresentazionali con il quale modellare un dominio di conoscenza o di discorso. È un sistema organizzato in categorie e relazioni. Le loro definizioni includono informazioni sul loro significato e vincoli su come applicarle in maniera consistente.

Note:-

Per esempio una categoria rappresenta tutti i tipi di entità che possono fungere da soggetto in un predicato.

Problema: questa definizione ammette che quasi tutto può essere considerato un'ontologia.

Definizione 1.1.3: Ontologia come Concettualizzazione

Un'ontologia è una specifica esplicità di una concettualizzazione. Una struttura formale di un pezzo della realtà come percepito e organizzato da un agente indipendentemente da:

- Vocabolario utilizzato.
- L'occorrenza in una situazione specifica.

Note:-

Situazioni diverse coinvolgenti lo stesso oggetto descritte da un vocabolario diverso possono avere la stessa concettualizzazione.

Ontologia e semantica:

- Un'ontologia riguarda ciò che c'è, la semantica si riferisce a ciò che c'è.
- Differenti aspetti del linguaggio hanno differenti ruoli nell'ontologia.

Ontologie e Knowledge Bases:

- Componente *terminologica* (ontologia):
 - Indipendente da un particolare stato.
 - Pensata per supportare servizi terminologici.
- Componente *asserzionale*:
 - Riflette specifici stati.
 - Pensata per problem solving.

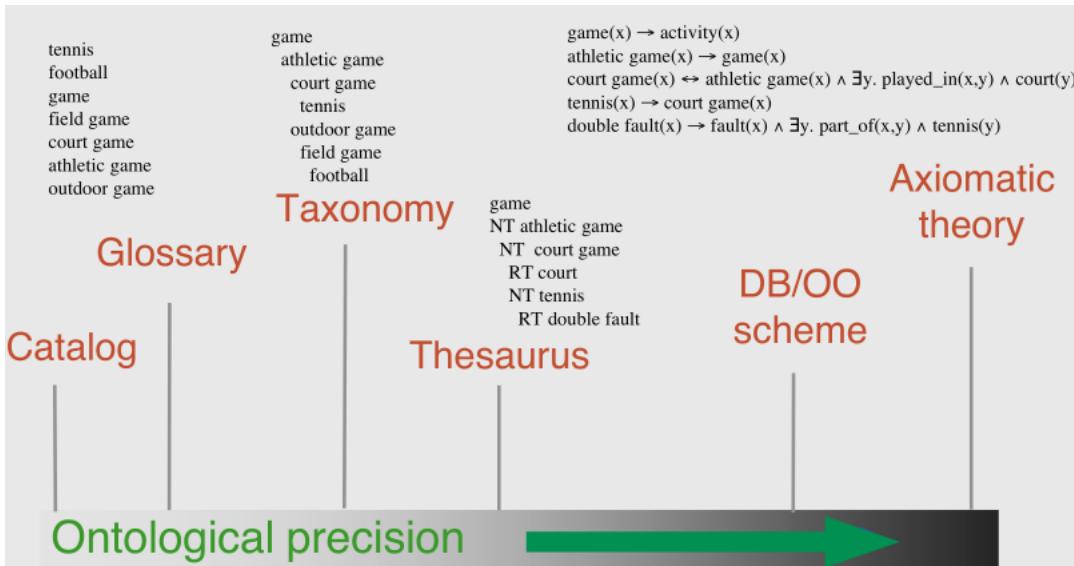


Figure 1.1: Livelli di precisione delle ontologie.

Ontologia e lessico:

- Livello lessicale: il *lessico* è un elenco di parole in una lingua (vocabolario + conoscenza sull'utilizzo delle parole).
- Relazioni lessicali: sinonimia (termini con lo stesso significato), iponimia (relazione di sottoclasse), iperonimia (relazione di superclasse), meronimia (costituente di qualcosa), olonimia (intero), antonimia (termini di significato opposto).

Caratteristiche del lessico:

- *Overlapping Word Senses:*
 - Nelle ontologie le sottocategorie di una categoria generale sono *mutualmente esclusive*.
 - Nel lessico esistono *sovraposizioni di significato* (e.g. quasi sinonimi).
- *Buchi nel lessico:*
 - Il lessico di un linguaggio omette riferimenti a *categorie ontologiche* che non sono lessicalizzate in quel linguaggio.
 - Sono categorie che richiedono perifrasi (giri di parole) per essere definite.
- *Lessici tecnici:* nei contesti tecnici il linguaggio è molto vicino all'ontologia del dominio.

Domanda 1.2

Perché costruire ontologie?

- *Condivisione* della comprensione delle entità di un certo dominio.
- *Riutilizzo* dei dati e dell'informazione.
- Creare comunità di ricercatori.

1.2 Design di Ontologie

Entità ed Eventi:

- *Entità*: oggetti che continuano per un periodo di tempo mantenendo la propria identità.
- *Eventi*: oggetti che accadono, si svolgono o si sviluppano nel tempo.

Definizione 1.2.1: Ontologie Fondazionali

Un'ontologia fondazionale cattura un insieme di distinzioni base valide in vari domini.

Note:-

Alcune celebri sono DOLCE, SUMO e CYC.

1.2.1 DOLCE

Definizione 1.2.2: DOLCE

DOLCE è un'ontologia fondazionale con lo scopo di rappresentare le strutture concettuali di base che emergono dal linguaggio naturale e dalla cognizione umana.

Scelte alla base di DOLCE:

- *Endurant*: entità che sono completamente presenti in ogni momento della loro esistenza.
- *Perdurant*: eventi che si estendono nel tempo e sono parzialmente presenti in ogni istante.
- *Quality*: proprietà specifiche di un'entità, dipendenti da essa.
- *Approccio moltiplicativo*: differenti oggetti ed eventi possono essere co-localizzati nello spazio-tempo.

Definizione 1.2.3: Criteri di Identità

I criteri di identità sono utilizzati per valutare se due entità sono in relazione di sottoclasse e sono proprietà necessarie delle entità confrontate.

Note:-

Per esempio "intervallo di tempo" non può essere sottoclasse di "durata di tempo" perché utilizzano criteri diversi.

Endurants vs. Perdurants in DOLCE:

- Gli endurants possono cambiare nel tempo: lo stesso endurant può avere proprietà incompatibili in tempi diversi.
- I perdurants non cambiano, hanno una locazione spaziale ben definita dagli endurants che vi partecipano.
- La relazione tra endurants e perdurants è la *partecipation*: un endurant vive partecipando in qualche perdurant.

Qualities e Quality regions in DOLCE:

- Le qualities possono essere viste come entità base che possono essere percepite e misurate (e.g. forma, colore, suono, etc.).
- Le qualities sono caratteristiche per specifici individui.
- Si distingue tra una qualità e il suo valore (e.g. il colore di una specifica rosa e quale sia la particolare sfumatura di rosso).

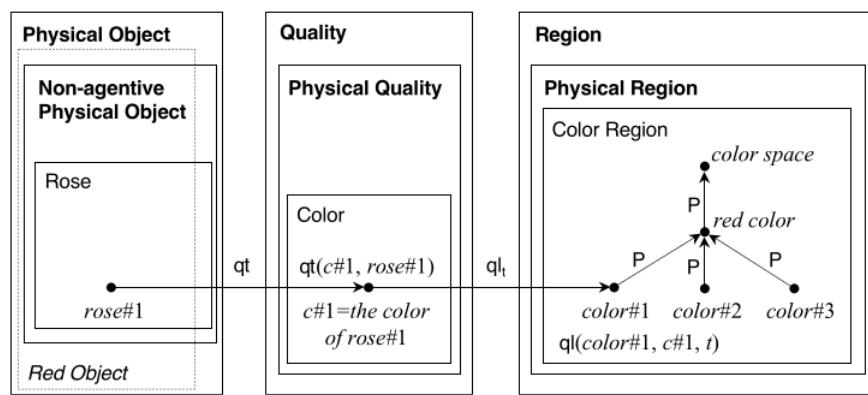


Figure 1.2: Rappresentazione del colore di una rosa.

2

Knowledge Representation

2.1 Knowledge Bases

2.1.1 Rappresentazioni Strutturate

Durante la prima fase iniziale dell'IA ci si è concentrati sullo sviluppo di *metodi generali* per la risoluzione di problemi¹. Successivamente, dalla seconda metà degli anni '60, si vollero abbandonare i domini astratti e semplificati per passare a problemi reali in cui era richiesta una *conoscenza sul mondo in cui il sistema opera*.

Linguaggio e Operazioni:

- Vengono studiati i formalismi adatti a rappresentare le conoscenze necessarie.
- Un sistema di conoscenza deve consistere di:
 - Un *linguaggio di rappresentazione*: un insieme di strutture sintattiche adatte a codificare le informazioni da rappresentare.
 - Un *insieme di regole o di operazioni*: per manipolare il linguaggio.
- L'implementazione delle regole deve portare a *inferenze desiderate* e le regole devono poter essere espresse come *procedure*.

Strutturazione dell'informazione:

- Per rappresentare conoscenze è possibile utilizzare *formule logiche* rappresentanti proposizioni indipendenti.
 - Con le formule logiche non è possibile collegare le varie formule, organizzandole in blocchi omogenei.
 - La logica necessita di ulteriori meccanismi computazionali.
- Esistono *formalismi* per *aggregare conoscenze elementari* in strutture più complesse.
- Si deve poter accedere alla struttura in cui le conoscenze relative all'oggetto in questione sono direttamente disponibili.

Definizione 2.1.1: Reti Semantiche

Le reti semantiche sono un formalismo nato dai primi progetti di traduzione automatica. Hanno la caratteristica comune di utilizzare una struttura a grafo (rete) in cui i nodi rappresentano dei concetti e gli archi rappresentano relazioni tra concetti o proprietà dei concetti.

¹Tali metodi erano indipendenti da specifici domini.

Corollario 2.1.1 Grafi Relazionali

I grafi relazionali sono le reti semantiche più semplici. Sono costituiti da grafi relazionali che permettono di descrivere le relazioni tra le diverse entità del grafo stesso.

Note:-

Per esempio lo si può usare per descrivere uno stato nel mondo dei blocchi.

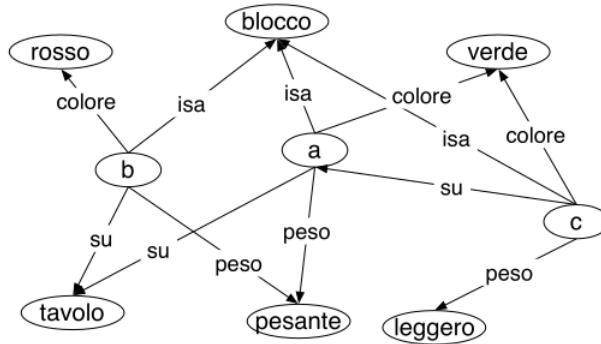


Figure 2.1: Rappresentazione del mondo dei blocchi mediante grafo relazionale.

Osservazioni 2.1.1

- Il grafo è composto da diversi *nodi* ognuno rappresentante un'entità.
- Da ciascun nodo si dipartono *archi* che lo collegano ad altri, sono *etichettati* in modo da esplicitare la relazione che intercorre tra i nodi collegati.
- Una relazione importante è *isA*: il tipo di concetto che un nodo rappresenta.

Espressività di un grafo relazionale:

- È implementato un *sottoinsieme del calcolo dei predicati del primordine*: gli archi sono i predicati e i nodi sono i termini.
- *Limitazioni di efficacia espressiva*: i grafi rappresentano la congiunzione in maniera implicita.
- È difficile rappresentare disgiunzione o implicazione.
- Inoltre è complicato esprimere la quantificazione universale.
- Le relazioni espresse dagli archi sono per loro natura binarie ma i predicati logici possono avere qualsiasi arietà.
- Una possibile soluzione è quella di tradurre tutte le relazioni in relazioni binarie:
 - Accresce la granularità (e quindi l'espressività) e richiede l'introduzione di nodi per rappresentare oggetti e insiemi di oggetti e situazioni e azioni.
 - I predicati con arietà superiore a 2 esplodono in una serie di relazioni binarie: una che chiarisce il tipo di predicato, altre che esplicitano ruolo e funzione degli argomenti.

Note:-

Anche se una relazione ad alta granularità e una come formula del primordine formano un isomorfismo mettono in luce aspetti differenti.

Corollario 2.1.2 Reti Proposizionali

Le reti proposizionali sono reti semantiche i cui nodi possono rappresentare non solo entità semplici, ma intere proposizioni.

Note:-

Ammettendo la possibilità di avere nodi proposizionali si accresce l'espressività del linguaggio. Sono state proposte reti fortemente legate alla logica del primordine.

La negazione:

- Può essere rappresentata mediante un arco che collega il risultato della negazione con la proposizione che viene negata.
- Si possono rappresentare idee piuttosto articolate e distinguere tra:
 - *Negazione di un'intera proposizione*.
 - *Negazione di una proposizione incassata* all'interno di un'altra proposizione.

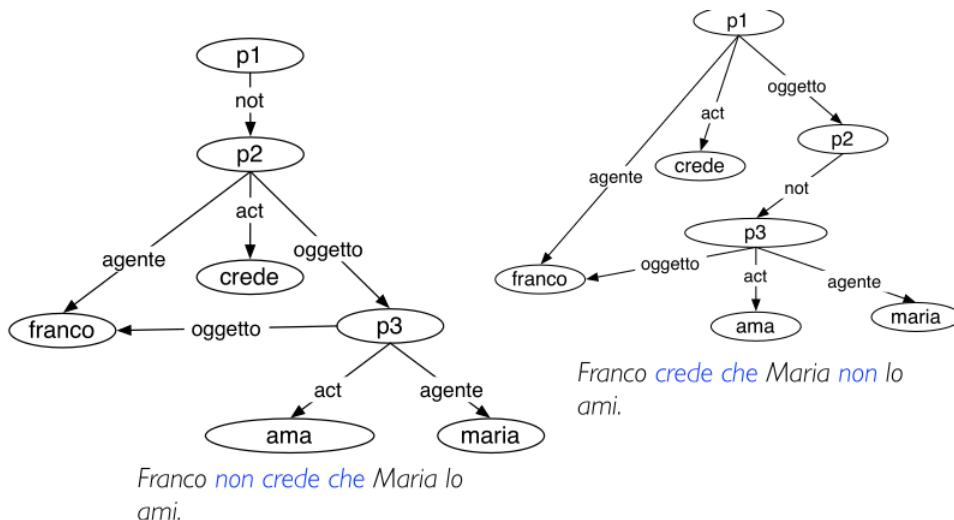


Figure 2.2: Introduzione della negazione.

Disgiunzione:

- Viene introdotta dopo la negazione.
- Questo perché la congiunzione è implicita e si può rappresentare la disgiunzione in sua funzione mediante le *leggi di De Morgan*.

Note:-

La scelta di quale rete utilizzare va effettuata in base a leggibilità, flessibilità, efficienza, facilità di espressione, etc.

2.2 Rappresentazioni Gerarchiche

Molte delle conoscenze sono organizzate in *gerarchie*. Varie entità sono raggruppate in *classi* che a loro volta sono raggruppate in *sottoclassi*. Queste gerarchie non sono limitate a oggetti, ma si possono estendere ad azioni (e.g. camminare, marciare, etc.), eventi, stati, proprietà, etc. Per esempio, nelle reti semantiche se si vuole esprimere l'idea "gli elefanti sono mammiferi" è sufficiente un nodo per gli elefanti e uno per i mammiferi connessi da un arco etichettato *isA*. Così facendo si può fare inferenza, sfruttando proprietà delle relazioni (la relazione *isA* è transitiva).

2.2.1 Eredità delle Proprietà

Definizione 2.2.1: Eredità delle Proprietà

Il meccanismo di eredità delle proprietà afferma che le proprietà asserite per un nodo valgono anche per i nodi che si trovano al livello inferiore della gerarchia.

Se la rete che rappresenta le diverse proprietà *isA* è un *albero* è facile stabilire se un concetto X gode della proprietà *p*:

- Per capire se *p(X)* è vero basta considerare gli antenati di *X* e vedere se qualcuno gode di *p*.
- Questa ricerca è più efficiente dei processi di inferenza.

Vantaggi:

- *Economia di rappresentazione*: invece di replicare una proprietà per tutti i nodi essa viene asserita solo al livello più alto in cui si applica.
- *Semplifica la manutenzione*: una modifica a una proprietà richiede una sola operazione.

Trattamento delle eccezioni:

- Si può risolvere con un approccio procedurale.
- Esempi: gli uccelli solitamente volano, ma alcuni no. I mammiferi partoriscono i figli, ma l'ornitorinco no.

Definizione 2.2.2: Validità per Default

Vengono rappresentate conoscenze che valgono fino a prova contraria. Le eccezioni vengono memorizzate in corrispondenza dei nodi a cui si riferiscono.

Note:-

In questo caso si ottiene l'eccezione prima di raggiungere il nodo di Default.

Definizione 2.2.3: Ereditarietà Multipla

Nel caso una classe abbia più di una classe superordinata (*X isA Y*, *X isA Z* con *Y ⊑ = Z*) la rete semantica passa da albero a grafo.

Note:-

Reti Semantiche 1, Java 0. *If you know, you know.*

Ammettendo l'ereditarietà multipla:

- Il tempo di ricerca passa da lineare a esponenziale, non esistono euristiche.
- Nel caso di un *confitto di valori* non è possibile stabilire un criterio risolutivo che abbia carattere generale.
- Diverse strategie:
 - Ricerca in profondità.
 - Ricerca in ampiezza.

Problemi:

- ✗ Le tecniche di ricerca non permettono di giungere a risultati intuitivamente corretti.
- ✗ I risultati sono inconsistenti. Esempio con Nixon Diamond: Nixon è sia pacifista che guerrafondaio.

Definizione 2.2.4: Dissonanza Cognitiva

Si può considerare la rete stessa come ambigua e in grado di esprimere più interpretazioni: ogni interpretazione è consistente con sé stessa, ma diventa inconsistente quando viene considerata con le altre.

Appartenenza e Inclusione:

- Non esiste una distinzione tra nodi che rappresentano individui e nodi che rappresentano classi o insiemi di individui.
- Il legame isA viene usato per denotare sia l'*appartenenza* (elemento in insieme) che l'*inclusione* (insieme in insieme).
- Non si può distinguere tra proprietà vere per tutti gli individui di una classe e proprietà vere della classe in quanto tale.
- È impossibile separare la semantica di una rete dal suo utilizzo.

2.3 Frame

Le persone utilizzano un insieme strutturato di conoscenze per interpretare le diverse situazioni che si trovano a dover affrontare. Quando ci si imbatte in una situazione si recupera una *rappresentazione a carattere generale* che viene poi raffinata e modificata.

Definizione 2.3.1: Frame

Un frame è una struttura che rappresenta le conoscenze di carattere generale che un individuo ha riguardo situazioni, luoghi, oggetti, etc. Fornisce una cornice concettuale all'interno della quale vengono interpretati nuovi dati alla luce delle conoscenze derivate dall'esperienza precedente.

Caratteristiche:

- Un sistema che usa i frame può *formulare previsioni* e *avere aspettative*.
- Aiuta il processo di interpretazione di situazioni ambigue.
- Vengono facilitati il recupero di informazioni e i processi inferenziali.

Corollario 2.3.1 Script

Lo script è una struttura di conoscenza di alto livello che integra e organizza le frasi.

2.3.1 I Frame per Rappresentare la Conoscenza

Mancanza di una semantica formale:

- I frame rappresentano la conoscenza in maniera dichiarativa, ma senza una semantica formale.
- Si presuppone l'esistenza di procedure in grado di utilizzare le informazioni contenute nei frame.

Concetti nei frame:

- Livello di base: costituisce il modo naturale di categorizzare gli oggetti e le entità.
- Livello superordinato/subordinato: rispettivamente generalizzazioni e specializzazioni di concetti di base.
- *Prototipi*: membri tipici di una categoria. L'appartenenza viene categorizzata in maniera di maggiore o minore somiglianza rispetto ai prototipi (e.g. passero è migliore rappresentante di uccello rispetto ad airone che è migliore rispetto a struzzo).

2.3.2 Strutture dei Frame

Strutture gerarchiche:

- Gli elementi di un frame sono collegati da relazioni isA o *ako* (a kind of) che consentono l'ereditarietà.
- Le proprietà ad alto livello sono fisse, mentre ai livelli più bassi (sottoclassi o istanze individuali) possono essere specializzate.

Domanda 2.1

Com'è fatto un frame?

- Ha un nome univoco.
- Le caratteristiche sono rappresentate da un insieme di slot (caselle in cui viene inserito un determinato tipo di informazione).
- Valori di default per gli slot.
- Conflitti tra valori ereditati in caso di ereditarietà multipla.
- Uno slot può essere una struttura complessa.
- Procedure per rendere la computazione efficiente:
 - If needed: calcolo del valore di uno slot.
 - If added: solo quando si tenta di riempire uno slot.

2.4 Teorie del Significato

Alcune teorie tipiche:

- Prototipi: come accennato in precedenza, si effettuano approssimazioni per rappresentare una categoria.

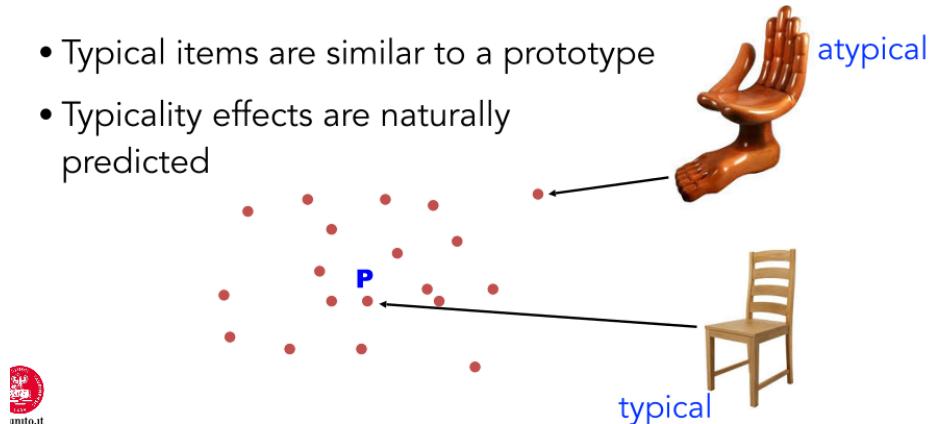


Figure 2.3: Teoria dei prototipi.

- Esempi: la rappresentazione mentale di un concetto è l'insieme di alcuni esempi di quella categoria.
- Teoria: i concetti sono parte della comprensione del significato, collegati ad altri concetti.

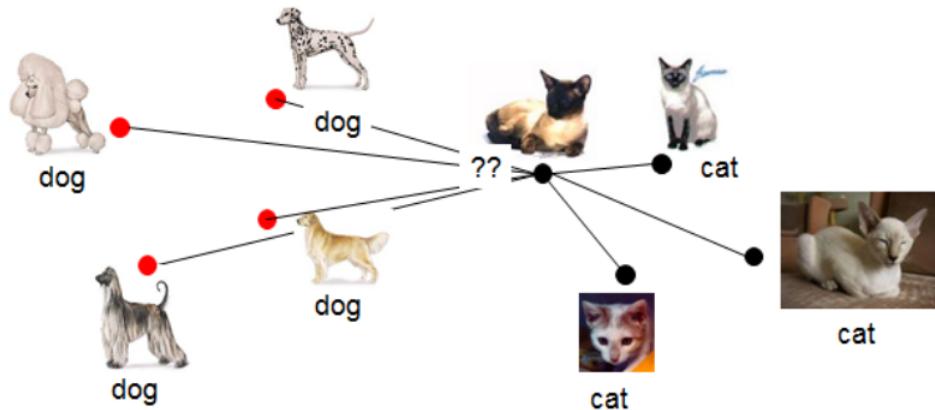


Figure 2.4: Teoria degli esempi.

Note:-

Quelle elencate qui sopra sono proprio tre teorie (probabilmente non avevano voglia di trovare nomi migliori).

Approccio duale:

- *Sistema 1* (implicito): categorizzazione non monotona, rappresentazione continua.
- *Sistema 2* (esplicito): categorizzazione monotona, rappresentazione dei dati proposizionale.

2.4.1 Il Significato delle Parole**Due problemi:**

- Polisemia.
- Semantica frasale.

Definizione 2.4.1: Contesto

Il contesto è l'insieme degli elementi adiacenti a una parola. Può essere:

- Sintattico: elementi adiacenti a una parola dal punto di vista delle loro proprietà sintattiche.
- Semantico: elementi adiacenti a una parola dal punto di vista delle loro proprietà semantiche.
- Linguistico.
- Situazionale (pragmatico): richiede una conoscenza del mondo esterna alla lingua.

Ambiguità e Polisemia:

- L'*ambiguità* è la proprietà di una forma lessicale di avere più di un significato:
 - *Contrastativa* (o omonimia): due significati contraddittori.
 - *Complementare* (o polisemia): stessi significati in contesti diversi.
- *Principio di economicità linguistica*: si usano le stesse parole per esprimere più di un significato, per contenere le dimensioni del lessico, etc.

Note:-

Soltanamente i verbi tendono a essere più polisemici dei sostantivi.

Definizione 2.4.2: Teoria Referenziale del Significato

Le parole sono uno strumento attraverso il quale facciamo riferimento a ciò che esiste. Il significato delle parole consiste nella loro capacità di stabilire una relazione con elementi della realtà al di fuori della lingua.

Il riferimento può realizzarsi attraverso due procedimenti:

- *Denotazione*: si indica la classe di elementi.
- *Designazione*: si indica un particolare elemento della classe.

Interpretazioni:

- Ampia: ciascun elemento della lingua istituisce un riferimento con la realtà extralinguistica.
- Restrittiva: distingue tra atto di riferimento e atto di predicazione.

Definizione 2.4.3: Teoria Mentalista

La teoria mentalista (o concettuale) arricchisce la teoria referenziale con l'ipotesi che il riferimento tra parole e realtà non sia diretto, ma mediato dall'immagine mentale dei concetti. Le parole non fanno riferimento diretto alla realtà extralinguistica, ma soltanto al modo in cui tale realtà è concettualizzata e categorizzata nella mente del parlante.

Mediazione concettuale:

- Si può parlare non soltanto di entità esistenti o di eventi che accadono, ma anche di cose astratte, immaginarie o ipotetiche.
- L'accento è sugli aspetti psicologici e sul legame tra concettualizzazione ed esperienza psico-percettiva.

Concetti e lessicalizzazione:

- I concetti cognitivi sono entità instabili: possono differire individualmente e culturalmente:
 - Appartengono alla struttura mentale.
 - Possono essere considerati come degli universali.
- I concetti lessicalizzati sono più stabili: individualmente e socialmente condivisi:
 - Appartengono alla struttura linguistica.
 - Variano da lingua a lingua.

Definizione 2.4.4: Teoria Strutturele

Il significato dei termini non è il suo riferirsi a un oggetto, ma nel valore che la parola assume in relazione alle altre parole presenti nella lingua che fanno parte dello stesso campo semantico.

Note:-

Il valore semantico di un termine è il suo contenuto informativo.

Definizione 2.4.5: Teoria Distribuzionale

Il significato delle parole è determinato in larga misura dall'insieme di altre parole con cui queste co-occorrono. Questa teoria è ritornata di moda di recente grazie all'enorme disponibilità di corpora.

Corollario 2.4.1 Metafora Geometrica del Significato

I significati delle parole corrispondono a punti in uno spazio multidimensionale, in maniera tale che a punti vicini corrispondono parole con significato prossimo.

Note:-

Questo è alla base delle rappresentazioni vettoriali e della cosine similarity.

2.4.2 Calcolo Sintagmatico del Significato

Problemi fondamentali delle teorie del significato:

- Contestualità del significato.
- Polisemia.

Definizione 2.4.6: Principio di Composizionalità del Significato

Spiega come il significato degli enunciati si formi a partire dal significato degli elementi lessicali che li compongono.

Osservazioni 2.4.1

In alcuni casi questo principio viene meno:

- Espressioni idiomatiche.
- Usi metaforici.
- Polisemia.

Possibili approcci ai problemi:

- *Enumerazione dei sensi*: i diversi sensi sono elencati nella parola, specificano i contesti in cui i diversi significati possono attivarsi.
- *Concezione dinamica*: le parole vengono concepite come entità permeabili e il significato di ciascuna interagisce con il significato delle parole adiacenti.

Principi di interazione semantica:

- Co-composizione: il significato di un verbo può essere determinato dai suoi argomenti. Ciascun verbo ha una parte di base che non cambia, ma viene ridefinita e specializzata dal suo complemento.
- Forzatura o conversione di tipo: un verbo in combinazione con un nome specifico lo spinge a significare determinate cose anche variandone il tipo semantico.
- Legamento selettivo: l'aggettivo può selezionare una specifica porzione del significato del nome.

3

WordNet e FrameNet

3.1 WordNet

Definizione 3.1.1: WordNet

WordNet è un sistema di riferimenti lessicali online il cui design è ispirato dalle teorie psicolinguistiche sulla memoria lessicale umana. Nomi, verbi e aggettivi sono organizzati in insiemi di sinonimi ognuno rappresentante uno specifico concetto lessicale.

Problemi dei dizionari "classici":

- Vengono messe insieme parole che sono simili per posizionamento di lettere.
- Parole con significati simili o collegati sono molto sparse.
- Non esiste un'alternativa ovvia per permettere al lettore di cercare una specifica parola.

Note:-

Per i computer è molto facile cercare in pochi istanti tra migliaia di termini, ma sarebbe uno spreco limitarsi a quello.

1985, Princeton:

- Un gruppo di psicologi e linguisti inizia sviluppare un database lessicale.
- L'idea iniziale era quella di fornire un aiuto nella ricerca nei dizionari a livello *concettuale*.
- Avrebbe dovuto essere usato in congiunzione con un dizionario online convenzionale.

WordNet divide il lessico in 4 categorie:

- Nomi: organizzati come gerarchie.
- Verbi: organizzati come una serie di relazioni.
- Aggettivi.
- Avverbi.

3.1.1 La Matrice Lessicale

Definizione 3.1.2: Parola

Una parola è un'associazione convenzionale tra un concetto lessicalizzato e un'espressione (utterance) sintattica. Per ridurre l'ambiguità si utilizzano i termini:

- Word form: per riferirsi all'espressione fisica.
- Word meaning: per riferirsi al concetto lessicalizzato espresso dalla forma.

Matrice lessicale:

- La word form rappresenta le colonne.
- Il word meaning rappresenta le righe.
- Se due entry sono nella stessa colonna le parole sono *polisemiche*.
- Se due entry sono nella stessa riga le parole sono *sinonimi*.

Word Meanings	Word Forms				
	F ₁	F ₂	F ₃	...	F _n
M ₁	E _{1,1}	E _{1,2}			
M ₂		E _{2,2}			
M ₃			E _{3,3}		
:				⋮	⋮
M _m					E _{m,n}

Figure 3.1: Matrice lessicale.

Definizione 3.1.3: Synsets

Il significato della parola M_1 può essere usato semplicemente scegliendo una forma F_n che la esprime.

Note:-

Le persone che conoscono l'inglese hanno già assimilato i concetti e ci si aspetta che siano in grado di riconoscerli dalla parole appartenenti a un synset.

Corollario 3.1.1 Gloss

Breve definizione o descrizione del senso di una parola. Se non sono presenti sinonimi il gloss funziona da synset.

3.1.2 Relazioni Semantiche

WordNet è organizzato con *relazioni semantiche*. Una relazione semantica è una relazione tra significati (rappresentati da synsets). In altre parole sono dei puntatori tra synsets. Una loro caratteristica è che sono simmetriche.

Relazioni:

- **Sinonimia:** relazione lessicale, similarità di significati. Due espressioni sono sinonimi se sostituendo l'una con l'altra non cambia il valore della frase. Tuttavia i veri sinonimi sono rari per cui si fa sempre riferimento a un determinato contesto linguistico.
- **Antonimia:** è una relazione lessicale tra le forme delle parole, ma non tra i significati delle stesse. Per esempio "ricco" è antonimo di "povero", ma non sono opposti. "Ascesa" e "caduta" sono opposti, ma non sono antonimi.
- **Iponimia:** relazione semantica del tipo *ako* (a kind of). È una relazione transitiva e asimmetrica.
- **Meronimia:** relazione *hasa* (part-whole). Anche questa relazione è transitiva e asimmetrica.

3.1.3 Sostantivi

I *nomi* in WordNet sono organizzati in definizioni che forniscono un termine sopraordinato e alcune caratteristiche:

- Attributi.
- Parti (meronomia).
- Funzioni.

Le definizioni:

- Non specificano quale senso di un termine sopraordinato sia quello appropriato.
- Non forniscono informazioni su termini coordinati.
- Le definizioni puntano verso l'alto, non in termini laterali o in iponimi.
- Un problema dei dizionari sono le definizioni circolari ($w - a$ per definire w_b e w_b per definire w_a). In WordNet si è imposta una struttura ad albero per evitare questo.

Definizione 3.1.4: Assunzione Psicolinguistica

La decisione di organizzare i nomi come un sistema di ereditarietà riflette un giudizio psicolinguistico sul lessico mentale. Ci sono evidenze che una struttura gerarchica sia più facile da concettualizzare.

Superordinati:

- I nomi superordinati possono servire come anafore per riferirsi ai loro iponimi.
- I superordinati e i loro iponimi non possono essere confrontati.
- Spiegazioni in termini di gerarchie di relazioni semantiche.

Evidenza sperimentale:

- Il *tempo di reazione* può essere usato per indicare il numero di livelli gerarchici separanti due significati.
- Esempio: le persone rispondono più in fretta a "un canarino può cantare" rispetto a "un canarino può volare".
- Il tempo di reazione varia perché è richiesto tempo extra per estrarre caratteristiche da concetti sopraordinati.

Problemi aperti:

- L'informazione generica è ereditata oppure è "salvata" in modo ridondante?
- Alcuni termini condividono lo stesso collegamento semantico, ma alcuni vengono confermati più rapidamente rispetto ad altri.
- Ma se la memoria semantica non è organizzata come un sistema di eredità allora come?
 - L'assunzione di ereditarietà è corretta, ma il tempo di reazione misura una distanza *pragmatica*, non semantica.

In WordNet:

- I nomi sono divisi in insiemi semantici che selezionano un certo numero piccolo di concetti e ognuno funziona da punto iniziale di una gerarchia (*ereditarietà multipla*).
- Ci sono 25 di questi "punti iniziali".
- Concetti generici:
 - Le gerarchie di nomi hanno un livello, nel mezzo, dove la maggior parte delle caratteristiche sono attaccate (basic-level).
 - Sopra il livello base i concetti sono brevi e generali.
 - Sotto il livello base ci sono cose più specifiche.
- Caratteristiche:
 - La struttura è generata da relazioni di iponimi e i dettagli sono dati dalle caratteristiche che distinguono i concetti.
 - Esistono 3 tipi di caratteristiche: attributi (aggettivi), parti (nomi), funzioni (verbi).

Le caratteristiche:

- *Attributi*:
 - Il valore è espresso dagli aggettivi.
 - Gli aggettivi *modificano* il significato dei nomi.
 - Gli attributi associati a un nome si riflettono negli aggettivi che possono normalmente modificare quel nome.
- *Parti*:
 - La relazione part-whole tra nomi (meronimia).
 - La meronimia ha una relazione inversa: se w_m è un meronimo di w_h allora w_h è un olonimo di w_m .
 - I meronimi sono caratteristiche distintive che gli iponimi possono ereditare e sono asimmetriche e transitivi (in maniera limitata).
- *Funzioni*:
 - Sono descrizioni di qualcosa che le istanze di un concetto fanno normalmente.
 - Tutte le caratteristiche dei nomi che sono descritte da verbi o frasi verbali.
 - Ci sono motivazioni linguistiche per cui si assume che una funzione è una caratteristica del significato.
 - Un oggetto che non è X non può essere un buon X se fa bene una funzione che viene effettuata normalmente da X.

3.1.4 Verbi

Definizione 3.1.5: Verbi

I verbi forniscono un framework relazionale e semantico per le frasi.

Sottocategorizzazioni:

- Strutture sintattiche che specificano quanti e quali tipi di argomenti un verbo può avere.
- Ogni argomento ha un ruolo semantico, una funzione del significato dell'azione.
- *selectional restrictions*: specificano le proprietà semantiche di una classe di nomi.
- Tutto ciò fa parte della voce del verbo in un dizionario mentale.

Osservazioni 3.1.1

- Ci sono molti meno verbi rispetto ai nomi.
- I verbi sono generalmente più polisemici dei nomi (i verbi più comuni sono estremamente polisemici).
- I verbi sono più flessibili e possono cambiare significato in base ai nomi con cui co-occorrono.
- Non vale l'iponimia: non si può dire che il verbo V_1 sia a kind of del verbo V_2 .
- Però è presente un concetto di *intensità*.

Definizione 3.1.6: Troponimi

Relazione per cui un verbo V_1 sia in relazione con un verbo V_2 in un determinato modo.

Gerarchia per i verbi:

- Utilizzando la troponimia è difficile realizzare una struttura ad albero come per i nomi.
- Non tutti i verbi possono essere raggruppati sotto un'unica parola iniziale.
- Si devono rappresentare con una serie di alberi indipendenti.
- Le gerarchie di verbi tendono ad avere livelli molto più ricchi rispetto ad altri.

3.1.5 Conceptual Similarity e Word Sense Disambiguation**Definizione 3.1.7: Conceptual Similarity**

Dati in input due termini si vuole fornire un punteggio numerico di similarità che ne indichi la vicinanza semantica.

Note:-

Per risolvere questo task è possibile sfruttare la struttura ad albero di WordNet.

Esistono varie misure di similarità:

- *Wu & Palmer*: $\text{cs}(s_1, s_2) = \frac{2 \cdot \text{depth(LCS)}}{\text{depth}(s_1) + \text{depth}(s_2)}^1$.
- *Shortest path*: $\text{sim}_{\text{path}}(s_1, s_2) = 2 \cdot \text{depthMax} - \text{len}(s_1, s_2)$.
- *Leacock & Chodorow*: $\text{sim}_{\text{LC}}(s_1, s_2) = -\log \frac{\text{len}(s_1, s_2)}{2 \cdot \text{depthMax}}$.

Termini vs. Sensi:

- L'input è costituito da coppie di *termini*, ma le formule richiedono *sensi*.
- Per calcolare la similarità tra due termini si prende la massima similarità tra tutti i sensi del primo termine e tutti i sensi del secondo termine. L'ipotesi è che i due termini funzionino come contesto di disambiguazione l'uno per l'altro.

Definizione 3.1.8: Word Sense Disambiguation

Il Word Sense Disambiguation è un problema aperto che consiste nell'identificare quale senso di una parola (il significato) è utilizzato in una data frase quando quella parola più sensi (è polisemica).

¹LCS è il primo antenato comune, depth misura la distanza tra la radice di WordNet e il Synset x.

WSD:

- È utile per altri tasks: traduzione, Q&A, information retrieval, text classification.
- Nella forma base un algoritmo di WSD prende in input una parola in un contesto insieme ai suoi potenziali sensi e restituisce in output il senso corretto.

Approcci per classi di features:

- *Collocation*: sono parole o frase in una relazione in cui la posizione è importante.
- *Bag-of-words*: sono parole in un insieme non ordinato.

Definizione 3.1.9: Algoritmo di Lesk

Algoritmo per il WSD che si basa sul contesto, ossia le parole vicine a quella da disambiguare, e sull'overlapping.

Note:-

Questo approccio semplicistico può essere espanso utilizzando un corpus: si aggiungono tutti i contesti di parole appartenenti a un corpus con i rispettivi sensi.

3.2 FrameNet

Ipotesi:

- Le persone comprendono le cose effettuando operazioni mentali su ciò che conoscono già.
- Tale conoscenza può essere descritta in *pacchetti di informazioni*.

3.2.1 La Teoria dei Frame

Definizione 3.2.1: FrameNet

FrameNet è un progetto di costruzione lessicale per l'inglese che collega le parole ai loro significati:

- Registrando i modi in cui le frasi sono costruite attorno a essi.
- Usando le evidenze trovate in testi in inglese moderno.

Corollario 3.2.1 Frame

I Frame funzionano utilizzando situazioni "stereotipate". I significati dei concetti dipendono dai frame a cui sono collegati.

Domanda 3.1

Ma come si fa con la polisemia?

- Invece di usare parole si usano *Unità lessicali* (LUs) coppie di parole e sensi.
- In WordNet differenti LUs appartengono a diversi synsets.
- In FrameNet differenti LUs sono solitamente appartenenti a frame diversi.
- Se una parola comunica differenti significati in diversi contesti e la differenza non è evidente dal contesto forse la parola ha più di un significato.
- Alcuni, ma non tutti i verbi che indicano il parlare hanno un *uso cognitivo* identificando fonti o credenze.
- Patterns complementari dovrebbero andare con un particolare significato di una parola.
- Se un verbo ha due eventi derivati ed entrambi hanno significati diversi che si trovano anche nel verbo allora è il verbo stesso a essere polisemico.

Corollario 3.2.2 Frame Element

Si sviluppa un vocabolario descrittivo per i componenti di ogni frame. Questi frame element (FE) sono usati per etichettare i costituenti di una frase nel frame.

3.2.2 Il Frame "Revenge"

Il concetto *Revenge* rappresenta una situazione in cui:

- *A* ha fatto qualcosa per ferire *B*.
- *B* decide di far qualcosa per ferire *A*.
- L'azione di *B* è portata a termine indipendentemente da conseguenze legali o istituzionali.

Vocabolario per Revenge:

- *Nomi:* revenge, vengeance, reprisal, retaliation, retribution.
- *Verbi:* avenge, revenge, retaliate (against), get back (at), get even (with), pay back.
- *Adjectives:* vengeful, vindictive.
- *V + N:* take revenge, exact retribution, wreak vengeance.

FE per Revenge:

- A causa di qualche lesione a qualcosa o qualcuno di importante per un vendicatore (forse se stesso), il vendicatore infligge una punizione all'autore del reato. l'autore del reato è la persona responsabile dell'infortunio.
- Lista:
 - Avenger.
 - Offender.
 - Injury.
 - Injury_party.
 - Punishment.

FN work:

- Vengono selezionate le frasi che mostrano i maggiori contesti sintattici.
- I costituenti delle frasi che esprimono questi FEs sono etichettati con i nomi associati agli FEs.

Due tipologie di target:

- *Predicati:* parole che evocano frames, creano contesti per le informazioni da inserire. L'obiettivo dell'annotazione è trovare gli argomenti.
- *Fillers:* parole che soddisfano ruoli semantici nei frames evocati dai predicati. L'obiettivo dell'annotazione è identificare il frame e FE della frase.

Note:-

Tipicamente parole diverse nello stesso frame mostrano variazione in come gli FE sono realizzati grammaticalmente.

Definizione 3.2.2: Valenza

La valenza è il numero e il tipo degli argomenti controllati da un predicato.

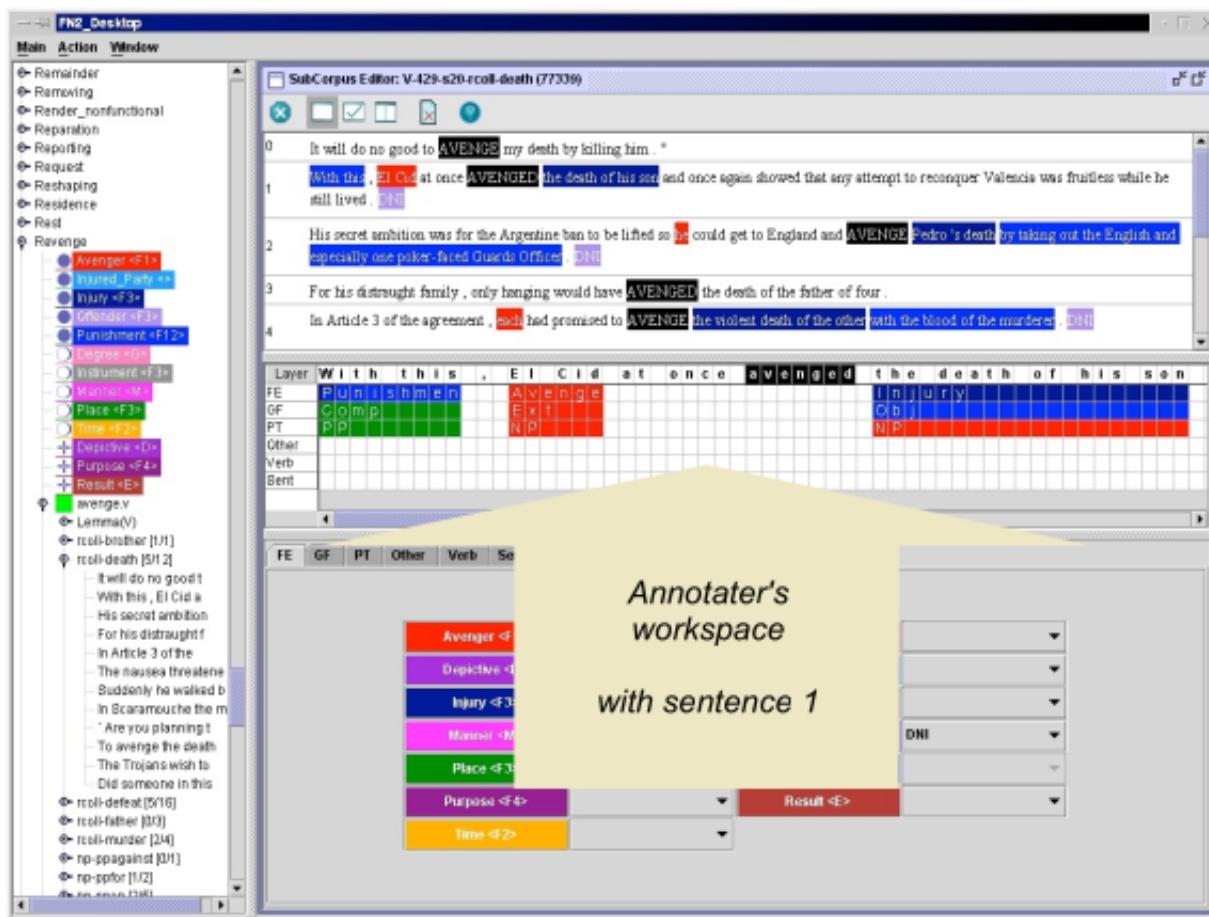


Figure 3.2: Esempio di visualizzazione di FrameNet con il Frame Revenge.

La valenza può essere di vario tipo:

- Impersonale.
- Intransitiva.
- Transitiva.
- Ditransitiva.
- Tritransitiva.

Domanda 3.2

Nel frame Revenge attraverso quali significati sintattici è realizzato l'offender?

- Come un oggetto diretto.
- Con la preposizione *on*.
- Con *against*.
- Con *with*.
- Con *at*.

Domanda 3.3

Quale FE è espressa dall'oggetto di avenge?

- L'injured_party.
- L'injury.

Altre definizioni:

- *Copertura lessicale*: tutte le parole importanti associate con ogni frame.
- *Combinatoria*: tutti i patterns sintattici in cui ogni parola serve per esprimere il frame.
- *Dati di frequenza*: non sono collezionati direttamente da FrameNet.

3.2.3 Frame Semantics e Text Understanding

Information extraction:

- Si vuole progettare un'applicazione per estrarre informazioni dalle notizie di un giornale.
- I dati di FrameNet dovrebbero poter essere usati per:
 - WSD.
 - Composizione semantica.
 - Scelta tra possibili alternative analisi sintattiche.
 - Attivazione di un vocabolario topic-related.

Risoluzione dell'anafora:

- *Pronominale*: ci si riferisce a un referente con un pronome.
- Frasi nominali definite: l'antecedente si riferisce a una frase con "<the><nounphrase>".
- Quantificatori/Ordinali: l'anafora è quantificata come "uno" e ordinalizzata come "primo".

Principi di utilizzo:

- Si può scegliere una parola per capire quali frame possano essere attivati da quella parola.
- Identificati i ruoli semanticci si cerca di accoppiare le necessità semantiche di ogni frame attivato con le porzioni della frase in input.
- Il frame che si adatta meglio è quello scelto.

Disambiguazione:

- I significati possono essere inferiti mediante un criterio di coerenza.
- FN non li fornisce direttamente, ma offre descrizioni di frame collegati e legami semanticci tra le parole per facilitare i giudizi.

Blocchi di più parole:

- L'analisi non può procedere sulla base delle parole prese una alla volta.
- È riconosciuto lo stretto legame tra le parole.

Altre cose:

- L'autore di un articolo può aggiungere informazione testimoniale a una parte della descrizione.
- Esistono frame metalinguistici.

4

Natural Language ToolKit (NLTK)

Note:-

Dato che si parla di python avrei evitato di prendere appunti su questa merda, ma per completezza lo faccio lo stesso.

Definizione 4.0.1: NLTK

NLTK è un libro e una libreria di python che fornisce molti pacchetti di text processing e un gran numero di datasets. Permette di effettuare molto facilmente varie operazioni di tokenizzazione, parsing, stemming, etc.

4.1 Introduzione ai Tasks di NLP

4.1.1 Tokenizzazione

Definizione 4.1.1: Tokenizzazione

Tokenizzare una stringa significa separarla in parole o frasi:

- Word tokenization: per identificare le parole, la più piccola unità di testo con un significato.
- Sentence tokenization: un testo è diviso nelle frasi che contiene.

4.1.2 Stop-Words Filtering

Definizione 4.1.2: Stop-Words Filtering

Alcune parole non hanno significato di per sé ma sono solo congiunzioni, in tal casi devono essere rimosse.

Osservazioni 4.1.1

- Le *context words* forniscono informazioni sui topics presenti nell'articolo.
- Caratterizzano lo stile di scrittura.

```
from nltk.tokenize import sent_tokenize, word_tokenize

example_string = "IT was 2 p.m. on the afternoon of May 7, 1915. The Lusitania had been struck and was sinking rapidly, while the boats were being launched with all possible speed. The women and children were being lined up awaiting their turn. Some still clung desperately to husbands and fathers. One girl stood alone, slightly apart from the rest. She was quite young, not more than eighteen. She did not seem afraid, and her grave, steadfast eyes looked straight ahead."

sent_tokenize(example_string)
```

```
['IT was 2 p.m. on the afternoon of May 7, 1915.', 'The Lusitania had been struck and was sinking rapidly, while the boats were being launched with all possible speed.', 'The women and children were being lined up awaiting their turn.', 'Some still clung desperately to husbands and fathers.', 'One girl stood alone, slightly apart from the rest.', 'She was quite young, not more than eighteen.', 'She did not seem afraid, and her grave, steadfast eyes looked straight ahead.]
```

Figure 4.1: Tokenizzazione.

- listing stopwords for all available languages

```
import nltk
from nltk.corpus import stopwords

nltk.download('stopwords')
print(stopwords.fileids())
```

```
['arabic', 'azerbaijani', 'basque', 'bengali', 'catalan', 'chinese', 'danish', 'dutch', 'english', 'finnish', 'french', 'german', 'greek', 'hebrew', 'hinglish', 'hungarian', 'indonesian', 'italian', 'kazakh', 'nepali', 'norwegian', 'portuguese', 'romanian', 'russian', 'slovene', 'spanish', 'swedish', 'tajik', 'turkish']
```

Figure 4.2: Meccanismo per elencare le Stop-Words.

4.1.3 Stemming

Definizione 4.1.3: Stemming

Lo stemming consiste nel ridurre le parole alla loro radice ossia il nucleo delle parole.

Note:-

Per esempio "Helping" e "Helper" condividono "Help".

Possibili errori:

- *Understemming*: due parole collegate dovrebbero essere ridotte alla stessa ma non lo sono.
- *Overstemming*: due parole non collegate sono ridotte alla stessa.

```

from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

port_stem = PorterStemmer()

example_string = "The Lusitania had been struck and was sinking rapidly, while
the boats were being launched with all possible speed."

words = word_tokenize(example_string)
stemmed_words = [port_stem.stem(word) for word in words]

print(stemmed_words)

```

[['the', 'lusitania', 'had', 'been', 'struck', 'and', 'wa', 'sink',
'rapidli', ',', 'while', 'the', 'boat', 'were', 'be', 'launch',
'with', 'all', 'possibl', 'speed', '.']]

Figure 4.3: Stemming.

PORTER	SNOWBALL
<pre> ['the', 'lusitania', 'had', 'been', 'struck', 'and', 'wa', 'sink', 'rapidli', ',', 'while', 'the', 'boat', 'were', 'be', 'launch', 'with', 'all', 'possibl', 'speed', '.'] </pre>	<pre> ['the', 'lusitania', 'had', 'been', 'struck', 'and', 'was', 'sink', 'rapid', ',', 'while', 'the', 'boat', 'were', 'be', 'launch', 'with', 'all', 'possibl', 'speed', '.'] </pre>

Figure 4.4: Confronto tra due stemmer di NLTK.

4.1.4 PoS Tagging

Definizione 4.1.4: Part of Speech Tagging

Il Part of Speech Tagging è il task che consiste nell'etichettare le parole con la corrispettiva parte del discorso.

tagset	
<pre> import nltk nltk.download('tagsets') # further information at the URL # https://www.nltk.org/data.html nltk.help.upenn_tagset() </pre>	CC: conjunction, coordinating CD: numeral, cardinal DT: determiner IN: preposition or conjunction, subordinating JJ: adjective or numeral, ordinal VB: verb, base form VBD: verb, past tense VBG: verb, present participle or gerund VBN: verb, past participle VBP: verb, present tense, not 3rd person singular VBZ: verb, present tense, 3rd person singular

Figure 4.5: Tagset.

4.1.5 Lemmatizzazione

Definizione 4.1.5: Lemmatizzazione

La lemmatizzazione riduce le parole alla loro forma canonica (lemma).

Corollario 4.1.1 Lemma

Un lemma è la forma canonica che si può trovare nel dizionario. Un lemma è una parola che rappresenta un intero gruppo di parole chiamato lesema.

```

import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
nltk.download('punkt')

lemmatizer = WordNetLemmatizer()
example_string = "The Lusitania had been struck and was sinking rapidly, while the boats were being launched with all possible speed."
words = word_tokenize(example_string)
print(words)
lemmatized_words = [lemmatizer.lemmatize(word) for word in words]
print(lemmatized_words)

```

['The', 'Lusitania', 'had', 'been', 'struck', 'and', 'was', 'sinking', 'rapidly', 'while', 'the', 'boats', 'were', 'being', 'launched', 'with', 'all', 'possible', 'speed', '.']
 ['The', 'Lusitania', 'had', 'been', 'struck', 'and', 'be', 'sink', 'rapid', 'while', 'the', 'boat', 'be', 'be', 'launch', 'with', 'all', 'possibl', 'speed', '.']

Figure 4.6: Lemmatizzazione.

Note:-

La performance di un lemmatizer può essere aumentata aggiungendo PoS tagging.

4.1.6 NER

Definizione 4.1.6: Named Entity Recognition

Named Entities sono frasi con nomi che si riferiscono a specifiche località, persone, organizzazioni, etc. Il task NER consiste nel trovare le Named Entities in un testo e determinarne il tipo.

4.2 SpaCy

Definizione 4.2.1: SpaCy

SpaCy è una libreria opensource per NLP in python. Offre varie funzioni:

- Tokenizzazione.
- PoS tagging.
- Lemmatizzazione.
- Parsing a dipendenze.
- Sentence Boundary Detection (SBD).
- NER.

Funzionalità avanzate:

- Entity linking: disambiguare entità testuali in identificatori univoci in un'abase di conoscenza.
- Similarity: comparare parole, testi, documenti, etc.
- Text classification: assegnare categorie o etichette a un documento o a parti di esso.
- Matching a regole: trovare sequenze di token basandosi su annotazioni linguistiche (simile alle regex).
- Training: aggiornare e migliorare predizioni di modelli statistici.
- Serializzazione: salvare oggetti in file o byte.

Corollario 4.2.1 Modello

Alcune features di SpaCy sono indipendenti, mentre altre richiedono una pipeline per essere caricate. Una pipeline addestrata consiste di molteplici componenti che usano modelli statistici su dati etichettati.

Note:-

Sono inclusi diversi linguaggi.

Corollario 4.2.2 Annotazioni Linguistiche

Le annotazioni linguistiche offrono una visione delle strutture grammaticali e sintattiche.

4.2.1 La Pipeline

Chiamando `nlp` su un testo SpaCy effettua tokenizzazione, altri processi e infine restituisce in output un `Doc`.

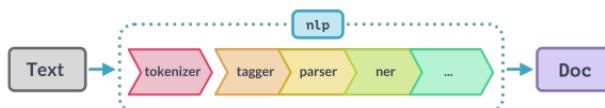


Figure 4.7: La pipeline.

```
nlp = spacy.load("en_core_web_sm")
print(nlp.pipe_names)
print(nlp.pipeline)
```

```
['tok2vec', 'tagger', 'parser', 'attribute_ruler', 'lemmatizer', 'ner']
[('tok2vec', <spacy.pipeline.tok2vec.Tok2Vec object at
0x7da6e8c4d060>), ('tagger', <spacy.pipeline.tagger.Tagger object at
0x7da6e8c4d900>), ('parser',
<spacy.pipeline.dep_parser.DependencyParser object at 0x7da6e8bffa00>),
('attribute_ruler', <spacy.pipeline.attributeruler.AttributeRuler
object at 0x7da6e99277c0>), ('lemmatizer',
<spacy.lang.en.lemmatizer.EnglishLemmatizer object at 0x7da6e98c7ac0>),
('ner', <spacy.pipeline.ner.EntityRecognizer object at
0x7da6e8bff8b0>)]
```

Figure 4.8: Ottenerne informazioni sulla pipeline di SpaCy.

Note:-

Nel processare grandi volumi di testo questi modelli sono più efficienti con insiemi di testi.

Osservazioni 4.2.1

- Per aumentare la velocità si può scegliere di disabilitare o escludere una parte della pipeline.
- `disable`: i componenti sono caricati, ma non utilizzati.
- `exclude`: i componenti non sono caricati.

5

Rappresentazioni di Significato

5.1 Modelli Probabilistici

Domanda 5.1

Perché assegnare una probabilità a una frase:

- Traduzione.
- Correzione di spelling.
- Speech recognition.
- Summarization.
- Q&A.

Definizione 5.1.1: Modello Probabilistico del Linguaggio (LM)

Un modello che assegna probabilità a una sequenza di parole.

5.1.1 Ngrams

Il modello più semplice è **n-grams**, ossia sequenze di n word:

- 2-gram (bigramma): sequenza di due parole.
- 3-gram (trigramma): sequenza di tre parole.

Note:-

Gli n-grams possono essere usati per stimare la probabilità dell'ultima parola date le parole precedenti assegnando una probabilità all'intera sequenza.

Questa probabilità può essere stimata contando:

- Dato un corpus molto grande contare il numero di volte in cui si vede una determinata parola.
- Predire una parola w data una storia h equivale a calcolare $P(w|h)$.

- Rappresentiamo la probabilità di una variabile casuale X_i assumendone il valore. E.g. "the" rappresentato come $P(\text{the})$ e non come $P(X_i = \text{the})$.
- Una sequenza di N parole può essere rappresentata come $w_1 \dots w_n$ e la relativa probabilità come $P(w_1 \dots w_n)$.

Domanda 5.2

Come calcolare le probabilità per un'intera sequenza?

$$P(X_1, \dots, X_n) = P(X_1)P(X_2 | X_1)P(X_3 | X_{1:2}) \cdots P(X_n | X_{1:n-1}) = \prod_{k=1}^n P(X_k | X_{1:k-1})$$

Note:-

Tuttavia il linguaggio è creativo e quindi una particolare contesto potrebbe non essersi mai presentato.

Intuizione:

- La storia può essere approssimata considerando solo le ultime poche parole (assunzione di Markov).
- $P(w_n | w_{1:n-1}) \approx P(w_n | w_{n-1})$

Definizione 5.1.2: Bigramma

In un bigramma la storia è ristretta solo alla parola precedente.

$$P(w_1 w_2 | w_i) = P(w_1) \times P(w_2 | w_1) \times \cdots \times P(w_i | w_{i-1})$$

La probabilità condizionale è: $P(w_i | w_{i-1}) = \frac{|w_{i-1} * w_i|}{|w_{i-1}|}$.

Generazione di sequenze:

- Solo specifiche sequenze possono essere costruite con un modello di linguaggio senza ridurre le probabilità a 0.
- Sequenze che compaiono nel training set.
- Si possono generare sequenze di lunghezza arbitraria ma non c'è garanzia che saranno grammaticalmente corrette.

Definizione 5.1.3: Maximum Likelihood Estimation

Si ottiene un conteggio da un corpus e lo si normalizza tra 0 e 1. Le frequenze relative sono un mezzo per stimare le probabilità.

Conoscenza Sintattica:

- Le probabilità raccolgono le regolarità sintattiche. Esempio: "to" ha un'alta probabilità di essere seguito da un verbo.

Note:-

Dai bigrammi si può passare a trigrammi, tetragrammi, etc.

Log probabilities:

- Le probabilità sono minori o uguali a 1 quindi più probabilità moltiplichiamo assieme più il prodotto diventa piccolo (rischio di underflow).
- Per cui vengono impiegate le probabilità logaritmiche (sommare è più veloce che moltiplicare): $p_1 \times p_2 = \log(p_1) + \log(p_2)$.

5.1.2 Valutare un LM

Tipi di valutazione:

- **Valutazione Estrinseca:** usare gli LMs in applicazioni e vedere come cambiano le performance.
- **Valutazione Intrinseca:** misurare la qualità di un modello indipendentemente da una data applicazione:
 - Le probabilità di un ngram sono acquisite da un training set e la sua qualità può essere testata su un test set.
 - Questo tipo di valutazione implica il partizionamento dei dati in due sets distinti e comparare quanto il modello allenato si adatti al test set.

Definizione 5.1.4: Perplessità

La perplessità (perplexity) misura quanto bene un modello linguistico predice una sequenza di parole non vista.

Consideriamo una sequenza di parole di lunghezza k :

$$W = \{w_1, w_2, \dots, w_k\}$$

Dato un modello linguistico LM, la probabilità della sequenza W è:

$$\text{LM}(W) = \prod_{i=1}^k \text{LM}(w_i | w_{1:i-1})$$

La log-probabilità media è:

$$\frac{1}{k} \sum_{i=1}^k \log \text{LM}(w_i | w_{1:i-1})$$

La perplessità della sequenza W è quindi definita come:

$$\text{PPL}(\text{LM}, W) = \exp \left\{ -\frac{1}{k} \sum_{i=1}^k \log \text{LM}(w_i | w_{1:i-1}) \right\}$$

Note:-

Valori di PPL più bassi indicano che il modello è più abile nel predire la sequenza, ovvero assegna probabilità più alte alle parole corrette.

5.1.3 Smoothing

Domanda 5.3

Cosa succede se delle parole sono nel nostro vocabolario, ma appaiono in un test set in un nuovo contesto?

Note:-

Si vuole impedire che quelle parole abbiano zero probabilità.

Definizione 5.1.5: Laplace Smoothing

La **Laplace smoothing** è una tecnica di smoothing che consiste nell'aggiungere uno a ciascun conteggio prima della normalizzazione in probabilità.

Per i **unigrammi**, la stima di massima verosimiglianza per la probabilità della parola w_i è:

$$P(w_i) = \frac{c_i}{N}$$

dove c_i è il conteggio della parola w_i e N è il numero totale di token.

Con la Laplace smoothing, si aggiunge 1 a ogni conteggio, e si regola il denominatore per tener conto delle V parole nel vocabolario:

$$P_{\text{Lap}}(w_i) = \frac{c_i + 1}{N + V}$$

In alternativa, si può descrivere l'effetto dello smoothing attraverso un *conteggio corretto*:

$$c_i^* = (c_i + 1) \cdot \frac{N}{N + V}$$

Questo conteggio corretto viene poi normalizzato su N per ottenere una nuova probabilità:

$$P_i^* = \frac{c_i^*}{N}$$

Lo smoothing può anche essere visto come una forma di **discounting**, cioè una riduzione di alcuni conteggi per redistribuire la massa di probabilità anche su eventi precedentemente con probabilità nulla. Il *discount relativo* d_c è:

$$d_c = \frac{c^*}{c}$$

Corollario 5.1.1 Normalizzazione

Le probabilità dei bigrammi si ottengono normalizzando ciascun conteggio di bigramma rispetto al conteggio del primo elemento del bigramma:

$$P(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n)}{C(w_{n-1})}$$

Con la **Laplace smoothing** (add-one), si aggiunge 1 al numeratore e si incrementa il denominatore con V , il numero di parole distinte nel vocabolario:

$$P_{\text{Lap}}(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$$

Questo garantisce che nessun bigramma abbia probabilità nulla, anche se non è stato osservato nel corpus di addestramento.

Esempio: se $V = 1446$, ogni conteggio unigramma nel denominatore sarà aumentato di 1446.

5.2 Rappresentazioni Vettoriali

5.2.1 Modello di Spazio Vettoriale

Alcuni termini:

- **Documento:** un'unità di testo indicizzata nel sistema e disponibile per la ricerca.
- **Collezione:** un insieme di documenti utilizzati per soddisfare le richieste dell'utente.
- **Termine:** l'elemento che occorre nella collezione.
- **Interrogazione:** la richiesta dell'utente espressa come insieme di termini.

Definizione 5.2.1: Vector Space

Il vector space è una collezione di vettori con una certa dimensione. La loro dimensione è $|V|$, ossia la dimensione del vocabolario di riferimento.

Information Retrieval:

- Le matrici term-document sono state sviluppate per trovare documenti simili¹.
- Task: trovare il documento d nella collezione D che matchi al meglio una query q .
- Per risolvere questo task bisogna comparare quanto siano simili due vettori²

5.2.2 Confrontare Vettori**Definizione 5.2.2: Cosine Similarity**

La metrica più comune è la similarità del coseno. Si basa sul dot product: $v \cdot w = \sum_{i=1}^N v_i w_i$.

Tuttavia questa metrica tende a favorire i vettori lunghi mentre noi vogliamo una metrica che sia valida a prescindere dalla lunghezza. Introduciamo quindi il vettore lunghezza: $|v| = \sqrt{\sum_{i=1}^N v_i^2}$.

La similarità del coseno normalizza il prodotto scalare per le lunghezze dei vettori:

$$\cos(v, w) = \frac{v \cdot w}{|v| |w|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

Il valore risultante è compreso tra -1 e 1 , dove 1 indica che i vettori hanno la stessa direzione, 0 che sono ortogonali, e -1 che puntano in direzioni opposte. Tuttavia le frequenze non possono essere negative.

Note:-

Il dot product tende a essere alto quando i vettori sono alti nelle stesse dimensioni.

Per attribuire il giusto peso ai termini sono necessari due fattori:

- La frequenza in un documento.
- La non frequenza in un intera collezione.

¹Che contengono parole simili.

²Sono molto sparsi (pieni di 0) quindi si possono trovare modi efficienti per salvarli.

Definizione 5.2.3: Inverse Document Frequency

L'idea di base è che i termini presenti in pochi documenti sono utili per distinguere quei documenti dal resto della collezione. L'Inverse Document Frequency (IDF) è definito per mezzo del rapporto $\frac{N}{n_i}$ dove N è il numero totale di documenti nella collezione e n_i è il numero di documenti in cui il termine i occorre. Dato l'elevato numero di documenti in varie collezioni il rapporto è schiacciato con una funzione logaritmica: $idf_i = \log(\frac{N}{n_i})$.

Utilizzando lo schema tf-idf il peso del termine i nel vettore del documento j è calcolato come il prodotto della sua frequenza totale in j per il logaritmo idf nella collezione:

$$\text{sim}(\vec{q}, \vec{d}) = \frac{\sum_{w \in q, d} \text{tf}_{w,q} \cdot \text{tf}_{w,d} \cdot (\text{idf}_w)^2}{\sqrt{\sum_{q_i \in q} (\text{tf}_{q_i, q} \cdot \text{idf}_{q_i})^2} \times \sqrt{\sum_{d_i \in d} (\text{tf}_{d_i, d} \cdot \text{idf}_{d_i})^2}}$$

Note:-

I termini che ricorrono in tutti i documenti non sono utili: stop words.

5.2.3 Il Metodo di Rocchio

Alcuni classificatori lineari consistono in un profilo esplicito della categoria.

Definizione 5.2.4: Centroidi

Un centroide di un insieme di X vettori di termini è definito come:

$$\vec{t}_X := \frac{1}{|X|} \sum_{t_d^i \in X} \vec{t}_d^i$$

Un classificatore costruito usando il metodo di Rocchio ricompensa:

- La vicinanza di un documento di test al centroide degli esempi di training positivi.
- La distanza di un documento di test dal centroide degli esempi di training negativi.

Definizione 5.2.5: Metodo di Rocchio

Il vettore della classe è definito come:

$$\vec{c}_i = \langle f_{1i}, \dots, f_{|T|i} \rangle$$

- Per ogni classe c_i (con i che varia tra le classi) calcoliamo il peso della k -esima feature f come:

$$f_{ki} = \beta \cdot \sum_{d_j \in POS_i} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{d_j \in NEG_i} \frac{w_{kj}}{|NEG_i|}$$

dove:

$$POS_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = Vero\}$$

$$NEG_i = \{d_j \in Tr \mid \Phi(d_j, c_i) = Falso\}$$

- β e γ sono parametri di controllo che permettono di regolare l'importanza relativa degli esempi positivi e negativi.
- Se β è impostato a 1 e γ a 0, il profilo di c corrisponde al centroide dei suoi esempi di training positivi.

5.2.4 WordToVec

Definizione 5.2.6: Embeddings

Gli Embeddings sono vettori:

- Corti: d compresa tra 50 e 1000.
- Densi: invece che essere pieni di 0.

Osservazioni 5.2.1

- I vettori densi funzionano meglio in ogni task NLP rispetto ai vettori sparsi.
- Un possibile motivo è che uno spazio minore contribuisce a evitare l'overfitting.

Definizione 5.2.7: Skip-Gram

Gli Skip-Gram mirano a imparare rappresentazioni numeriche che catturano le ipotesi di distribuzione. L'obiettivo del training è predirre il contesto in cui una determinata parola è probabile che occorra.

Corollario 5.2.1 Skip-Gram with Negative Sampling (SGNS)

L'idea è quella che usiamo i testi come dati per il training supervisionato隐式的:

1. Tratta la parola target e i suoi vicini come esempi positivi.
2. Vengono scelte casualmente altre parole fuori dal vicinato per ottenere esempi negativi.
3. Si utilizza la regressione logistica per addestrare un classificatore per distinguere i due casi precedenti.
4. Viene utilizzato per apprendere i pesi degli embeddings.

5.2.5 Altri Embeddings

Problemi di word2vec:

- word2vec non ha buoni modi di gestire le parole sconosciute.
- Un altro problema è la sparsità delle parole.

Definizione 5.2.8: Fasttext

Fasttext è un'estensione di word2vec. Utilizza un modello si sottoparole rappresentando ogni parola come sé stessa più un insieme di n-grams. Uno skipgram è appreso per ognuno di questi n-grams e le parole sono rappresentate dalla somma di tutti gli embeddings.

Note:-

Un altro embedding molto utilizzato è GloVe.

6

Un'Introduzione ai Transformers

6.1 Contextualized Embeddings

Definizione 6.1.1: Meaning Conflation Problem

Gli algoritmi di distribuzione basati sulla similarità racchiudono tutti i sensi di una parola in un unico embedding (anche quando sono diversi).

Note:-

Per risolvere questo problema sono state introdotte le reti di transformers/transfomers network (TNs).

Definizione 6.1.2: Contextualized Embeddings

Le TNs hanno lo scopo di apprendere contextualized embeddings i cui significati cambiano in base al contesto in cui i termini appaiono. Ogni parola riceve un contextualized embedding che è la media pesata di qualche embeddings di input (simile a word2vec)

Le TNs consistono di uno stack a più strati:

- L'embeddings di output per lo strato i diventa l'input embeddings dello strato $i + 1$.
- Solitamente gli strati variano tra 2 e 24.
- Più livelli performano meglio permettendo alla rete di apprendere funzioni più complesse.

6.1.1 Architettura

L'input entra nell'encoder del transformer attraverso uno strato di attention e uno trato di FeedForward Network (FFN). L'output entra nel decoder attraverso due strati di attention e uno di FFN.

Transformer originale:

- Il modello originale era uno stack a 6 strati.
- L'output dello strato l è l'input dello strato $l + 1$ finché non è stata raggiunta la predizione finale.

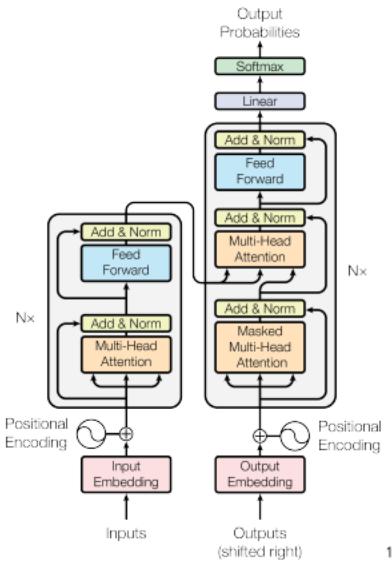


Figure 6.1: Architettura attuale di un transformer.

Definizione 6.1.3: Encoder

L'encoder è composto da uno stack di 6 strati identici ognuno di esso composto da due sottostrati:

- Un meccanismo multi-head di self-attention.
- Una semplice FFN.

Attorno a ogni sottostrato sono applicati:

- Residual Connection: si somma l'input originale del sottostrato con il suo output.
- Normalizzazione.

Note:-

Per facilitare l'applicazione della residual connection ogni sottostrato e strato produce vettori con la stessa dimensione.

Definizione 6.1.4: Decoder

Il decoder è composto da uno stack di 6 strati identici, ognuno dei quali contiene tre sottostrati:

- Un meccanismo multi-head di self-attention con mascheramento per impedire l'accesso a posizioni future nella sequenza.
- Un meccanismo multi-head di attenzione sull'output dell'encoder.
- Una semplice FFN.

Attorno a ogni sottostrato sono applicati:

- Residual Connection: si somma l'input originale del sottostrato con il suo output.
- Normalizzazione.

Note:-

Il mascheramento nella self-attention del decoder assicura che la predizione alla posizione i dipenda solo dalle posizioni precedenti, non da quelle future.

Ogni strato implementa una sequenza di operazioni tra cui:

- Aggiungere informazione posizionale all'input degli embeddings: l'embedding di ogni parola cambia in base alla sua posizione nel testo in input.
- *Self-attention*: Vengono generati embedding che sono la media pesata degli embedding nello stage precedente.
- Somma e normalizzazione.

Tipi di Transformers:

- *Encoder-only*: convertono una sequenza di testo in input in una rappresentazione numerica dove la rappresentazione calcolata per un dato token dipende sia dal contesto sinistro che da quello destro (bidirectional attention). BERT e derivati appartengono a questa categoria.
- *Decoder-only*: completano un dato input predicendo iteramente la parola più probabile (autoregressive attention). In questa categoria rientrano i modelli GPT.
- *Encoder-Decoder*: utilizzati per modellare mapping complessi, solitamente in traduzione e sommarizzazione. E.g. BART.

6.1.2 Strati

Definizione 6.1.5: Feed-Forward Sublayer

Il sottostrato Feed-Forward (FFN) viene applicato dopo i meccanismi di attenzione e ha due obiettivi principali:

- Permette la trasformazione non lineare delle rappresentazioni, introducendo complessità nel modello.
- Aiuta ad apprendere caratteristiche astratte, come ruoli sintattici, classi semantiche o categorie latenti (es. tipi di verbi, entità nominate).

Viene applicato posizione per posizione in modo indipendente, ed è composto da due layer lineari separati da una funzione di attivazione non lineare (tipicamente ReLU).

Note:-

Apprende attraverso backpropagation, ma solitamente non da etichette a livello di token.

Questo strato:

- Si occupa di apprendere trasformazioni intermedie della rappresentazione dei token che sono utili per obiettivo finale.
- Per esempio:
 - Ruoli sintattici: soggetto, oggetto, etc.
 - Ruoli semanticci: agente, paziente, luogo, etc.
 - Enfatizzare caratteristiche salienti: polarità, tensione, etc.
 - Creare clusters.

Definizione 6.1.6: Self-attention

La self-attention è lo strumento che permette di associare chi sta parlando di cosa.

Note:-

Esempio: "The animal didn't cross the street because it was too tired.", la self-attention ci dice che "it" si riferisce ad "animal".

Definizione 6.1.7: Positional Encoding

È un modo per tener conto dell'ordine delle parole in una sequenza di input. Il transformer aggiunge un vettore a ogni input embedding. Questi vettori seguono uno specifico pattern che il modello impara per determinare la posizione di ogni parola o la differenza tra parole diverse in una sequenza.

Definizione 6.1.8: Encoder-Decoder Attention

Il meccanismo di attenzione encoder-decoder permette al decoder di focalizzarsi sulle parti rilevanti dell'input. In questo sottostrato:

- I vettori di Key (K) e Value (V) sono ottenuti dall'output dell'encoder.
- I vettori di Query (Q) sono ottenuti dal layer inferiore del decoder.

Funziona come un'attenzione multi-head standard, ma incrocia le rappresentazioni:

- le Query vengono dal decoder, mentre Keys e Values provengono dall'encoder.

Questo consente al decoder di accedere direttamente all'informazione contestuale dell'input durante la generazione dell'output.

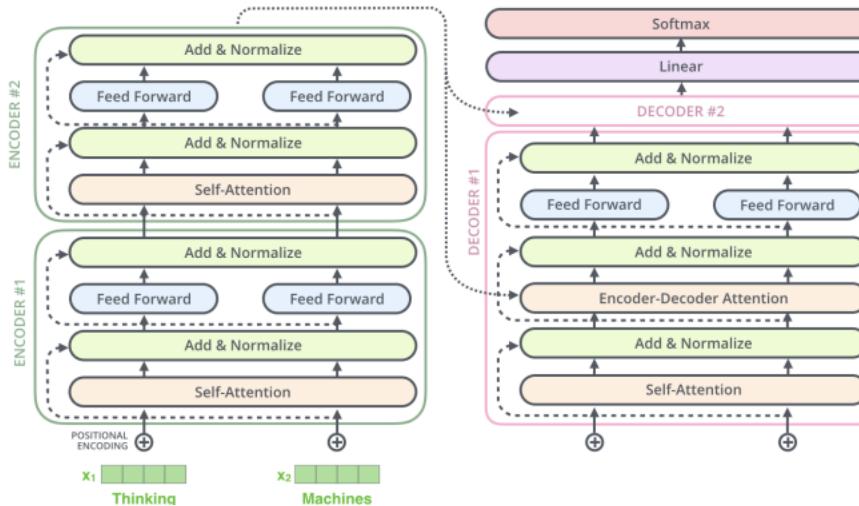


Figure 6.2: Encoder-Decoder.

6.2 Training

Quando si va a fare training su un dataset etichettato si può valutarne l'output confrontandolo con quello corretto.

Definizione 6.2.1: Loss Function

La loss function (funzione di perdita) misura quanto le predizioni del modello si allontanano dalla realtà (cioè, dalla sequenza corretta).

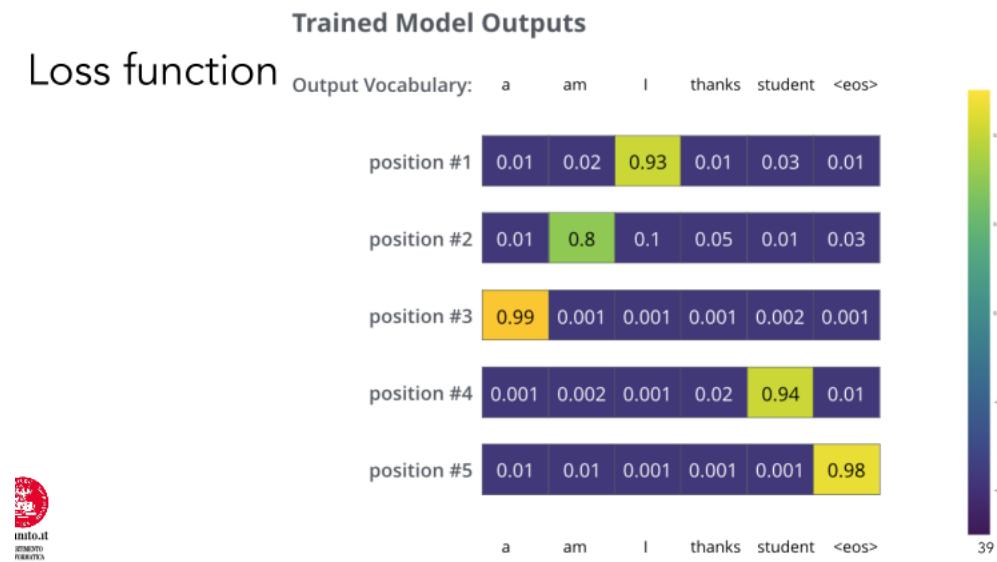


Figure 6.3: Loss Function.

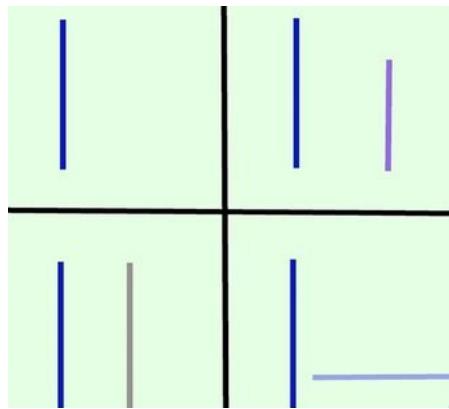


Figure 6.4: "Loss" Function.

6.2.1 Positional Embeddings

Definizione 6.2.2: Positional Embeddings

Nei Transformer, l'ordine delle parole non è implicito, perciò si aggiunge una rappresentazione della posizione di ciascuna parola.

- Ogni parola w_i ha un embedding standard w_i e una rappresentazione di posizione p_i .
- L'embedding finale in input è dato dalla somma: $x_i = w_i + p_i$.

A differenza di approcci precedenti che concatenavano la posizione come vettore separato, nei Transformer la posizione è integrata direttamente nel vettore parola.

- La funzione p_i è definita in modo deterministico (hard-coded), tipicamente tramite funzioni sinusoidali che variano in modo unico per ogni posizione.

Note:-

Le positional embeddings forniscono al modello informazioni sul contesto sequenziale, pur mantenendo l'architettura completamente parallela del Transformer.

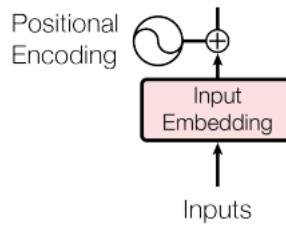


Figure 6.5: Positional Encoding.

Definizione 6.2.3: Positional Encoding

I positional encodings sono definiti in modo deterministico tramite funzioni seno e coseno di frequenze diverse. Per una posizione pos e una dimensione i :

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad \text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

- Le dimensioni pari del vettore usano la funzione seno.
- Le dimensioni dispari usano la funzione coseno.
- d_{model} è la dimensione del modello (es. 512).

Note:-

L'uso di funzioni sinusoidali permette al modello di apprendere relazioni tra posizioni tramite semplici operazioni lineari, e generalizzare a lunghezze di sequenza mai viste.

6.2.2 Sub-word Tokenization**Definizione 6.2.4: Sub-word Tokenization**

I Transformer operano su unità sub-word, ovvero frammenti di parole piuttosto che parole intere. Queste unità sono generate automaticamente mediante l'algoritmo di Byte Pair Encoding (BPE):

- Il vocabolario iniziale contiene tutti i singoli caratteri, inclusi simboli speciali come </w> per indicare la fine di una parola.
- L'algoritmo scansiona un corpus di testo e conta le coppie di simboli più frequenti.
- La coppia più frequente viene sostituita con la sua concatenazione e aggiunta al vocabolario come nuovo simbolo.
- Questo processo viene ripetuto iterativamente, costruendo un dizionario di unità sub-word che cattura sequenze frequenti.

La sub-word tokenization:

- Rendono la TN più *robusta* quando incontra parole sconosciute: utilizzando delle sottoparole conosciute.
- Salva spazio: si utilizza un vocabolario molto più piccolo.

6.2.3 Attention

Definizione 6.2.5: Multi-Head Attention

Il meccanismo di multi-head attention permette al modello di focalizzarsi su diverse parti della sequenza in parallelo. Ogni testa h_n utilizza tre proiezioni:

- Query (Q): vettore che rappresenta la parola corrente e cerca informazioni rilevanti nelle altre parole. Dimensione $d_q = 64$.
- Key (K): vettore associato a ogni parola, che consente di determinare quanto essa debba essere presa in considerazione. Dimensione $d_k = 64$.
- Value (V): vettore che contiene l'informazione effettiva che può essere “letta” se la parola è rilevante. Dimensione $d_v = 64$.

L'attenzione di ogni testa è calcolata come:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V$$

Dopo il calcolo parallelo di 8 teste indipendenti, i risultati vengono concatenati e proiettati in uno spazio di dimensione $d_{\text{model}} = 512$.

Note:-

La TNs espandono lo strato dell'attention ripetendolo più volte.

Definizione 6.2.6: Self-Attention

Il meccanismo di self-attention consente a ogni parola della sequenza di “osservare” tutte le altre parole e combinare le loro rappresentazioni in modo pesato.

- Per ogni parola w_i si generano tre vettori:
 - Query (q_i)
 - Key (k_i)
 - Value (v_i)
- I vettori q_i e k_j vengono usati per calcolare uno score di attenzione tra la parola w_i e ogni altra parola w_j .
- Lo score determina quanto w_i deve “prestare attenzione” a w_j .
- L'embedding finale z_i di w_i è una combinazione pesata dei vettori v_j delle altre parole, in base ai punteggi di attenzione.

L'intuizione può essere spiegata con un'analogia:

- La query è un tipo particolare di pane.
- Le chiavi sono il nome di tutti i prodotti della panetteria.
- Il valore di ogni prodotto è il prezzo.

Note:-

L'obiettivo è di porre più attenzione al prezzo (values) dei prodotti (keys) che sono più vicini ai nostri interessi (query).

6.2.4 Training

Definizione 6.2.7: Training

L'addestramento di un Transformer consiste nell'ottimizzare i suoi parametri:

- gli embedding di input dei token sub-word,
- le matrici di pesi W nei diversi layer (es. proiezioni di Q, K, V e FFN).

Il processo di training si articola in due fasi:

- *Pre-training (non supervisionato)*: il modello apprende rappresentazioni linguistiche generali da grandi quantità di testo, tipicamente usando compiti come il masked language modeling.
- *Fine-tuning (supervisionato)*: il modello viene specializzato per un compito specifico (es. classificazione, traduzione, Q&A) utilizzando un dataset etichettato.

Corollario 6.2.1 Pre-training (MLM)

Le procedure di pre-training permettono di catturare una serie di languages pattern che sono indipendenti dall'applicazione. I modelli di pre-training utilizzano un Masked Language Model (MLM):

- I tokens sono mascherati rimpiazzandoli con token speciali (e.g [MASK]) e la TN deve indovinare il token sotto la maschera.
- Solitamente un 15% dei tokens viene mascherato.



Figure 6.6: MLM or something idk.

MLM:

1. Il classificatore che predice la masked word funziona al di sopra del contextualized embedding prodotto dallo strato n .
2. Il pre-training è un algoritmo non supervisionato perché non richiede annotazioni da esperti del dominio.
3. I masked tokens possono apparire dovunque in un dato testo. Un contesto che può essere usato per indovinarli sia da destra che da sinistra della maschera è chiamato *bidirectional language model*¹.

¹Tradizionalmente gli LM procedono da sinistra a destra.

Corollario 6.2.2 Pre-training (NSP)

Un'altra procedura per il pre-training è il next sentence prediction (NSP). I transformers sono allenati a predire quali frasi seguono da una frase data in input.

NSP:

- Utilizzano Token e Positional embeddings.
- Utilizzano un nuovo tipo di embedding che codifica a quale segmento il current token appartiene.
- Viene introdotto il token [CLS] che viene inserito all'inizio dell'input:
 - È trattato come un normale token: partecipa alla self-attention con tutti gli altri token.
 - La sua rappresentazione contestualizzata (embedding finale) viene usata come input per un classificatore.
 - Grazie all'attenzione, accumula informazioni da tutta la sequenza.
- Genera esempi positivi dalla frase presa in esame ed esempi negativi da pezzi a caso del corpus.

Definizione 6.2.8: Fine-Tuning

Il fine-tuning consiste nell'adattare un Transformer pre-addestrato a un compito specifico di NLP:

- I testi di input sono preceduti da [CLS] e separati, se necessario, da [SEP].
- Il modello di base è già stato pre-addestrato (e.g. con MLM o NSP).
- Si prosegue l'addestramento con una funzione di loss specifica per il task, per esempio cross-entropy per la classificazione.

Problemi:

- L'attention algorithm ha complessità quadratica.
- Solitamente non è un problema perché si usa il parallelismo delle GPU, ma quando queste non sono disponibili le TNs sono molto più lenti di semplici reti neurali.
- I positional embeddings codificano posizioni assolute dei token, anche se nuove architetture implementano un sistema relativo.

