

# LINGUISTICA APPLICATA

---

## Lezione 1 – 26/02

### **Cosa e dove sono i significati?**

- Il **significato di un'espressione linguistica** è visto nell'ottica di ciò a cui si riferisce nel mondo (e.g. Semantica esternalista, Semantica referenziale, Semantica vero-condizionale, Semantica formale).
- Il **significato nella testa**: significato visto come qualcosa che accade nella nostra mente. Nel caso di una lingua che non conosciamo, la parola non suscita una rappresentazione.
- Un approccio internalista come questo tiene conto di processi mentali: associazione a concetti astratti.
- **Significati nel linguaggio**: Come le parole sono usate nella lingua determina il significato contenuto nelle stesse. Es. Dimensione connotativa:
  - Questo articolo è stato scritto da un gruppo di ricercatori
  - Questo articolo è stato scritto da una banda di ricercatoriBanda qui ha una connotazione negativa, poiché si riferisce a un gruppo, ma tipicamente è un termine associato ad un'organizzazione criminale.  
L'uso della parola in certi contesti condiziona l'aspetto del contenuto semantico determinandone il significato.

La semantica distribuzionale si occupa proprio di questo per capire come i chatbot conoscano il linguaggio. Attraverso la co-occorrenza distribuzionale si arriva a modelli come ChatGPT.

### **Cosa fa il significato? Cosa ci permette di fare?**

Conoscere il significato significa **avere la competenza semantica** da utilizzare per risolvere determinati compiti.

"**General Semantic**" di Lewis dice di partire dai fenomeni e poi arrivare al significato in base a dei test applicati a modelli.

Conoscere il significato delle parole permette di trovare inferenze.

Ci sono diversi tipi di inferenza:

- a. **Implicazione/competenza logica**:  
Tutte le volte che accetto la verità della prima frase accetto la verità anche della seconda frase.  
L'inferenza è un test fondamentale per capire gli elementi linguistici e se il soggetto o la macchina possiede quella conoscenza.
- b. **Presupposizione**: se accettiamo la verità di una frase dobbiamo accettarne il contenuto dell'assunzione.
- c. **Inferenza probabilistica**: sono credenze del mondo "Il neonato beve latte"
- d. **Implicative**: condivide con la c il fatto che, se dico "alcuni studenti hanno fallito un esame potrebbe anche essere che tutti lo hanno bocciato".  
È importante considerare il fatto che sono inferenze cancellabili: se aggiungo del contesto posso pensare a un'altra inferenza.
- e. **Contraddizione**: quando non posso aggiungere del contesto

**Inferire:** acquisire della nuova conoscenza non ancora acquisita, ci permettono di non dire tutto quello che può essere vero. La solidità delle inferenze deriva dalla plausibilità. "Vedo un gatto a metà e inferisco che abbia anche le zampe dietro"

### **Similarità**

Se due parole sono semanticamente simili o no.

Se non so il significato, ad esempio, di abdicare non saprò nemmeno il contesto nel quale si usa la parola.

Grazie alla similarità sono in grado di capire la similarità semantica in base alla **semantica combinatoria**.

L'equivalenza semantica significa che abbiamo capito il significato e la sua rappresentazione che per ChatGPT non è altro che un vettore di numeri.

## **Lezione 2 – 27/02**

### **Tema dell'inferenza**

Relazioni tra espressioni linguistiche.

Anche la **similarità** può farne parte sebbene non sia un'inferenza.

Un'altra capacità di fare inferenza è **attribuire significati a oggetti del mondo**.

"*Lo studente studia un libro in biblioteca*" e nel frattempo **vediamo l'immagine**.

Capiamo subito se la frase è vera oppure è falsa rispetto all'immagine avendo sottocchio entrambe le cose...

Fare **riferimento** significa poter far riferimento a specifiche parti della frase che corrispondono a specifiche parti dell'immagine.

Riesco a **identificare** lo studente nell'immagine, la biblioteca, il libro...

Da un punto di vista umano questo è un elemento fondamentale del linguaggio: comprendere il significato mi permette di far riferimento a entità del mondo.

Esiste anche un mondo di elementi astratti che noi comunque capiamo.

Per fare un altro parallelismo tra strumenti artificiali ed esseri umani possiamo ricordare che esistono modelli addestrati capaci di svolgere questo tipo di task: sono capaci di partire da un'immagine e generare una frase che descrive l'immagine oppure il contrario come nel caso di Midjourney.

### **Referential vs. Inferential competence**

La competenza **referenziale** e **inferenziale** è diversa.

La distinzione è stata attribuita da **Diego Marconi** in **Lexical Competence**.

I computer comprendono il significato delle espressioni linguistiche ma il significato è una realtà complessa, multiforme.

La **competenza inferenziale** è quella conoscenza che permette a noi di individuare quelle che sono le relazioni che lessemi hanno con altri lessemi, che le espressioni linguistiche hanno con altre espressioni linguistiche.

"*Gianni guida la macchina*" - "*Gianni guida il veicolo*" capisco che le frasi sono simili per via del rapporto di iperonimia tra i due termini.

Per capire meglio andiamo avanti.

**Aardvark:** si tratta di un **piccolo** animale **notturno africano**.

Ora nella mia memoria **inferenziale** c'è un collegamento tra il termine e l'animale so che è piccolo ed è africano ma attualmente non so come sia fatto.

Ora posso accettare frasi del tipo "*L'aardvark è marrone*" ma non so ancora come sia fatto.

Questa è una conoscenza inferenziale che mi permette di avere una conoscenza sul significato anche se non so ancora riconoscere e distinguere l'animale.

In poche parole, la **conoscenza inferenziale** avviene quando non ho mai visto l'oggetto in questione ma in base alle caratteristiche che ho appreso su quel determinato oggetto allora io posso fare determinati ragionamenti su di esso, e quindi fare delle **inferenze**.

Al contrario **la competenza referenziale** è quando vedo direttamente l'immagine dell'aardvark e imparo a capire come sia fatto e che cos'è, riesco a dargli una forma e collocarlo nel mondo.

In questo modo c'è un collegamento tra un oggetto del mondo e il mio **vocabolario**.

Non avrei mai saputo che è un animale notturno e che vive in Africa in questo modo.

In realtà vedendo l'immagine faccio nuovamente inferenze perché paragono l'animale con altre immagini di animali che ho nella mente.

La **conoscenza referenziale** avviene quando osservo direttamente l'oggetto in questione.

**Referenza e inferenza sono dissociate, posso avere una competenza referenziale senza quella inferenziale ed il contrario.**

Ci sono **aree del cervello** più legate al mondo inferenziale mentre altre legate al mondo della competenza referenziale.

### What does meaning do?

Prima di capire cos'è il linguaggio siamo tenuti a capire cosa si può fare col linguaggio.

La **co-referenza** è la capacità di una espressione linguistica di riferirsi ad altre entità menzionate nel testo. Prendo una entità che è stata menzionata precedentemente nel contesto linguistico e le attribuisco delle caratteristiche.

Il fenomeno della co-referenza mette insieme aspetti della **competenza referenziale** con quella **inferenziale**. Capire l'entità alla quale fa riferimento l'altra entità sta alla base della nostra capacità inferenziale.

Facciamo un esempio di **co-referenza**:

*Aristotele era il maestro di Alessandro il grande, il filosofo era nato a Stagira.*

In questo modo io capisco che il filosofo è riferito ad Aristotele.

Se al posto di filosofo avessi avuto pittore la cosa inizierebbe a suonare strana.

È fondamentale quindi la conoscenza del mondo da parte dell'individuo.

Vediamo adesso un esempio un esempio di **bridging**:

*L'autobus si è fermato, l'autista ha spento il motore.*

Il **fenomeno di bridging** consiste nel fatto che è l'autista dell'autobus e non un autista a caso che ha fatto spento il motore. Questo ponte si crea nella nostra mente in maniera naturale e ci fa capire quelle che sono le conoscenze di senso comune.

**Conoscenza semantica:** so che l'aardvark è un animale;

**Conoscenza fattuale:** so che l'aardvark vive in Africa;

**Il linguaggio figurativo**, cioè la capacità di usare espressioni linguistiche non in maniera letterale è un grande capacità fondamentale della **competenza umana**.

Vediamo qualche **esempio di Metonimia (da intendere come uso del linguaggio figurativo):**

- *Gigi legge Seneca* -> Gigi legge “**il libro di**” Seneca
- *Gigi beve la bottiglia* -> Gigi beve “**l'intero contenuto della**” bottiglia

Alcuni hanno definito questa figura retorica come la “**scorciatoia del pensiero**”.

Esistono anche le **Metafore (usate sempre a questo scopo):**

- *Il tempo vola* -> il tempo “passa velocemente”
- *L'avvocato è uno squalo* -> l'avvocato è “aggressivo”

Generalmente i termini della metafora sono contestualizzati mentre per quelli della metonimia c’è un oggetto al quale viene associato un evento.

Dobbiamo quindi capire quali tipi di meccanismi noi mettiamo in campo per far comprendere al computer questo tipo di espressioni.

## **Symbolic models**

Come possiamo **rappresentare il significato**.

Prima di poterlo rappresentare dobbiamo comprendere cosa fa il significato.

Ma come posso rappresentare il significato delle parole in modo tale che sia in grado di rappresentare queste parole?

Sullo studio del significato ci sono due grandi approcci:

### **1) Modello simbolico:**

Per rappresentare il significato di una espressione linguistica uso altri simboli che sono tratti da un altro linguaggio di tipo formale. Un linguaggio che serve per descrivere un altro linguaggio è chiamato metalinguaggio.

Uso quindi dei simboli non linguistici per rendere esplicito il significato dei simboli linguistici.

Questi simboli non linguistici hanno una struttura capace di spiegare le proprietà semantiche delle parole del linguaggio.

Spiego quindi perché si fa un’inferenza tramite l’associazione con dei simboli formali che rappresentano in maniera esplicita questo significato.

I modelli simbolici sono **stati dominanti fino agli ultimi decenni** e sono ortogonali tra **approccio internalista ed esternalista**.

Quello che caratterizza i modelli simbolici è che le rappresentazioni semantiche sono di tipo qualitativo e discreto. Questo lo vedremo poi meglio su R.

**Modello:** entità capace di spiegare un fenomeno.

Sostanzialmente qui l’informazione rappresentata è sempre variabile in parte del processo, per questo si definisce **qualitativa**.

### **2) Utilizzare strutture matematiche: rappresentare la semantica con vettori**

Ci sono dei modelli che permettono di partire da un insieme di dati (puramente linguistici, immagini...), estraggono informazioni da questi dati e poi li rappresentano sotto forma di **valori numerici**.

Questi valori numerici rappresentano proprietà semantiche degli elementi linguistici.

Qui abbiamo delle rappresentazioni **puramente quantitative** e non qualitative come avviene per i simboli. Queste rappresentazioni semantiche sono distribuite, nel senso che tutto viene trasformato

in un numero che appartiene all'interno della struttura vettoriale completa. Ovviamente a differenza del modello simbolico.

Il problema maggiore è: **come faccio a interpretare l'informazione presente nei vettori**, come faccio a sapere che l'informazione effettivamente è presente nei valori del vettore?

Si parla quindi di **BLACKOUT BOX**. Delle scatole nere che contengono l'informazione che catturano informazioni semantiche in modo non lineare e complesso. È spesso difficile interpretare cosa rappresentano esattamente queste dimensioni e come contribuiscono alla comprensione semantica complessiva della frase.

**Chat GPT** fa questo, impara dai testi e sfrutta i vettori per imparare e rispondere.

Funziona! Quindi significa che l'informazione arriva.

L'informazione diventa un vettore di dati, il cosiddetto **embedding**.

Una volta che l'informazione è diventata vettore non si capisce più quali sono i dati di partenza.

## **Produttività nel pensiero e nel linguaggio**

Noi possiamo dare al modello delle frasi che non ha mai sentito prima e può comunque risponderci.

In realtà non siamo sicuri al 100% che la frase non sia già stata studiata e osservata dal modello.

La capacità spettacolare umana è il fatto che noi abbiamo **la potenzialità di comprendere situazioni completamente nuove e di descrivere queste situazioni col linguaggio e con strutture mai usate prima**: “*il cane gioca a basket*” — è una struttura linguistica nuova.

## **Meaning, productivity and compositionality**

Quello che contraddistingue l'intelligenza umana è la sua **produttività**: innovativa, creativa capace di interpretare sempre frasi nuove.

**G.Frege (Padre della logica formale contemporanea e della filosofia del linguaggio)** dice:

*È meraviglioso ciò che il linguaggio realizza, con poche sillabe esprime pensieri complessi e gli altri che magari sentono per la prima volta quel discorso lo comprendono.*

*Questo non sarebbe stato possibile se noi non avessimo potuto distinguere le parti del pensiero che corrispondono a delle parti nella frase così che la costruzione della frase diventa una ricostruzione del pensiero stesso.*

Sostanzialmente i nostri pensieri non sono delle entità olistiche (interne, complete) ma sono delle cose composte di parti per le quali esistono delle corrispondenze precise con delle parti linguistiche.

## **Compositionality**

È la ricetta per comprendere il perché il linguaggio naturale è così produttivo.

**Il principio di composizionalità o composizionalità Freigiana (Fregean Compositionality):**

*Il significato di un'espressione è la funzione del significato delle sue parti e del modo in cui queste sono sintatticamente combinate.*

**Il principio di composizionalità** presuppone che ci sia un insieme finito di parti ovvero il nostro lessico, cioè le parti elementari immagazzinate con un significato e dei principi combinatori che permettono di combinare queste parti in modo tale da creare delle espressioni complesse (grammatica).

**Lessico + grammatica** = possibilità di creare delle rappresentazioni semantiche complesse.

## **Lezione 3 – 05/03**

**Il principio di composizionalità** cerca di definire il principio di composizionalità linguistica.

Questo è possibile perché il linguaggio, ed il pensiero funzionano nel seguente modo: ci sono delle parti del linguaggio che si combinano con quelle del pensiero e quindi tutto funziona.

### Functions and arguments

Oggi parleremo di **modelli simbolici** che determinano il linguaggio naturale attraverso il **linguaggio formale** cioè attraverso anche alcuni modelli matematici.

Questo paradigma inizia a nascere tra fine 800 e inizio 900 per poi arrivare ai Language models come Chat GPT.

È importante in questo tipo di approccio il rapporto con la matematica.

Tutti questi linguisti e filosofi del linguaggio vogliono interpretare il linguaggio naturale come un modello logico-matematico imperfetto.

Il linguaggio naturale, dunque, può essere modellato come un linguaggio formale.

Fondamentale parlare del linguaggio naturale come uno strumento matematico.

In questo campo il rapporto tra **funzione** e **argomento** è fondamentale.

Rappresentiamo il significato come la risultante della composizione di strutture che sono funzioni con argomenti.

Allo stesso modo di come componiamo funzioni con argomenti così noi costruiamo il linguaggio naturale.

C'è dunque un parallelismo tra la nozione linguistica di **predicato** e quella di **funzione**. Io posso rappresentare i predicati del linguaggio naturale come delle funzioni.

Tom *runs* → **PREDICATO**

**Scompongo** questa frase come **predicato** e **argomento** e il predicato è una funzione con variabile o variabili.

Le funzioni si rappresentano con la **notazione lambda**:

$$\lambda x[\text{run}(x)] \quad \left\{ \begin{array}{l} \text{run}() \rightarrow \text{FUNZIONE CHE' IL PREDICATO} \\ x \rightarrow \text{ARGOMENTO} \end{array} \right.$$

Questo serve per indicare che la variabile **x** è **una variabile libera** che richiede un argomento.

Qui siamo a livello della rappresentazione del significato, rappresentiamo quindi il modo in cui noi percepiamo il significato del predicato *run*.

Dunque, in questo modo definiamo che *run* esprime un processo e ha bisogno di determinati argomenti.

Il processo di costruzione del significato significa applicare funzioni ad un gruppo di argomenti.

Quanto questo sia fondamentale nel nostro modo di produrre il significato si vede da qui:

- **Jekendoff**: ogni teoria semantica Fregeiana riconosce che i significati delle parole possono contenere delle variabili, cioè degli slot che sono saturabili da degli argomenti.

- **Frege:** rapporto tra logica e matematica del linguaggio: come le equazioni o le espressioni matematiche possono essere divise in due parti, le frasi del linguaggio naturale funzionano come le espressioni matematiche. Una parte è **satura** l'altra è **da completare**: noi dividiamo "Cesare ha conquistato la Gallia":

Cesare è la parte **satura**, la seconda parte **non lo è** ed in questo caso è saturata da Cesare. Io do il termine **funzione per ciò che non è già saturo** quindi Cesare è l'argomento e la funzione è il resto.

Dire che c'è una separazione tra funzione e argomento significa dire che il significato di una espressione è indipendente dal significato dell'argomento, "Cesare" e "ha conquistato la Gallia" sono due parti diverse.

**Il concetto di rappresentare i dati come funzione significa che il significato del verbo è indipendente dal significato degli argomenti che ci metto dentro.**

**Proprio come avviene anche per una funzione matematica  $\log()$  fa sempre la stessa operazione indipendentemente dal suo argomento  $x$  ( $\log(x)$ ) che può essere qualsiasi numero tranne lo 0.**

"*Il cavallo corre*", corre può ospitare tanti altri argomenti diversi da cavallo.

Se non ci fosse l'indipendenza tra i due elementi non potrei mai che il significato di "*il cavallo corre*" è un qualcosa di simile alla comprensione della stessa frase sostituendo il "cavallo". Anche il puledro corre, l'asino, il bimbo corre...

La sistematicità si riferisce alla proprietà per cui la capacità cognitiva di un sistema è strutturata in modo tale che la comprensione e la produzione di certe espressioni linguistiche (o pensieri) sono intrinsecamente legate alla comprensione e alla produzione di altre espressioni linguistiche (o pensieri) correlate. In altre parole, se un sistema cognitivo può comprendere o produrre una certa combinazione di concetti, allora può anche comprendere o produrre altre combinazioni logicamente correlate.

Questo principio è il principio che due filosofi chiamano **il principio della sistematicità**. (Fodor e Pylyshin)

Questi due dicono nel 88' che le reti neurali non possono spiegare il meccanismo della cognizione.

*Read the book, the letter, the paper...*

*Kick the bucket (die) != kick the ball, empty the bucket*

Il concetto di funzione è il processo razionale di spiegare come avere un elemento principale che si applica a diversi tipi di argomenti.

Tipo funzione  $\log_2(x)$  fa la stessa cosa sempre qualsiasi sia l'argomento di  $x$ .

**Vediamo ora una definizione più formale di cos'è una funzione:**

Una funzione è un **mapping** dal punto di vista matematico ovvero un'associazione tra due elementi, cioè il **dominio** e il **codominio**.

*Per ogni elemento del dominio viene assegnato uno e uno solo elemento del codominio.*

$\lambda [f(x)] : A \rightarrow B$

Il processo di applicazione di una funzione al suo argomento è chiamato "**associazione funzionale**":

$a \in A, \lambda x [f(x)] | (\circ)$

Applico ora la funzione all'argomento a.

**PRED**      **ARG**      **RISULTATO**  
 $\lambda x [run(x)] (Tom) \Rightarrow run(Tom)$

Significa che applico la funzione  $run(x)$  a Tom cioè all'argomento Tom ottenendo  $run(Tom)$

**Ma cosa fa tutto questo?**

Io ho semplicemente manipolato dei simboli, dei componenti ed ho ottenuto in fine un altro simbolo.  
Ecco perché **modello simbolico**: manipolo simboli.  
Queste espressioni e simboli stanno per dei significati, ma che cosa sono questi significati?

### Meaning and truth conditions

Torniamo a modelli **internalisti** ed **esternalisti**.

**Esternalista**: ciò a cui ci riferiamo nel mondo con le nostre espressioni linguistiche.

Qui ha la meglio appunto la prospettiva di tipo esternalista.

Questa prospettiva esternalista ha visto come fondamentale la nozione di verità.

Il **concetto di verità** lega prototipicamente il linguaggio al mondo.

**David Lewis** dice chiaramente che il significato di una frase determina le condizioni in base alle quali una frase è vera oppure è falsa.

Non si può fare semantica senza chiedersi il rapporto tra il linguaggio e mondo.

*“Conoscere il significato di una frase significa conoscere le sue condizioni di verità”.*

### La semantica verso condizionale:

Comprendere il significato di una frase significa comprendere le condizioni che rendono una frase vera o falsa.

Il punto di vista fondamentale in questo caso è il punto di vista dell'enunciato del quale dobbiamo conoscere le verità.

Il secondo aspetto fondamentale è che la semantica non può che non porsi le condizioni di verità.

Chat GPT, per esempio, può dire delle cose corrette ma semanticamente false.

Posso fidarmi da un sistema che non distingue una cosa vera da una cosa falsa?

### Alfred Tarski:

Un capitolo fondamentale sulla **semantica e verità** nasce grazie a questo personaggio.

Nell'articolo *“Il concetto di verità nei linguaggi formalizzati”* (1956)

Tarski non voleva occuparsi di linguaggi naturali che per lui sono imperfetti e pieni di paradossi.

Lui voleva definire il concetto di verità nei linguaggi formali.

Lo schema di Tarski definisce cosa per lui è il concetto di verità:

*“La frase s è vera se e solo se p”*

**P** è l'espressione di un metalinguaggio formale che ha il compito di rendere esplicite le condizioni che rendono vere la frase s.

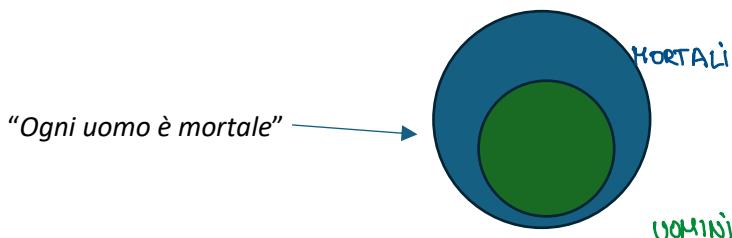
Per esempio: "Due è un numero naturale" vera se e solo se  $2 \in N$ .

Tarski crea una serie di regole per creare e definire queste affermazioni in maniera compositiva, per costruire schemi a partire dalle loro parti.

Il passaggio da linguaggi formali a naturali avviene grazie a un filosofo americano che è **Donald Davidson**:  
*"Per descrivere la semantica del linguaggio naturale cioè il modo in cui noi costruiamo il linguaggio naturale, questa deve essere fatta così come Tarski ha fatto per i linguaggi formali"*.

Per il linguaggio naturale **dovrà avere una teoria** che spiega perché una frase come "Gianni corre" sia vera o falsa, per fare questo servono delle condizioni di verità. In questo caso non so se effettivamente Gianni davvero corre.

**Devo ogni volta tradurre la mia frase in un set di Insiemi:**



**Non basta però solo tradurre ogni frase in un insieme.**

Questa operazione deve essere anche **compositiva**, cioè, ci deve essere una teoria che mi spieghi come sistematicamente un'espressione complessa dipenda dal significato delle singole parti.

*"Una teoria semantica del linguaggio naturale non può essere considerata adeguata a meno che non fornisca una spiegazione del concetto di verità per quel linguaggio lungo le linee generali proposte da Tarski per i linguaggi formalizzati [...] Per teoria della verità intendo un insieme di assiomi che comportino, per ogni frase nella lingua, una dichiarazione delle condizioni in cui essa è vera" ("Semantics for Natural Languages" 1970: 55).*

Cioè, il significato finale è la somma di piccoli significati dai quali poi determino le condizioni di verità.

Il linguaggio naturale può essere visto come un sistema logico, ma quanto è vera questa cosa?

Per Davidson questo era possibile.

*"In ogni caso, il tentativo è istruttivo, poiché nella misura in cui riusciamo a fornire una tale teoria per una lingua naturale, vediamo la lingua naturale come un sistema formale; e nella misura in cui facciamo della costruzione di una tale teoria il nostro obiettivo, possiamo pensare ai linguisti e ai filosofi analitici come collaboratori" (Davidson 1984, ibid.)*

### Sense, reference, and truth conditions

**Ogni espressione linguistica è composta da due parti:**

**L'intensione** (o il senso) e **l'estensione** (cioè il referente).

L'estensione di un'espressione (rappresentata con []) è l'entità del mondo a cui l'entità linguistica si riferisce, è una **convenzione** della semantica formale di rappresentare l'estensione con due parentesi quadre.

e.g. [Aristotele] = l'individuo Aristotele

L'intensione è quell'insieme di proprietà che ci permettono di determinare l'estensione di ogni elemento linguistico in ogni circostanza.

Molto simile alla nozione di concetto. Ma attenzione **il concetto è una entità esclusivamente psicologica** mentre nella visione di questi personaggi Tarski, Donald... tutto è **formale** e non psicologico.

Le espressioni linguistiche hanno una intensione di cui l'estensione può cambiare.

Possono esistere anche espressione con intensione diversa ma con estensione uguale: [[il maestro di Alessandro il grande]] = [[l'autore dell'Organ]]

Ci possono essere frasi che non hanno intensione.

**Il valore di verità è equivalente a ciò a cui la frase si riferisce.**

“*Il docente di LA in questo momento è in piedi*” questa frase si riferisce a un individuo che quest’anno è A.Lenci e ha il valore di verità legato al fatto che è in piedi.

Se però il prossimo anno c’è Mario Rossi ed è seduto la frase diventa falsa.

**Dobbiamo quindi pensare che la nozione di valore di verità sono qualche cosa nel mondo a cui le frasi stesse si riferiscono.**

Frege: “*La frase Odisseo approdò a Itaca mentre dormiva*” ha un senso ma siccome il nome Odisseo non si riferisce a un individuo effettivo e quindi immaginario nel mondo attuale è anche dubbio che quella frase si riferisca a qualche cosa tuttavia è certo che non di meno chiunque seriamente pensasse che questa frase potesse essere vera e di conseguenza deve riferirsi a qualcosa di immaginario.

Cioè, io faccio un riferimento immaginario nella mia testa.

Il referente deve essere cerato ogni volta che è necessario.

Se io non immagino che Odisseo fosse esistito non posso nemmeno farmi la domanda se la frase è vera o falsa, dunque, la verità della frase è rappresentata dal suo referente.

**La verità della frase dipende dalla circostanza in cui questa frase è rappresentata.**

I valori di verità sono le entità a cui si riferiscono le frasi nel mondo.

### Truth-conditions semantics

Le condizioni di verità della **semantica vero-condizionale**:

- 1) Il **significato** di una frase è determinato dalle sue **condizioni di verità**;
- 2) Le **espressioni del linguaggio naturale** sono tradotte in **un linguaggio formale** che rende esplicite le sue verità, questo linguaggio formale è **la logica dei predicati e non la teoria degli insiemi**.
- 3) Questa traduzione avviene **in maniera compositonale**, i termini lessicali sono espressioni di questo linguaggio metaformale es: *tom runs* visto prima, e questi elementi di linguaggio logico sono processate attraverso le espressioni **logico-formali**.
- 4) Non possiamo rendere le traduzioni **da formale a meta formale** in maniera olistica ma **compositonale**. Ho bisogno di un algoritmo che costruisce le forme logiche delle frasi complesse a partire dai pezzi di partenza.
- 5) Il **linguaggio formale deve essere un linguaggio disambiguato** per essere interpretato univocamente **nel modello** che è una sorta di rappresentazione logica del mondo che è il banco su cui determiniamo se le frasi sono vere o false

## Lezione 4 – 11/03

La **semantica vero-condizionale** afferma che comprendere il significato di una frase significa comprendere e rappresentare le sue condizioni di verità.

Tradurre l'espressione in un metalinguaggio che rappresenta queste condizioni di verità.

Per vedere se una frase è vera o falsa bisogna analizzarla nel mondo reale.

Nozione di **modello (model-teoretica)** introdotta da Tarski.

### Models

Un modello è una forma semplificata e strutturata del mondo.

La nozione di modello è basata su termini puramente insiemistici.

Da un punto di vista formale un modello è una **struttura formata da due componenti**:

- Un **dominio D** che è l'insieme di individui nel mondo. Per individui del mondo intendiamo l'insieme di oggetti individuali di cui noi assumiamo l'esistenza, anche se tale nozione è arbitraria.  
Per esempio, un dominio di persone:  $D = \{\text{'Al pacino'}, \text{'Mattarella'}, \text{'Lenci'} \dots\}$   
Gli elementi del dominio sono chiamati oggetti ma possono essere anche degli eventi.
- **La seconda componente I è la funzione di interpretazione** che lega delle entità ad altre entità.  
Tale funzione assegna alle espressioni di un linguaggio formale la loro interpretazione.  
Per esempio, assegna a "Mattarella" la sua interpretazione cioè "Individuo". Questa funzione associa l'interpretazione a predicati ed intere frasi anche.  
Ci possono essere delle interpretazioni estensionali e intensionali.  
**Estensione** = "presidente della repubblica"  
**Intensione** = "Quello che rappresenta il ruolo di presidente della repubblica" (la semantica intensionale cerca di determinare le caratteristiche linguistiche andando a definire le proprietà di estensione in ogni mondo possibile).

### Montague Semantics

Di semantiche **model-teoretics** ci sono molte rappresentazioni.

Ora vedremo la più importante ovvero **la semantica di Montague**.

Questo rappresenta il modello più prototipico: prende Frege e Tarski e li applica a un modello di linguaggio naturale, con lo scopo di creare una semantica che si basa su verità e creare un linguaggio come se fosse formale.

L'idea è la seguente:

Per rappresentare la semantica di una espressione linguistica bisogna comprendere le sue condizioni di verità.

**Si opera una traduzione delle frasi naturali in espressioni di un linguaggio formale.**

La differenza fondamentale tra il naturale e il formale è che il primo è ambiguo il secondo no.

Le espressioni del linguaggio formale rendono esplicite le condizioni di verità delle frasi del linguaggio naturale. In questo modo, passiamo da un linguaggio ambiguo a uno non ambiguo e parliamo di forme logiche o logica formale.

La forma logica è un linguaggio formale che rende esplicita la struttura logica.

Queste forme logiche sono interpretate sul modello.

***LN → LF → interpretato sul modello (cioè sul mondo esterno)***

Si tratta di un approccio **esternalista, simbolico** e ci danno la **competenza referenziale**.

In questa maniera rendiamo esplicite la forma logica del linguaggio e le inferenze, cioè i ragionamenti che noi facciamo sul linguaggio stesso.

Montague usa una **variante della logica dei predicati** come calcolo logico.

Estende la logica del primo ordine.

L'idea alla base di queste operazioni è che si parte dalle frasi composte da parole.

Tale frase del LN viene tradotta in formula che esprime le condizioni di verità.

I componenti di base del linguaggio logico sono:

- **Simboli** che sono chiamati **costanti individuali** e rappresentano individui specifici es un simbolo per **Tom**, un simbolo per **Alessandro** ecc... i simboli possono essere alfanumerici
- **Variabili:** anche queste hanno come valori degli individui es  $x = \text{qualsiasi tipo di individuo}$  ecc.. sono chiamati anche termini
- Ci sono poi **i simboli predicativi** es  $\lambda x [x \in M]$
- **Connettivi logici:** And e Or ( $\wedge, \vee$ )
- **Quantificatori:**  $\exists, \forall$  esistenziale e per ogni
- **Formule**  $\text{run}(Tom)$

Montague aggiunge a questo linguaggio logico dei tipi.

**Tipo semantico** tau ( $\tau$ ).

Un tipo semantico permette di individuare chiaramente la denotazione di una determinata espressione linguistica.

Su quale pezzo del dominio viene interpretata l'espressione. (Dr)

Determino i tipi di entità ai quali fanno riferimento una espressione linguistica.

La nozione di tipo semantico viene introdotta da **Russel**.

Alla base di tutto questo precedentemente c'era la nozione di insieme.

**Russel invece dice:** anche la nozione di insieme può creare dei paradossi.

Funziona così: R è l'insieme che ha come elementi gli insiemi che non hanno al suo interno se stessi come elementi. Tipo il barbiere che rade tutti quelli che non radono sé stessi ma allora il barbiere rade se stesso o no?... è paradossale.

Questo è un problema perché, se ho un insieme ben formato devo dire se un elemento può appartenere o meno a sé stesso.

$$R = \{x | x \notin x\}, \text{ allora } R \in R \Leftrightarrow R \notin R$$

Se supponiamo che x appartiene a se stesso è sbagliato perché non può appartenere a se stesso data la premessa: R appartiene a R se e solo se R non appartiene a R. In questo modo Russel dice a Frege che il suo sistema non è perfetto.

La nozione di **appartenenza si applica a entità di diverso livello**.

Un individuo non è la stessa cosa di un insieme di individui.

Un insieme come quello sopra indicato è mal formato.

La nozione di appartenenza va applicata sempre a insiemi di tipo diverso.

Se i tipi vengono violati e non seguono questo principio allora diventa mal formata.

Montague a questo punto definisce in maniera ricorsiva i principi semantici.

- (espressione semantica) denota individui  $D_e = D$
- è il valore di verità  $D_t = \{0,1\}$

Questi due tipi sono saturi, non necessitano di un completamento, sulla base di questi tipi viene definita la regola che permette di creare infinite gerarchie di tipi funzionali:

$$f: D_a \rightarrow D_b \quad | \quad \begin{array}{l} \text{funzione i cui argomenti sono di tipo a} \\ \text{e i cui valori sono di tipo b: } D_{\langle a, b \rangle} \end{array}$$

Questa, quindi, è una **funzione ricorsiva**.

E e t sono i tipi quindi costruisco altri infiniti elementi come  $\langle e, t \rangle$  ma anche  $\langle \langle e, t \rangle, t \rangle$ ,  $\langle e, \langle e, t \rangle \rangle$  e così via.

Syntactic category	Example	Type	Logical form	Interpretation
Proper noun	<i>Tom</i>	$e$	Tom	individual
Sentence	<i>Tom runs</i>	$t$	run(Tom)	truth value
Common noun	<i>dog</i>	$\langle e, t \rangle$	$\lambda x[\text{dog}(x)]$	set
Intransitive verb	<i>run</i>	$\langle e, t \rangle$	$\lambda x[\text{run}(x)]$	set
Transitive verb	<i>chase</i>	$\langle e, \langle e, t \rangle \rangle$	$\lambda y \lambda x[\text{chase}(x, y)]$	binary relation

I tipi funzionali sono applicati ai **verbi intransitivi** e i **nomi comuni** quindi parole come cane sono tradotti in espressioni funzionali. Vedi per esempio *dog*.

L'interpretazione di questi due casi **non sono funzioni ma insiemi (set)** la differenza fondamentale è la seguente: una funzione di tipo  $\langle e, t \rangle$  è equivalente all'insieme di individui tali per cui quella funzione ci da valore 1.

$$f: D \rightarrow \{0,1\} \quad | \quad \text{tutti gli elementi con valore 1 compongono l'insieme.}$$

Vediamo con cane:

$$\lambda x[\text{dog}(x)] \rightarrow \text{e ottengo l'insieme dei cani ovvero gli elementi con valore 1.}$$

Questa è la caratteristica fondamentale di un insieme. Tale funzione la posso vedere equivalente all'insieme di tutti i cani poiché **denota tutti quelli che sono cani con 1 e con 0 tutto il resto**.

Ecco perché queste espressioni denotano un insieme.

Per i **transitivi** devo prendere in input due individui.

Tipo **inseguire** prende due argomenti chi insegue e chi viene inseguito.

Anche qui ho una notazione insiemistica.

$$[\lambda y \lambda x [\text{chase}(x, y)]]$$

Avrò un insieme di coppie di individui dove il primo insegue il secondo.

Gianni ha dato i promessi sposi a Maria.

Dare è **trivale**

Come rappresentiamo questa cosa? Quale è il tipo semantico di dare?

$$\langle e, \langle e, \langle e, t \rangle \rangle \rangle$$

$$[\lambda z \lambda y \lambda x [\text{due}(x, y, z)]]$$

Montague dice che tutto questo che viene mostrato nella tabella è il lessico.

## Tipo di applicazione funzionale

Non si può dire che una funzione si applica a un argomento e basta ma bisogna essere sicuri che tale funzione si applica alla funzione.

if  $\alpha$  is of type  $a$  and  $\beta$  is of type  $\langle a, b \rangle \Rightarrow \beta(\alpha)$  is of type  $b$

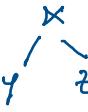
Le funzioni vanno applicate a tipi giusti.

**Il principio di funzionalità pone un rapporto tra strutture sintattiche e semantiche.**

In Montague c'è un principio di isomorfismo tra sintassi e semantica, ad ogni operazione di sintassi c'è un approccio semantico.

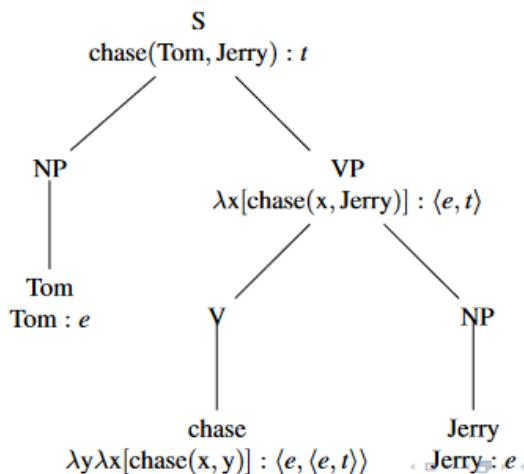
Ad ogni regola che costruisce l'albero sintattico ci deve essere una interpretazione semantica.

In questo modo funziona la type function application.

Let  $[x:y:z]$  be a syntactic node  $\Rightarrow$    
 $\alpha$  the logical form of  $y$   
 $\beta$  the logical form of  $z$

- if  $\alpha: a \wedge \beta: \langle a, b \rangle \Rightarrow$  logical form of node  $x$  is  $\beta(\alpha): b$
- if  $\alpha: \langle a, b \rangle \wedge \beta: a \Rightarrow$  logical form of node  $x$  is  $\alpha(\beta): b$

*Tom chases Jerry*



I tipi semantici di Montague determinano quali sono i tipi possibili degli argomenti di una funzione.

Le funzioni predicative, per esempio, si applicano solo a individui, vedremo poi come altre funzioni si applicano a insiemi di individui.

Restrizioni di selezione: predici del LN hanno vincoli semanticici sui tipi di predici ai quali sono applicabili.

I tipi individuali vengono separati in diverse categorie (animati, inanimati e così via .. )

Le restrizioni di selezione fanno sì che per esempio *drink* possa avere solo certi argomenti.

(2)  $\lambda y: \text{LIQUID } \lambda x: \text{ANIMATE} [\text{drink}(x, y)]$

- (1) a. The tall man drinks beer.  
     b. \* The red idea drinks beer.

Lezione 5 - 12/03

Una delle assunzioni di questo linguaggio è quello che esista una totale similarità tra linguaggio formale e logico. Stiamo parlando di Montague.

Questo ha avuto molto **impatto su IA**: se il linguaggio è trattabile come modello logico allora sia l'intelligenza umana che l'artificiale si basano su schemi logici.

Il linguaggio ha la caratteristica di essere un linguaggio formale mascherato.

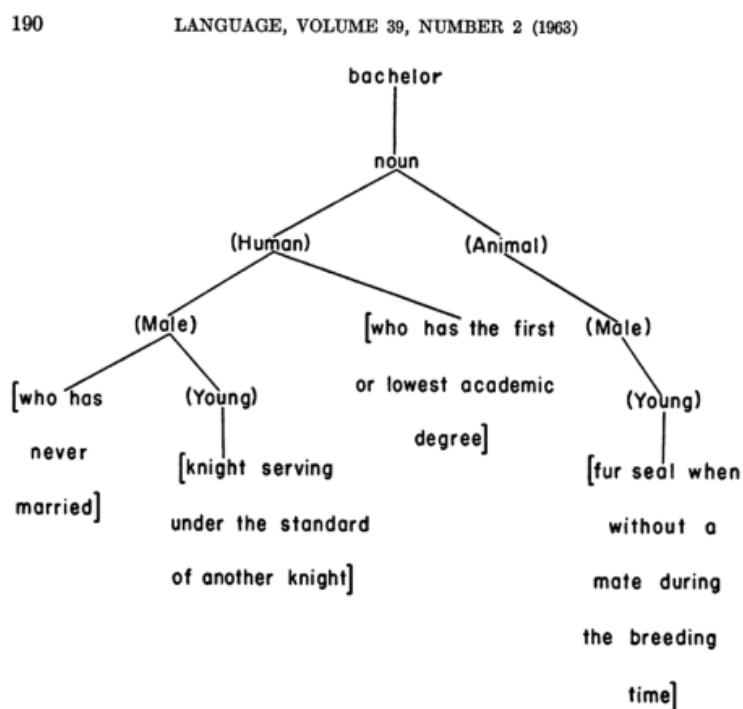
## Natural Language and logical form

**Quine:** i matematici lavorano allontanandosi dal linguaggio ordinario ovvero qualcosa che nasconde la realtà logica del mondo matematico. Questo allontanamento può però gettare luce sul linguaggio ordinario dal quale ci si allontana. Quindi bisogna esplorare le strutture soggiacenti.

**Non esiste una differenza teorica importante tra linguaggio naturale e formale quindi, allo stesso modo dei linguaggi formali va studiato anche il linguaggio naturale secondo Quine.**

**Montague** non pensa che abbiano successo trattamenti formali del linguaggio naturale da ceppi linguistici formali. Anche Chomsky tenta un modello formale del linguaggio. Montague considera la creazione di una teoria della verità per lo studio del linguaggio sia sintattico che semantico.

La polemica di Montague, quindi, è che Chomsky e la sua grammatica generativa tenta una ricostruzione del linguaggio ma allontanandosi dal linguaggio formale.



**Katz e Fodor** propongono una teoria internalista che fa a meno delle condizioni di verità e inaugurano un approccio che si basa sul partire dal significato dei termini lessicali che vengono scomposti in primitive lessicali. **Quindi praticamente si parte da una parola e si fa un albero con tutti i possibili significati.**

Tutto questo si evince dall'albero soprastante.

Le foglie sono dei residui sulla conoscenza del mondo in relazione a quel determinato argomento.

Per Montague l'albero si sarebbe rappresentato così:  $\lambda x [ \text{bachelor}(x) ]$

Katz e Fodor partono dall'analisi semantica che sidecomponen in concetti primitivi per vedere le relazioni tra i diversi significati.

Katz e Fodor, quindi, dicono **di partire dagli elementi lessicali per poi creare l'interpretazione semantica** dell'intera frase che è comunque l'unione di questi elementi primi.

Una teoria semantica interpreta le teorie sintattiche e le descrizioni grammaticali che il linguaggio rivela. Simile a Montague perché tutto si costruisce componendo ma diverso perché spariscono i concetti di verità, le marche semantiche qui vengono combinate tra loro per avere il significato completo della frase stessa. Questo modello non ha mai avuto successo.

**Semantica interpretativa:** una teoria semantica interpreta le strutture sintattiche.

**Generare** significa costruire delle strutture complesse.

**Interpretare** significa dare un senso a delle strutture che sono già state costruite.

Immaginiamo dei simboli fintizi:

A questo punto do un'interpretazione ai simboli, cioè, do un significato ai simboli.



È la sintassi che mi dice come si uniscono i simboli mentre la semantica li interpreta.

La sintassi, quindi, genera e costruisce strutture complesse mentre la semantica fornisce interpretazione ai simboli di partenza e alle strutture in generale.

Questo è ciò che Montague e Semantics at MIT hanno in comune ovvero la **composizionalità**.

Ci sono dei casi in cui questo allineamento tra sintassi e semantica non sembra esistere.

Quando questo accade ci sono due posizioni:

- In realtà esiste un allineamento ma va trovato e si fa complicando la semantica.

Prendiamo *Tom chased Jerry*:

**FORMA LOGICA di Tom chase Jerry :**

*Tom chased Jerry*  $\mapsto$  *chased (Tom, Jerry)* : 1

Il problema si pone quando noi prendiamo delle frasi con dei quantificatori universali o esistenziali come nei seguenti casi:

- (1) *A cat chased Jerry*
- (2) *Every cat chased Jerry*

Qui complico la sintassi per comprendere il rapporto con la semantica.

Ora invece faccio il contrario, complico la semantica per vedere se posso avere strutture semantiche di tipo diverso.

$\exists x [ \text{cat}(x) \wedge \text{chores}(x, \text{Jenny}) ] : r$

$\forall x [ \text{cat}(x) \rightarrow \text{chores}(x, \text{Jenny}) ] : r$

Le frasi così rappresentate hanno la stessa struttura sintattica ma vengono rappresentate in diverse forme logiche.

a.  $[s [DP_{NP} \text{Tom}][VP [V \text{chores} [DP_{NP} \text{Jenny}]]]]$

b.  $[s [DP_{NP} [\text{Det} A [NP [N \text{cat}]]] [VP [V \text{chores} [DP_{NP} \text{Jenny}]]]]$

c.  $[s [DP_{NP} [\text{Det} \text{Every} [NP [N \text{cat}]]] [VP [V \text{chores} [DP_{NP} \text{Jenny}]]]]$

Il ragionamento della grammatica generativa si basa sul complicare la sintassi.

Esiste un ulteriore livello di sintassi che noi non pronunciamo cioè non si inserisce nella fonetica e nella fonologia ma si mescola con la semantica creando un nuovo livello chiamato **deep structure o self structure** in cui i due elementi (quello legato alla pronuncia e quello legato al pensiero) sono diversi, qui la sintassi si divarica.

Montague non complica la sintassi in questo senso ma vede la sintassi come un livello visibile del linguaggio ponendo la differenza interpretativa sulla semantica.

Sebbene la sintassi sia ben visibile nella frase esistono tipi semanticci che hanno un proprio significato ben specifico.

Montague è n grado di fornire un modello da una grammatica uniforme.

Cambia la forma logica assegnata dei quantificatori: quale è il significato da associare a questi quantificatori?

**Una relazione tra predicati è la risposta.**

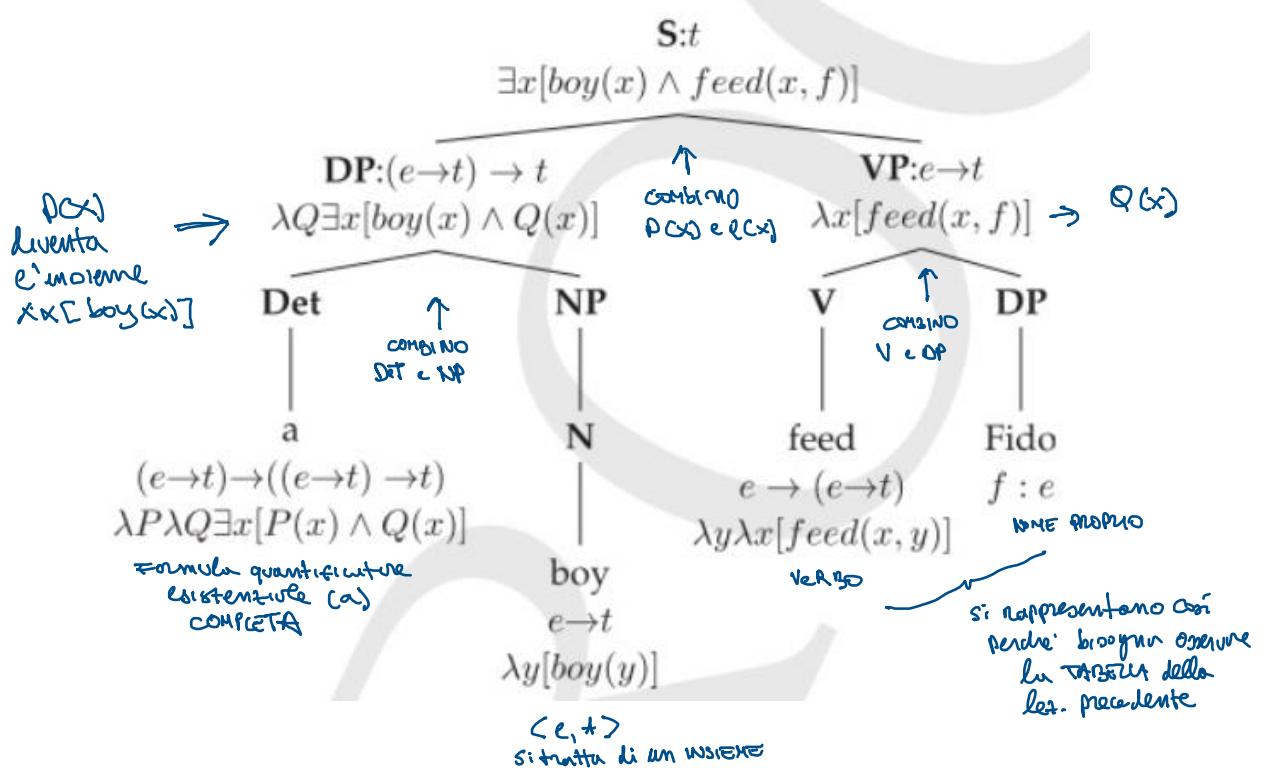
I quantificatori esprimono relazioni tra funzioni di individui, tra predicati di individui.

Un **quantificatore** avrà la forma seguente:

$\exists x [P(x) \wedge Q(x)] : \langle \langle e, + \rangle, \langle e, + \rangle, + \rangle \quad (\text{esistenziale})$

$\forall x [P(x) \rightarrow Q(x)] : \langle \langle e, + \rangle, \langle e, + \rangle, + \rangle \quad (\text{universale})$

Cioè, è la logica non applicata semplicemente a singoli elementi ma a insiemi di elementi.



Si parte sempre dal basso per raggiungere la formula finale che sta in cima.

Al posto di P(X) e Q(X) metto boy(X) e feed(x,f)

Concludiamo ora la semantica model-teoretica e andiamo verso un approccio che si allontana da Montague per degli aspetti fondamentali.

Si tratta di un modello di semantica sviluppato da **Jekendoff** e si chiama **conceptual semantics** che prende in considerazione gli aspetti concettuali e mentali.

Jekendoff si allontana anche da Montague e da Chomsky per quanto riguarda il tipo di rapporto tra sintassi e semantica ovvero che la sintassi genera e la semantica interpreta.

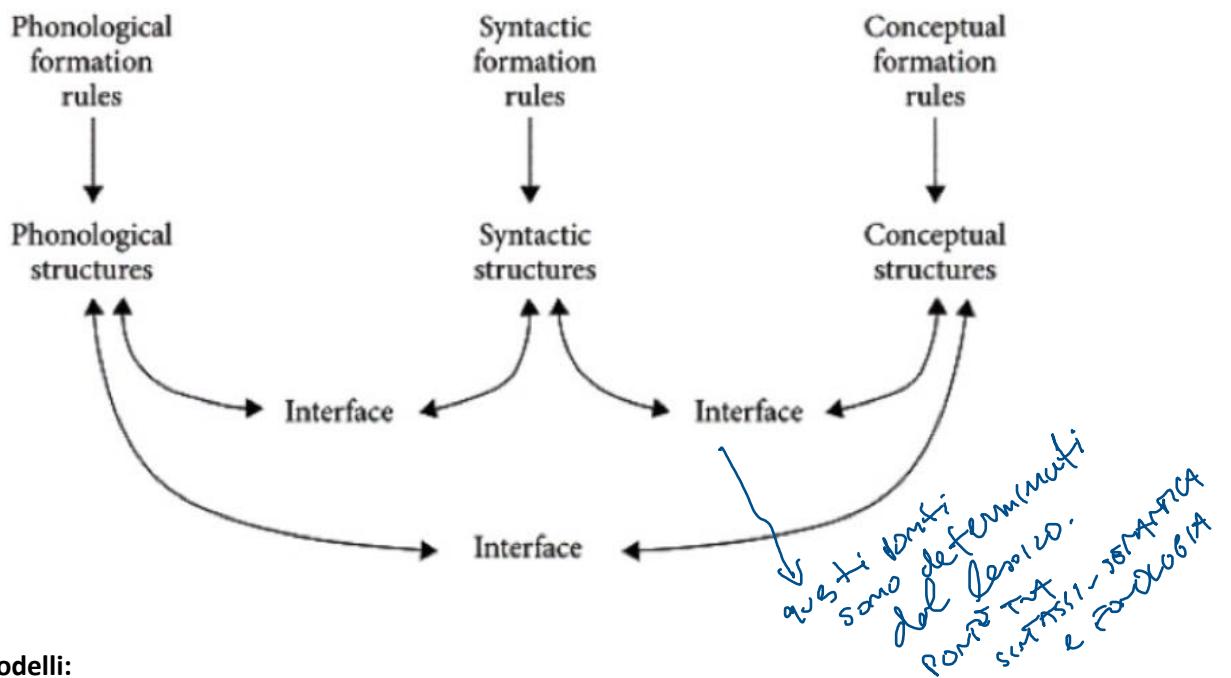
La prima cosa fondamentale è che siamo di fronte a un **modello internalista** ovvero una rappresentazione concettuale di un evento che ci dà la frase.

La conceptual semantics si occupa della forma delle rappresentazioni mentali interne, prospettiva internalista e mentale.

Il modello di Jekendoff è di tipo **simbolico**, assume che ci sono vari livelli semanticci ma c'è un insieme degli aspetti del significato che secondo lui sono grammaticalmente rilevanti e codificati in strutture di simboli detti **conceptual structure**.

La conceptual semantics **deriva dalla grammatica generativa** in particolare **dalla generative semantics** cioè un particolare modo di interpretare la grammatica tra la fine degli anni 60 e inizio 70.

Il ruolo della semantica è diverso: per Chomsky la semantica è interpretativa e basta per Jekendorff no, le strutture profonde del linguaggio sono di tipo semantico ovvero la semantica ha una dimensione generativa.



Ci sono 3 modelli:

- Fonologico
- Sintattico
- Semantico

Sono dei livelli che costruiscono autonomamente delle strutture rispettivamente fonologiche sintattiche e concettuali.

Il linguaggio fa sì che queste strutture si parlino l'uno con l'altro.

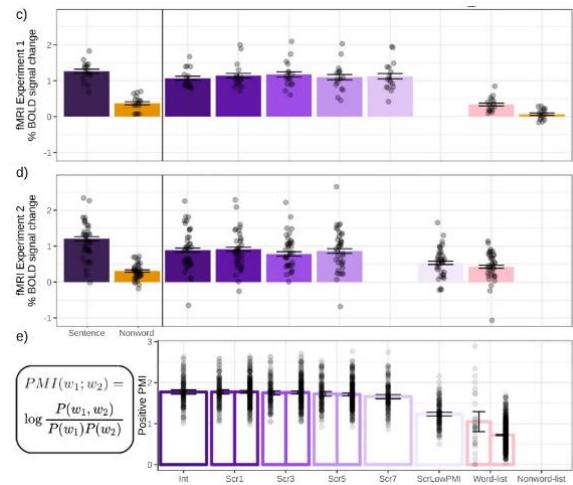
Parallelismo non significa allineamento ma la creazione di strutture autonome che si possono parlare e mettere in contatto.

**Principio di autonomia della sintassi:** nel senso che la sintassi è indipendente dalla semantica.

Jekendorff dice che la sintassi è indipendente dalla semantica ma anche viceversa ecco perché sono autonome.

## Ill-formed but meaningful

a	Int on their last day they were overwhelmed by farewell messages and gifts
	Scr1 on their last day they were overwhelmed by farewell messages and gifts on their last day they were overwhelmed by farewell and messages gifts
	Expt 1 Scr3 on their last day they were overwhelmed by farewell messages and gifts on their last they day were overwhelmed farewell messages by and gifts
	Scr5 on their last day they were overwhelmed by farewell messages and gifts on their last day were overwhelmed they farewell messages by gifts and
	Scr7 on their last day they were overwhelmed by farewell messages and gifts their last on they overwhelmed were day farewell by messages and gifts
b	Expt 2 LowPMI last they farewell gifts on were and their by day overwhelmed messages



Cosa succede se facciamo leggere a dei tizi delle frasi con violazioni sintattiche?

Sebbene le frasi contengano inflazioni grammaticali contengono elementi con una coerenza e quindi è possibile creare frasi graticoli.

Più si modificano le regole più i soggetti dicono che le frasi sono strane.

Poi hanno fatto risonanza magnetica con frasi corrette e scorrette e visto le attivazioni neurali per le frasi. Tutte le frasi attivano le stesse aree delle frasi corrette è come se il cervello si concentrasse sulla semantica ignorando le strutture di connessione sintattica.

La loro conclusione è che in qualche modo il cervello è in grado di costruire rappresentazioni semantiche di tipo linguistico anche quando sono violate le regole della sintassi.

In sintesi, la semantica è autonoma dalla sintassi perché costruisco frasi corrette indipendentemente dalla correttezza della sintassi.

## Lezione 6 – 18/03

L'altra volta abbiamo parlato di Jekendoff e della rappresentazione internalista del significato e la struttura parallela tra semantica e sintassi che sono indipendenti l'una dall'altra.

Oggi vediamo più in dettaglio la **conceptual semantics**.

### Conceptual semantics

Tutti gli aspetti del significato delle espressioni linguistiche che sono grammaticalmente rilevanti sono codificati dentro strutture simboliche che Jekendoff chiama **conceptual structures**.

Queste CS si interfacciano con la sintassi. Bisogna immaginare che accanto al CS ci sono degli aspetti che non sono rappresentabili simbolicamente. Pensiamo alla differenza che c'è tra un verbo come "uccidere" e "morire".

La differenza sta nel **rappporto di causalità**.

Oppure anche tra "consegnare" e "andare".

In questo caso i due verbi hanno in comune che c'è qualcosa che si sposta.

Abbiamo quindi dei componenti del significato come causalità, movimento che si interfacciano in maniera precisa con la grammatica.

Ci sono anche delle componenti del significato che sono fondamentali nella frase stessa.

Secondo Jekendoff la struttura concettuale codifica i primi elementi ovvero quelli grammaticalmente rilevanti.

Poi ci sono degli altri aspetti del significato che hanno una rappresentazione spaziale o geometrica che non hanno una rilevanza linguistica ma riguardano il nostro modo di concettualizzare.

Per Jekendoff la struttura concettuale codifica quella dimensione del significato rilevanti per comprendere come il sistema si interfaccia con la sintassi e la grammatica.

L'altro è l'**approccio componenziale** all'analisi del significato vige quindi la scomposizione del significato in elementi primitivi di base ovvero le primitive semantiche come per esempio: "concetto di movimento", "concetto di causa".

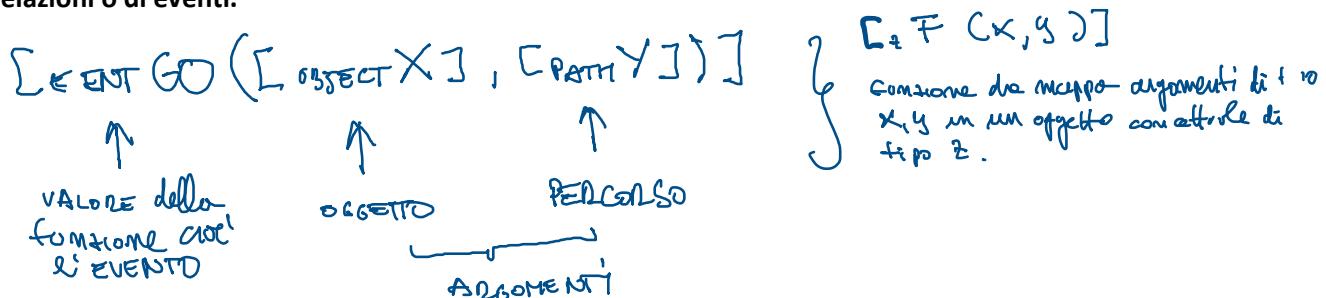
Queste primitive semantiche sono universali e innate. (**Deriva da grammatica generativa**).

Grazie a queste primitive l'individuo analizza il significato e si differenzia dal metodo di Montague.

Per Jekendoff la CS sono sequenze di simboli costruite a partire da due ingredienti principali:

- **Categorie concettuali:** repertorio universale di tipi semanticici divisi per categorie cognitive (oggetto, evento, percorso, luogo).
- **Insieme di funzioni concettuali** ovvero cose come "Go", "To", "in" che sono utilizzate per esprimere il significato del predicato.

Queste **funzioni concettuali** sono delle vere e proprie funzioni che sono diverse da quelle di Montague (le vedeva come entità logiche che producono un valore di verità) qui esprimono una **schematizzazione di relazioni o di eventi**.



Costruzione del significato attraverso funzioni concettuali che creano a loro volta altre rappresentazioni concettuali.

Diversa da Montague **ma anche simile per via del concetto di composizionalità** ma diverse perché qui il risultato è rappresentato da concetti e non valori di verità.

Il concetto di "stanza" ci permette di vedere bene la differenza tra i due pensieri.

Nella visione di Montague stanza è:

Un insieme di elementi, l'insieme delle stanze.

Se dico ho dipinto la stanza intendo che ho dipinto le pareti.

Se dico il tavolo nella stanza intendo la stanza come spazio

Se dico Gianni entra nella stanza questa diventa il punto di arrivo di un percorso.

i pointed [object the room] black

the table is [ place in the room]  $\rightarrow$  [ place IN [object X]

John went [ path into the room]  $\rightarrow$  [ path TO [ place IN [object X ]]]

diverse interpretazioni per il concetto di room:  
seconda del an ente in cui è posto.

Per Jekendoff quindi il linguaggio non si interfaccia con una realtà oggettiva esterna ma con il modo individuale di comprenderla. La stanza è un oggetto fisico 3D dove io dipingo le pareti ma anche come lo

spazio interno occupato dalla stanza... Ci possono essere diverse visioni per un determinato elemento. Il linguaggio si interfaccia con una mia rappresentazione concettuale della realtà.

Jhon went into the room:



Per Jekendoff noi rappresentiamo lo spazio in maniera indipendente dal linguaggio anche se ci sono alcuni elementi che sono dipendenti e servono alla codificazione del linguaggio stesso.

Gli elementi lessicali possiedono delle regole intrinseche che associano vincoli tra fonologia, sintassi e strutture semantiche.

- $\text{TO } \text{GO}$ 
  - Phonology: /goʊ/
  - Syntax: NP<sub>1</sub> V<sub>2</sub> PP<sub>3</sub>
  - Semantics: [EVENT GO<sub>2</sub> ([OBJECT X]<sub>1</sub>, [PATH Y]<sub>3</sub>)]
- $\text{INTO }$ 
  - Phonology: /intəʊ/
  - Syntax: P<sub>1</sub> NP<sub>2</sub>
  - Semantics: [Event TO (Place IN (Object Y))]<sub>1</sub>

### Conceptual Semantics and Compositionality

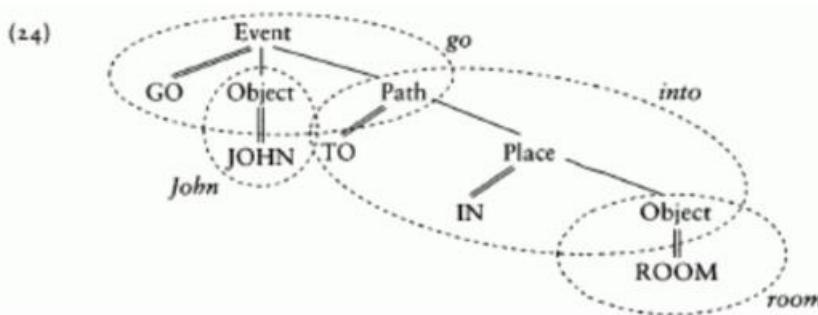
Per Jekendoff la rappresentazione sintattica e semantica sono costruite in maniera parallela e composte tra di loro.

Per comporre si usa la **Typed function application** come in Montague.

In pratica creo un albero che riguarda sia la rappresentazione sintattica che semantica.

Tutto questo non esprime un valore di verità ma una rappresentazione concettuale del mondo.

- (2) a. [S [NP The dog]<sub>1</sub> [VP went<sub>2</sub> [PP into<sub>3</sub> [NP the room]<sub>4</sub>]]].  
b. [EVENT GO ([OBJECT DOG]<sub>1</sub>, [PATH TO ([PLACE IN ([OBJECT ROOM]<sub>4</sub>)]])<sub>3</sub>)]<sub>2</sub>



Una delle cose che caratterizza e rende autonoma la semantica dalla sintassi è il fatto che i termini lessicali differiscono per i frammenti che esprimono e i vari livelli sono associati tra loro.

Questa loro forma di associazione porta a dei disallineamenti.

I livelli sintattici e semantici sono autonomi e vengono collegati tra di loro ma questo collegamento non implica necessariamente una corrispondenza stretta.

Un esempio è la differenza tra "enter" e "go" uno è transitivo e l'altro no, enter codifica dentro se la preposizione INTO. Enter è più ricco di GO, vedi slide.

- TO ENTER
  - Phonology: /enter<sub>1</sub>/
  - Syntax: NP<sub>1</sub> V<sub>2</sub> NP<sub>3</sub>
  - Semantics: [event GO [object X], [path to ([force IN [object Y]<sub>3</sub>])]]
- TO KICK THE BUCKET
  - Phonology: /kick<sub>2</sub> the bucket<sub>3</sub>/
  - Syntax: NP<sub>1</sub> [V<sub>2</sub> NP<sub>3</sub>]<sub>4</sub>
  - Semantics: [event DIE<sub>4</sub> [object X]<sub>1</sub>] ]

Un elemento che compare a livello fonologico e sintattico "bucket" non compare a livello semantico.

Per Montague il linguaggio è esclusivamente e puramente composizionalità che presuppone un rapporto 1 a 1 tra composizione semantica e sintattica.

Nel linguaggio naturale ci sono molte strutture che vengono costruite non composizionalmente ma in maniera olistica perché magari sono molto frequenti.

**La composizionalità è solo uno dei tanti modi che abbiamo di comprendere il significato.**

## Lezione 7 – 19/03

### The assumptions of Fregean Compositionality

Alla base c'è l'idea che la nostra capacità di creare strutture illimitate e complesse è data dal fatto che noi siamo capaci di comporre le varie parti minori.

Ci sono delle assunzioni particolari che non necessariamente fanno parte del linguaggio naturale.

- Il fatto che le **espressioni lessicali siano ambigue e polisemiche non è un problema**, tutto viene dato per scontato nel linguaggio naturale, friggo con l'olio ... è ovvio che non intendo quello della macchina, ho bisogno del contesto.
- Il significato dei termini lessicali non viene modificato nel momento in cui i significati vengono composti l'uno con l'altro. **I significati quindi sono indipendenti dal processo di composizione.**
- I significati di termini lessicali o **la semantica delle espressioni linguistiche è del tutto indipendente dalla pragmatica** ovvero il modo in cui queste espressioni sono utilizzate nei testi.
- Una funzione o riceve argomenti di tipo giusto o altrimenti la funzione non si fa.
- Il significato di un'espressione linguistica è **funzione del significato delle sue parti** ciò implica che tutti gli elementi del significato di una frase devono essere rintracciate nel significato delle loro parti.

Il significato dei termini lessicali è visto come indipendente dal contesto in cui si trova tuttavia la combinazione degli elementi lessicali a volte porta a una modulazione dei loro significati.

Se il significato non fosse context-free significherebbe che il significato cambia a seconda dei contesti in cui questo si trova.

Un significato lessicale deve avere lo stesso contenuto in qualsiasi contesto in cui occorre.

Deve quindi essere indipendente dal contesto.

### Le sfide della composizionalità Fregeiana

- I concetti e i significati sono interamente **sensibili al contesto**.

- (cf. Yee, E., e **Thompson-Schill**, S. L. (2016). "Mettere i concetti nel contesto". *Psychonomic Bulletin & Review*, 23(4), 1015–1027)
- "Collettivamente, gli studi che abbiamo esaminato suggeriscono che **le rappresentazioni concettuali siano fluide**, cambiando non solo in funzione del contesto in relazione alla modalità degli stimoli e al compito, ma anche in funzione del contesto portato da un individuo specifico".
  - Dunque: ci deve essere un elemento di costanza dei significati, ma c'è in realtà più fluidità, portando quindi ai significati context-sensitive → non solo variazione in base al contesto ma anche in base alla singola persona (quindi diverse rappresentazioni concettuali in base al contesto presentato al singolo individuo)
  - Obiezione: come facciamo a comunicare se è vero? Tutti in realtà facciamo esperienze comuni, il linguaggio ha un nucleo costante che i significati delle parole portano con sé (anche se hanno delle parti fluide)
  - Quindi: le parole non sono entità statiche (basta pensare ai concetti astratti rispetto a quelli concreti, come "giustizia" o "libertà", che sono meno legati ad esperienze concrete, ma sono più legati ad una dimensione culturale)
  - Paradosso: significati che devono avere una natura costante per essere compresi ma che hanno anche una natura fluida

Es. sono lo stesso "run" e "open"?

Forse sono lo stesso concetto ad un certo liv. di astrazione (usiamo infatti la stessa stringa per rappresentarlo) ma in realtà in questi casi il verbo cambia molto

- (1)     a. The horse runs.  
          b. The ship runs before the wind.
  - (2)     a. Mary opened the letter from her mother.  
          b. The rangers opened the trail for the season.  
          c. John opened the door for the guests.  
          d. Mary opened up the application.
  - Molto diverso da quello che si immaginava Montague – posso rappresentare "open" come una funzione? Sì, MA porta a dei risultati diversi
- Quindi: l'argomento finisce esso stesso per modificare la funzione. Questa operazione viene chiamata da Pustejovsky fenomeno di **co-composizionalità**
- La composizione semantica non consiste sempre nella mera applicazione di una funzione a un argomento, poiché il predicato stesso può essere modificato dalle informazioni portate dall'argomento.
    - Pustejovsky (1995) chiama questo fenomeno **co-composizionalità**.

- **Metafora**
  - My car drinks gasoline. → uso figurativo del senso di bere ("drink" è infatti usato con un artefatto invece che con un essere umano, portando alla sua "umanizzazione")
  - The Prime Minister is raping democracy → "rape" di solito è usato con esseri umani
- **Trasferimento di riferimento (= metonimia)** → fluidità della capacità di fare riferimento a entità associate alla parola
  - Plato is on the second shelf. → libro di Platone
  - He drank the bottle. → il vino

- The omelette at table 12 wants the bill.
- The horse at lunch was delicious.
- I have Messi on the wall of my bedroom.

- **Metonimia logica**

- The student finished the book. → può voler dire finito di leggere, studiare ecc.
- The student finished the cigarettes. → finito di fumare\*
- Fast car; fast guitarist; fast lane → non esprimono proprio la stessa cosa, complicando il principio di composizionalità

\*dov'è fumare? → sigarette: identificano degli oggetti

Montague: sigaretta = solo l'insieme degli oggetti che identificano le sigarette

- ◆ Queste frasi violano i principi classici di composizionalità di Montague
- ◆ Altra violazione: i vincoli (tipi) semanticci sono continuamente violati
- ◆ Preferenza di selezione = tendenza ad usare determinati argomenti, che però non è assoluta

- Le restrizioni semantiche dovrebbero essere considerate come **preferenze di selezione** (Wilks 1978) piuttosto che come rigide restrizioni di tipo, poiché il loro fallimento non porta sempre a un'anomalia semantica.
- Il processo di composizione semantica deve includere **meccanismi di coercizione** (Pustejovsky 1995, The Generative Lexicon; Asher 2011, Significato lessicale nel contesto; Lauwers e Willems 2011, "Coercizione: Definizione e sfide, approcci attuali e nuove tendenze", Linguistics)
  - per **adattare il significato dei predicati** o degli **argomenti** al fine di **superare le violazioni delle preferenze di selezione**
  - e **aggiungere informazioni** che non sono esplicitamente espresse nell'input linguistico e dipendono dalla conoscenza contestuale
  - MA non tutto è coercibile – non posso combinare tutto con tutto (altrimenti il linguaggio non funzionerebbe). Per esempio, le idee non possono essere verdi.
- **i tipi semanticci:**
  - hanno strutture continue e gradienti, che è difficile definire con simboli categoriali (sono difficili da identificare, infatti hanno categorie molto "sfumate")
  - non possono essere definiti in termini di condizioni necessarie e sufficienti (come nei modelli classici dei concetti)
  - cf. teoria prototipale dei concetti

<b>stipulare</b>	<b>compilare</b>	<b>scrivere</b>
convenzione	modulo	libro
contratto	form	lettera
polizza	scheda	articolo
accordo	modello	post
assicurazione	domanda	testo
mutuo	lista	commento
patto	dichiarazione	storia
trattato	elenco	poesia

## Composizionalità e ambiguità

- Spiegazione a enumerazione dei sensi (Pustejovsky 1995, Il lessico generativo)

- i presunti problemi della Composizionalità Fregeana potrebbero invece essere considerati semplicemente come casi di **ambiguità lessicale**.
  - Ad esempio, sensi differenti di "corri", "veloce", ecc.
- Una volta selezionato il significato appropriato della parola in un dato contesto, la composizione procederebbe secondo l'applicazione standard della funzione.
- Questa soluzione non è soddisfacente perché
  - non tiene conto della stretta relazione tra i sensi delle parole
  - non riesce a modellare la sistematicità, la generalità e la produttività dei processi che portano i lemmi ad acquisire nuovi significati nel contesto. (ci vogliono meccanismi generali per modificare i significati in un contesto)
    - "The director finished the movie"
    - "fast garage; fast racket; fast food; fast book", ecc.

### Oltre la composizionalità Fregeiana

#### Composizionalità arricchita (Jackendoff 1997, The Architecture of the Language Faculty)

- La rappresentazione semantica di una frase **può contenere altro materiale che non è espresso lessicalmente** (quindi il linguaggio è molto più fluido anche rispetto al contesto), ma che deve essere presente:
  - sia per ottenere la correttezza nella composizione delle rappresentazioni semantiche
  - sia per soddisfare la pragmatica del discorso o del contesto extralinguistico.
- Il modo in cui le rappresentazioni semantiche sono combinate è determinato in parte dall'arrangiamento sintattico degli elementi lessicali e in parte dalla struttura interna delle rappresentazioni stesse.

Non possiamo più dire che la sigaretta denota l'insieme delle sigarette, ma le informazioni devono essere rilevanti quando vado a combinare più elementi (quindi tutta una serie di informazioni riguardano la sigaretta)

#### Composizionalità arricchita

- Il principio di un perfetto allineamento tra sintassi e semantica è abbandonato
- **Semantica autonoma** (Baggio 2018, Significato nel Cervello) – non c'è più allineamento tra sintassi e semantica
  - I significati delle frasi e delle frasi sono composti dai significati delle parole più principi indipendenti per la costruzione dei significati, solo alcuni dei quali correlano con la struttura sintattica.
- La competenza semantica include funzioni della Struttura Concettuale che non hanno un correlato sintattico o fonologico e che contribuiscono all'interpretazione di una frase
  - cf. arricchimento pragmatico
  - "Un costituente ha un significato aggiuntivo oltre ai significati delle sue parole. Il significato letterale del costituente, determinato solo dai significati delle parole, è semanticamente incorporato in un significato più ampio, determinato pragmaticamente" (Jackendoff e Wittenberg 2014, "Cosa Puoi Dire Senza Sintassi: Una Gerarchia della Complessità Grammaticale")
  - Es. ellissi → elisione elementi che possono essere ricostruiti dal contesto, mettendo informazioni e aspettandomi che l'altro ricostruisca il contesto

(molto importante: contesto → convenzionalità)

Es. caffè: se chiedo "un caffè" al bar è immediato il suo significato, mentre se lo chiedo in una biblioteca non è scontato che voglia un libro che parli di caffè

#### Enriched composition - Logical metonymy in Conceptual Semantics

- Arricchimento pragmatico: arricchimento messaggio con una serie di informazioni che l'ascoltatore ricostruisce sulla base del contesto
- Palese violazione della composizionalità di Montague, ma è principio fondamentale dell'economia del linguaggio
- Caso metonimia logica:

(7) John began the book

- *begin* → [EVENT BEGIN ([OBJECT X], [EVENT Y])]
- as the direct object is not of type EVENT, a conceptual function introduces an event in the CS and performs a sort of **type-shifting**
  - “Interpret NP as [EVENT F ([OBJECT NP ])]”
  - [EVENT BEGIN ([OBJECT JOHN], [EVENT F ([OBJECT BOOK ])])]
  - *F* is an **implicit event** related to *book* that must be reconstructed
    - what are the principles that govern the reconstruction of the implicit event?

♦ Qui c'è una coercion perché c'è un

mismatch tra il livello linguistico e quello semantico

- ♦ Begin = verbo *aspettuale* (identifica la parte iniziale di un evento) – dal punto di vista semantico prende due elementi, un oggetto e un evento
- ♦ Operazione di coercion: quando a volte c'è il type shifting, cioè una funzione che prende l'evento e lo trasforma in un oggetto che è associato all'evento
  - Book = non è un evento
  - Began = vuole un evento → ci vuole un'operazione di coercione per trasformarlo

Questa operazione di coercion ha un costo cognitivo, che può essere studiato con diverse metodologie, come il self-placed reading e l'eye-tracking. Nel self-placed reading si deve premere un tasto per andare avanti sulla parola successiva e viene misurato il tempo impiegato per passare da una parola ad un'altra.

Entrambe le metodiche registrano i tempi di lettura:

tempo + lento = significativamente più lungo nel caso di una frase type-shifted (libro = non fisico, ma azione associata) → il soggetto ci ha messo più tempo a integrare book con la frase precedente e poi il tempo si riallinea con gli altri due casi

### Il costo cognitivo della composizionalità arricchita

- Gli studi di self-paced reading (McElree et al. 2001) e eye-tracking (Traxler 2002) mostrano che le frasi con metonimia logica hanno bisogno di più tempo per essere processate
  - The author was starting the book (type-shifted)
  - The author was writing the book (preferred)
  - The author was reading the book (non-preferred)
- I costi cognitivi dipendrebbero dalla necessità di risolvere lo “scontro tra tipi” e recuperare l'evento implicito

Table 1  
Mean reading times

	Verb	Determiner	Noun	+ 1	+ 2	+ 3
		<i>the</i>	<i>book</i>	<i>in</i>	<i>his</i>	<i>house</i>
Type-shifted	388	364	377	385	348	355
Preferred	374	358	357	360	334	344
Non-preferred	380	367	380	361	345	349

Altro esempio: studio di Baggio sulla registrazione del segnale dell'attività cerebrale attraverso

l'elettroencefalogramma

Fenomeno più studiato: fenomeno N400

Se ho una frase con qualcosa di semanticamente inatteso (es. il bambino ha bevuto la vodka) in questo caso in corrispondenza della parola target (vodka) il cervello ha una punta di attività (spike) negativa che avviene 400 millisecondi dopo la comparsa della parola. Più alta è la curva, più negativo è il segnale (sull'asse y)

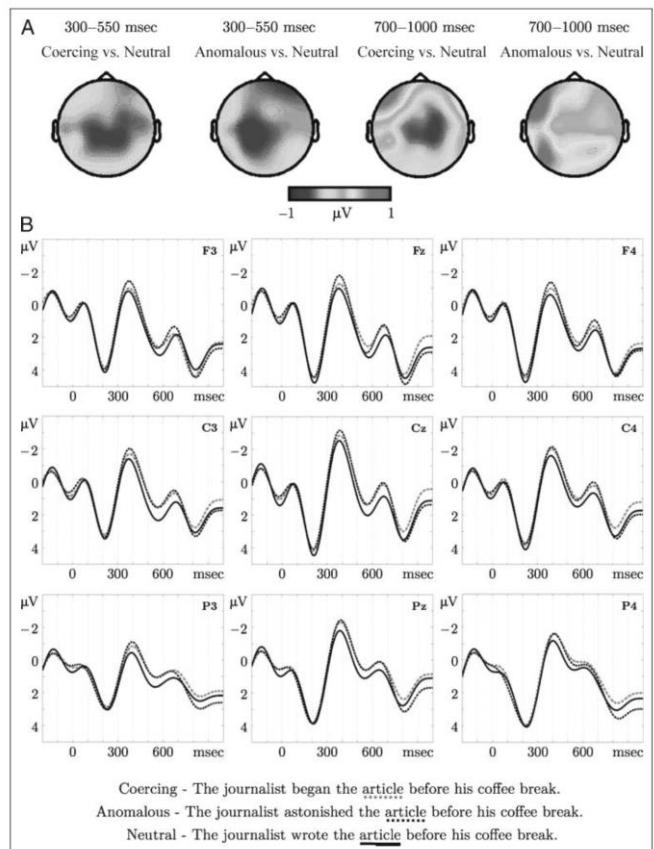
Curve = attività cerebrali registrate

Spike + basso = frase neutra

Spike + alto = frase anomala

Spike intermedio = quello della frase con coercion

Quindi: l'effetto N400 c'è sempre, in quanto processo di integrazione dell'oggetto nella frase, ma cambia in base alla composizione della frase.



## Lezione 8 - 25/03

### The cognitive costs of enriched composition

Ci sono alcuni esperimenti che hanno indagato il rapporto tra la **metonimia tradizionale** e quella **logica**: in questo studio è stato utilizzato il metodo dell'eye tracking: idea più l'occhio si sofferma su certi contenuti della frase, più questo ci indica una complessità.

Le metonimie logiche sono cognitivamente più complesse delle tradizionali.

E.g. - *The gentleman spotted Dickens while waiting for a friend to arrive* (forma convenzionale)

- *The gentleman read Dickens while waiting for a friend to arrive* (metonimia standard)

- *The gentleman started Dickens while waiting for a friend to arrive* (metonimia logica)

Esistono diversi tipi di misura registrate dai tracker: tempo nel quale l'occhio si sofferma su una zona alla

prima volta che ci arriva, quando gli occhi "regrediscono" ovvero ritornano indietro a leggere qualcosa che hanno già letto (indice di difficoltà di comprensione), e il *total time* cioè il tempo totale di lettura di una determinata frase.

I fattori più di interesse per la ricerca sono il *first-pass regression* e il *total time*.

Measure	Verb	Name	Spillover (Next Word)
<b>First-pass time</b>			
Conventional form	338 (11.1)	411 (15.7)	262 (9.8)
Standard metonym	354 (12.6)	392 (15.7)	258 (10.9)
Logical metonym	330 (11.4)	398 (15.3)	279 (9.8)
<b>First-pass regressions</b>			
Conventional form	5.5 (1.5)	11.8 (2.2)	3.0 (1.2)
Standard metonym	7.1 (1.9)	11.2 (2.2)	2.9 (1.0)
Logical metonym	4.8 (1.5)	21.3 (3.3)	5.5 (1.6)
<b>Total time</b>			
Conventional form	448 (21.7)	476 (22.2)	286 (13.5)
Standard metonym	472 (25.1)	474 (20.4)	295 (18.9)
Logical metonym	533 (41.2)	517 (46.0)	338 (23.7)

*Note.* Reading times are in milliseconds and regressions are in percentages. Standard errors are presented in parentheses.

Osserviamo per questi campi dei valori statisticamente significativi, soprattutto nell'ordine delle comparazioni dei vari risultati per le diverse metonimie

Un aspetto molto importante è il grado di convenzionalizzazione. Quanto è conosciuto Dickens influisce direttamente nel capire il verbo associato a questo soggetto che significato esprime.

Un altro esempio è basato proprio su questo ultimo tratto: la convenzionalizzazione.

Confronto tra Dickens e Needham.

TABLE 1  
Sample stimuli

*Unsupported metonym, unfamiliar producer (U-U)*

Not so long before she died, my great-grandmother met Needham in the street.  
I heard that she often read Needham when she had the time.

*Supported metonym, unfamiliar producer (S-U)*

My great-grandmother has all the novels written by Needham in her library.  
I heard that she often read Needham when she had the time.

*Literal, unfamiliar producer (L-U)*

My great-grandmother confessed that she once kissed Needham on the cheek.  
I heard that she often met Needham when she had the time.

*Unsupported metonym, familiar producer (U-F)*

Not so long before she died, my great-grandmother met Dickens in the street.  
I heard that she often read Dickens when she had the time.

*Supported metonym, familiar producer (S-F)*

My great-grandmother has all the novels written by Dickens in her library.  
I heard that she often read Dickens when she had the time.

*Literal, unfamiliar producer (L-F)*

My great-grandmother confessed that she once kissed Dickens on the cheek.  
I heard that she often met Dickens when she had the time.

Questi confronti sono basati proprio sul fattore conoscenza/non conoscenza del personaggio. Needham è uno scrittore poco conosciuto quindi dà vita ad una metonimia poco convenzionalizzata a differenza del caso Dickens.

La differenza significativa risiede nel fatto che non si sa che N. È uno scrittore; quindi, il problema si pone solo quando non viene specificato dal contesto il suo lavoro. Nell'altro è esattamente come per Dickens.

Quando il contesto ci da questa informazione il nostro cervello

processa in maniera diversa la frase. Se il contesto non ci dà l'informazione, il cervello ha bisogno di elaborare maggiormente la frase per capirne il significato (quindi di conseguenza possono accadere più frequentemente casi in cui l'eye tracking ha una regressione ecc....)

Quindi la **composizionalità arricchita** si basa sulla capacità dell'ascoltatore di usare la propria conoscenza per fare questo arricchimento.

I casi di metonimia e metonimia logica sono casi in cui c'è un arricchimento di informazioni non fornite dalle frasi (in diversi campi).

### La conoscenza contestuale

Esistono diversi modelli che hanno cercato di dare una soluzione al discorso legato alla conoscenza contestuale. Uno di questi è il **Generative Lexicon (GL)**.

Si basa sul principio che la composizionalità non sia una mera combinazione di significati lessicali, ma basata su processi generativi che permettono alle parole di acquisire nuovi sensi basati sul contesto.

GL perché? Generativo è da intendere nello stesso modo della grammatica generativa, cioè la generazione di un numero potenzialmente illimitato di frasi, poiché **basandosi su una serie di regole finite si arriva ad una ricchezza infinita**. Il lessico è l'insieme finito di sensi della lingua. Nel GL il lessico ha già un suo contenuto significato, ma poi interagendo con altri elementi del lessico nella frase si hanno nuovi significati. Quindi il lessico non è predefinito ma si costruisce all'interno del processo di composizione. La composizionalità avviene perché le strutture del lessico contengono un significato più ricco. Le entrate lessicali sono caratterizzate da un complesso di informazione più ricca che comprende una struttura fondamentale: **Qualia structure**.

La Qualia structure contiene 4 fondamentali conoscenze:

- 1- **Formale:** riguarda la struttura tassonomica degli elementi del lessico (*cancello* è un tipo di barriera, *penna* un tipo di strumento)
- 2- **Aggettivo:** riguarda la modalità di creazione di un'entità (come nasce il vino, una sigaretta)
- 3- **Telica:** scopo intrinseco di una determinata entità (es. Il cancello ha come scopo quello di chiudere, la penna di scrivere)
- 4- **Costitutiva:** riguarda le parti dell'oggetto (come questo è composto, libro composto da argine di carta, ecc...)

book
ARGSTR = $\begin{bmatrix} \text{ARG1} = \text{x:info} \\ \text{ARG2} = \text{y:physobj} \end{bmatrix}$
QUALIA = $\begin{bmatrix} \text{info-physobj-lcp} \\ \text{FORMAL} = \text{hold(y,x)} \\ \text{CONST} = \text{part\_of(z:page,y)} \\ \text{TELIC} = \text{read(e,w,x)} \\ \text{AGENT} = \text{write(e,v,x)} \end{bmatrix}$

Tutte queste conoscenze sono aspetti importanti: la conoscenza del mondo viene inserita nelle entrate lessicali (cioè, noi sappiamo cosa è e come è fatto un libro, questa informazione viene inserita nel lessico e la composizionalità del lessico rende queste informazioni esplicite).

Le entrate lessicali del lessico generativo sono arricchite da informazioni provenienti da eventi salienti che descrivono quella determinata entità.

E. g. *Gianni ha letto un libro – Gianni ha iniziato un libro*

Le entrate lessicali vengono integrate di ulteriori informazioni per la GL.

La struttura qualia ha come scopo quello di rappresentare una sorta di interfaccia tra il lessico e la nostra conoscenza del mondo.

Esistono nel mondo delle informazioni che sembrano legate alla nostra **conoscenza semantica** e altre legate alla **conoscenza fattuale** del mondo.

Es. Sapere che il cipresso è un albero significa capire il significato intrinseco di cipresso

Es. Sapere che il cipresso è un sempreverde e che è nelle zone mediterranee significa avere la conoscenza fattuale del mondo.

Uno scienziato francese ha testato frasi di questo tipo:

*"I treni olandesi sono gialli" - "i treni olandesi sono blu" - "I treni olandesi sono divertenti".*

La conoscenza fattuale ci dice che è vera la prima frase (i treni sono davvero gialli), la seconda frase è plausibile ma errata quindi non è una conoscenza fattuale, la terza frase è semanticamente errata.

Tra conoscenza del contesto (i libri si comprano in libreria e non dal macellaio) e conoscenza semantica (i libri si leggono) esiste differenza? Sono tipologie di informazioni diverse? **Per Pustejovsky c'è differenza.**

*"Possiamo pensare ai qualia come quell'insieme di proprietà o eventi associati a un elemento lessicale che meglio spiegano cosa significhi quella parola."* - Pustejovsky

*John began the book*

*Began* è un evento, *book* non lo è. Il modo sta nel fatto di non calcolare *began*, ma prendere il verbo dentro il termine lessicale ovvero in questo caso dentro *book*.

Questo metodo però non sempre funziona e non tiene conto di alcuni fattori: nel caso di leggere la soluzione è semplice va a prendere la conoscenza telica e così risolve, ma non c'è un metodo, una regola che vale per tutti i casi; in alcuni non si risolve così facilmente. P. Definisce solo un confine tra la conoscenza del mondo e il lessico rappresentato dal Qualia.

- *John enjoyed the book* (l'evento implicito è uno scopo tipico per la parola *book*)

- *John enjoyed the garden (sun, sea, wind, the birds, etc.)* (qui l'evento implicito non soddisfa la definizione di Qualia)

Il verbo "apprezzare" è complicato da analizzare con questo metodo. Es. *John enjoyed the sun* (*enjoy* sicuramente non è nel qualia di *sun*, cioè non è una funzione prototipica di sole). Inoltre, non è solo il complemento oggetto che determina la metonimia logica, ma anche altre parti del discorso; quindi, l'evento implicito non è esclusivamente determinato dal complemento oggetto. Es. *The cleaner began the suit*. La funzione prototipica di *suit* è indossare, quindi cosa ha iniziato il *cleaner*? A lavare il vestito, qui dipende dal soggetto e non dall'oggetto.

Altro metodo:

### **GEK (Generalized Event Knowledge) – McRae & Matsuki 2009**

Non esiste un confine tra elementi del significato ed elementi del contesto. Elman scrive un articolo importante nel 2011. Qui ci parla di **conoscenza lessicale**: ovvero insieme di conoscenze attivate da insiemi di parole. Modello che vede le **parole come attivatori**. Noi abbiamo una serie di conoscenze derivate dalla nostra esperienza e dalle nostre conoscenze. Una parola evoca un preciso script o frame di significato, cioè tutta una conoscenza legata ad una parola e grazie a queste informazioni siamo in grado di integrare il significato di un preciso contesto.

I termini lessicali attivano la GEK.

Quali sono le **sorgenti** di queste informazioni?

- **Esperienze dirette** nel fare o osservare gli eventi
- **Esperienze linguistiche** (info apprese leggendo).

**Semantic Priming**: idea che la nostra conoscenza del mondo è una rete di associazioni tra concetti lessicali che sono in grado di attivarsi reciprocamente.

Es. *Arrestare* evoca *ladro*, ma anche *poliziotto*, ma anche *prigione*. E lo stesso avviene al contrario.

### **Nell'esperimento del SP cosa avviene?**

Viene presentata una parola chiamata prime (attivatore) per pochi secondi per poi scomparire e sopraggiungere la parola target che viene utilizzata per creare un task di decisione, cioè colui sottoposto all'esperimento deve capire se è una parola o no, si misurano i tempi di reazione.

Per il semantic prime poi si studia se esiste un legame tra la parola prime e quella target, nel caso in cui esiste questa relazione allora il tempo di risposta alla prima domanda sarà più breve. Questo perché avviene? Quando si dice "*cat*" come prime e "*dog*" come target è ovvio che il tempo sarà breve, per la nostra conoscenza del mondo abbiamo in mente entrambe come animali, o come significati associati tra loro in qualche modo e che quindi vengono coattivati.

### **GEK e metonimia logica(Zarcone et al.)**

Data una frase, dopo averla pronunciata viene data un parola che è o no contenuta nella frase. Chi viene sottoposto all'esperimento deve dire sì se è contenuta o no se non lo è. È stato rilevato spesso un errore in questo tipo di risposta: spesso chi ascolta ha dato un risultato errato (es. Ha detto sì ma la risposta giusta era no) quando la seconda parola proposta era una parola contenuta nel semantic prime di qualche altra parola evocata dalla frase.

Esempi di errori quotati come alti:

- E.g.    - *The baker finished with the icing* -> SPREAD  
       - *The child finished with the icing* -> EAT

Esempi di errori bassi:

- e.g.    - - *The baker finished with the icing* -> EAT  
       - *The child finished with the icing* -> SPREAD

### Producing logical metonymies

Altro esempio di questo tipo.

La metonimia logica si alterna con le espressioni esplicite del verbo nascosto (costruzione EXP) come in altri casi di composizionalità arricchita:

- *The student began the book* (metonimia logica)
- *The student began reading/to read the book* (EXP)

I principi che governano la produzione delle metonimie logiche sono quello che Levinson (2000) chiama l'**I-principle** (in relazione alla massima quantità di Grice): il parlante dovrebbe produrre un input linguistico minimo per far sì che l'ascoltatore possa capire il messaggio, basandosi su ipotesi riguardanti la conoscenza pregressa dell'ascoltatore (massima della minimizzazione).

L'ascoltatore dovrebbe usare la sua conoscenza del mondo per trovare l'interpretazione più specifica possibile del messaggio del parlante, basandosi sull'assunzione che il parlante detenesse quell'informazione implicitamente (regola di arricchimento).

## Lezione 9 - 15/04

### Modelli vettoriali

Parlando di Montague o di *concept semantics* abbiamo parlato di **modelli simbolici**: rappresentazione delle forme linguistiche tradotte con un metalinguaggio formale.

Inseguire nella rappresentazione di Jackendoff:

v. *Chase*:

[GO [soggetto] [oggetto]]  
(EVENTO)

Questo tipo di approccio nasce per rappresentare composizionalmente la realtà.

**Modello vettoriale: modello per rappresentare le differenze lessicali.**

**Postulati di significato:** con Montague e la sua semantica c'è poca postularità. Montague, Karmapp, si sono posti domande del tipo semantico: Karmapp introdusse il concetto di postulato di significato (**meaning postulate**).

Cosa è?

Asserzioni del tipo:

*Se x uccide y: X uccide y se e solo se x è causa di y di diventare morto.*

- $\text{kill} \rightarrow \lambda x \lambda y. [\text{kill}(x, y)] \Leftrightarrow \lambda x \lambda y. [\text{CAUSE}(x, \text{BECOME}(\text{DEAD}(y)))]$

Questo tipo di rappresentazione ha un vantaggio: **rendere esplicito** e quindi spiegabile l'inferenza legata ad un verbo come *kill*. Cioè, si cerca un legame tra uccidere e morire.

Questo metodo ci dà la spiegazione del perché è possibile o meno un certo tipo di inferenza.

Ha però dei punti deboli legati alla dimensione lessicale.

Questi modelli non ci dicono come selezionare in modo giusto il significato di un certo tipo di contesto.

In questo metodo si assume che ci sia un significato univoco; quindi, non ci poniamo il problema della polisemia.

Nell'esempio, vediamo i significati associabili alla parola scuola:

*bat* → **bat<sub>1</sub>** (type of mammal); **bat<sub>2</sub>** (type of artifact)  
*school* → **school<sub>1</sub>** (group of fish); **school<sub>2</sub>** (location); **school<sub>3</sub>** (institution); **school<sub>4</sub>** (time), **school<sub>5</sub>** (group of people) etc.

Questi modelli simbolici inoltre hanno una notevole difficoltà nel capire che i termini lessicali cambiano il significato all'interno del contesto in cui ricorrono:

*the plane flies vs. the car flies*

*red hair vs. red wine*

Rosso inteso in due modi differenti: *red* dei capelli inteso come tendente all'arancio, quello del vino verso il bordoux.

Altro problema: come apprendiamo noi il significato delle parole?

L'acquisizione del significato è un compito complesso per gli umani che dipende da tanti fattori, ma ancora peggio avviene per l'AI. Deve essere in grado di capire i diversi significati.

Altro problema: come cambia il significato nel tempo?

Vengono associati progressivamente nuovi significati alle parole, e si deve tener conto anche di questo.

**Punto chiave:** relazione tra il significato delle parole e l'uso effettivo delle parole nel contesto.

Questo concetto è indispensabile per capire tutti gli usi della lingua e del significato. C'è dunque una forte relazione tra il significato delle parole e dei contesti, che i metodi tradizionali non prendono in considerazione. Vengono presi dei **significati context free** e si procede con l'analisi (come già spiegato).

Limiti delle rappresentazioni del significato simboliche:

- Rappresentare il significato con simboli traducendoli in altri simboli (**rappresentazioni di tipo discreto e qualitativo** che hanno difficoltà nel trattare tutti gli aspetti della semantica che hanno caratteristiche graduali, continue, ecc...)  
Nell'esempio di *school*, *school* è una variabile discreta che varia significato grazie al pedice, ma la distanza semantica che hanno questi significati tra di loro, non è uguale tra pedici diversi. Il problema delle variabili qualitative è che possiamo solo dire se sono identiche o meno, ma non quantificare la distanza effettiva.
- **Rappresentazioni troppo stipulative o a priori:** nell'esempio *kill* e *dead*, siamo noi che scegliamo a priori delle strutture semantiche (elementi primitivi semanticci) da usare nell'analisi. Siamo noi che stipuliamo gli aspetti semanticci primitivi.
- **Difficoltà nel fornire dei modelli che racchiudono elementi cognitivamente sensati** nel rappresentare il cambiamento contestuale del significato o richiedono di complicare il meccanismo semantico.
- **Difficoltà nel rappresentare significati multimodali** tipiche della nostra parte cognitiva. Parte del significato e questi simboli sono intrinsecamente annodati. "Il significato di rosso è un simbolo che noi chiamiamo RED". Ma come andiamo ad attaccare RED alla realtà esterna? Bisognerebbe attaccarci un vettore nello spazio dei colori.
- **Mancanza di metodi effettivi di apprendimento:** non abbiamo modelli di apprendimento di questo tipo; quindi, modelli che comprendono e acquisiscono l'informazione semantica simbolica.

Tutta questa serie di problemi ha portato ad esplorare altri modelli di rappresentazione dei significati.

## Rappresentazione del significato con vettori

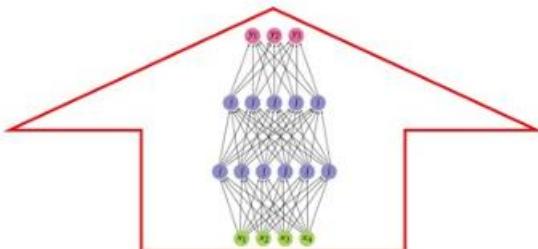
Visione diversa: non si associano più significati a simboli ma rappresentiamo invece i significati come dei vettori, cioè sequenze di numeri reali. Questi codificano l'informazione ricavata dai dati impliciti ricavata (soprattutto) da dati recanti informazioni testuale, ma anche modale (immagini). Questo processo è svolto da modelli computazionali che estraendo questa informazione da dati impliciti e la codificano all'interno di vettori.

Queste informazioni sono utilizzate anche dai modelli recenti di AI.

Esempio del vettore di RED:



Vettore contenente le diverse gradazioni di rosso.



Vantaggi di questi modelli:

- **Ammettono rappresentazioni continue** come sequenze di numeri che permettono quindi la **gradualità**. Inoltre, **sono distribuite**: le informazioni semantiche non sono localizzate in nessuna specifica posizione del vettore, ma sono distribuite lungo tutto il vettore; non è possibile dunque individuare un rapporto 1:1 tra il significato e un unico tratto semantico.
- **Non esiste nessuna primitiva semantica definita a priori**, sono **modelli del tutto data-driven**, totalmente intuitivi. Tutte le informazioni ricavate con questo metodo derivano direttamente dai dati;
- **Esistono modelli per imparare la rappresentazione semantica**: forniscono regole su come si ricava l'informazione semantica per addestrarne altri;
- **Permettono di integrare informazione multimodale**: l'uso di vettori numerici permette di rappresentare informazioni ricavate da altri modelli, quindi da più sorgenti diverse.

Questa tipologia di rappresentazione è la **semantica distribuzionale**.

Caratteristica base: rappresentare il contenuto delle espressioni linguistiche come vettori, e il contenuto di questi vettori venga usato nel linguaggio.

Nel linguaggio ci sono delle parole i cui significati apprendiamo ancor prima di usare il linguaggio (capacità tipica dei bambini). Altre volte ci sono parole i cui significati li apprendiamo esclusivamente grazie al linguaggio (significati astratti, o specifici di certi campi di studio, ecc...).

Lavorare su dati sia sensoriali che linguistici ha portato a capire che molti significati si capiscono esclusivamente analizzando dati linguistici. Ecco perché sistemi come ChatGPT funzionano: avendo miliardi di testi da cui apprendere, ha capito dai testi i significati da associare ai termini linguistici. Questo apprendimento funziona ancora di più se prendiamo in riferimento anche le immagini (GPT4).

## Semantica distribuzionale

**Semantica distribuzionale:** ricavare il significato delle parole dall'uso di queste parole nei contesti diversi.

Asserzione finale: Il modo con cui le parole vengono usate nel linguaggio condizionano anche i nostri significati mentali.

Si può modellare la semantica attraverso l'analisi statistica nei contesti linguistici delle parole.

Si rappresentano i significati delle parole con vettori di lunghezza n dove n è il numero delle caratteristiche associate a tali significati, i cui componenti rappresentano le co-occorrenze di tale parola nei diversi contesti linguistici.

**Ipotesi distribuzionale:** i lesseni con simile distribuzione delle proprietà, hanno significati simili. Questi vettori hanno preso il nome di embeddings (word o sentence). Alla base della semantica distribuzionale c'è un rapporto tra le proprietà distribuzionali dei termini linguistici e il significato. Se c'è similarità tra le distribuzioni c'è una similarità effettiva nei significati.

### Come si passa dai contesti alle distribuzioni semantiche?

Vedremo vari modelli semanticamente distribuzionali (DSMs): modelli che costruiscono vettori estraendo dati testuali.

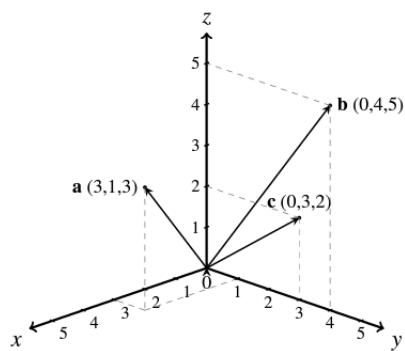
Es.

... dig a [hole. The	car <i>drove away] leaving behind ...</i>
... to directly [ <i>drive</i> the	car <i>wheel angle] 3. Force ...</i>
... to pet [ <i>the family's</i>	cat <i>and dog.] who tended ...</i>
... and then [ <i>wanted a</i>	cat <i>to eat] the many ...</i>
... bank, children [ <i>playing with</i>	dogs <i>and a] man leading. ...</i>
... vegetable material [ <i>and enzymes.</i>	Dogs <i>also eat] fruit, berries ...</i>
... hubby once [ <i>ate the</i>	dog <i>food and] asked for ...</i>
... go down [ <i>as the</i>	van <i>drove off.] As he ...</i>
... heavy objects, [ <i>driving transit</i>	vans <i>, wiring plugs] and talking ...</i>
... of the [ <i>fast food</i>	van <i>being located] outside their ...</i>
... each of [ <i>the six</i>	van <i>wheels , and] also under ...</i>

### Vettori

Un vettore di numeri reali è una lista di numeri reali, dove ogni numero  $v_i$  è la componente i-esima del vettore.

I vettori di n componenti definiscono punti (frecce) n-dimensional in uno spazio vettoriale.



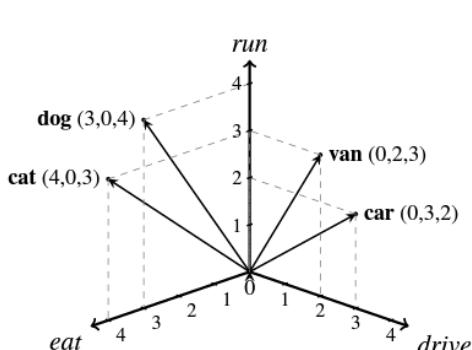
I vettori vengono usati per rappresentare oggetti con **attributi quantitativi o features**.

Un **vettore di features** per l'oggetto  $o$  è l'insieme delle features numeriche che caratterizzano  $o$ .

**Definizione di rappresentazione distribuzionale:** La rappresentazione distribuzionale di un elemento lessicale è un vettore distribuzionale n-dimensionale, i cui componenti sono caratteristiche distribuzionali che rappresentano le sue co-occorrenze con contesti linguistici.

Es.

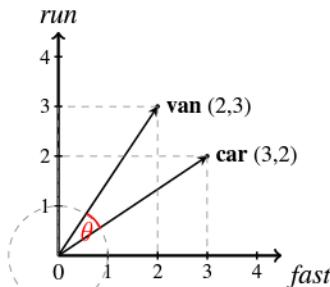
$\text{car} = (0, 3, 2)$   
 $\text{cat} = (4, 0, 3)$   
 $\text{dog} = (3, 0, 4)$   
 $\text{van} = (0, 2, 3)$



## Misurare la similarità distribuzionale

La similarità distribuzionale tra due lessemi  $u$  e  $v$  è misurata come la similarità tra i loro vettori distribuzionali  $\mathbf{u}$  e  $\mathbf{v}$ .

$$\text{Cosine } \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| |\mathbf{y}|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$



Data l'Ipotesi Distribuzionale, i lessemi distribuzionalmente simili tendono ad essere semanticamente simili (correlati).

<i>car</i>	1			
<i>cat</i>	0.33	1		
<i>dog</i>	0.44	0.96	1	
<i>van</i>	0.92	0.50	0.66	1
	<i>car</i>	<i>cat</i>	<i>dog</i>	<i>van</i>

## **Lezione 10 – 16/04**

La semantica distribuzionale ha le sue origini nella tradizione strutturalista americana.

Lo strutturalismo americano è l'approccio tipico prima della distribuzione generativa e di Chomsky.

Uno dei pionieri è Zellig Harris, un linguista che è allievo di Bloomfield e maestro di Chomsky.

Il criterio distribuzionale era il criterio fondamentale per determinare le varie funzionalità linguistiche.

Harris eredita da Bloomfield il rifiuto del significato come spiegazione nella linguistica. L'analisi semantica può ricevere una solida base empirica solo attraverso l'approccio distribuzionale.

Due strutture linguistiche appartengono alla stessa categoria se hanno gli stessi rapporti distribuzionali tra di loro.

Quindi per definire un nome andiamo a vedere i modi in cui le espressioni linguistiche si distribuiscono nei testi.

Harris in *Distributional structure* parla di questo metodo distribuzionale.

Differenze di significato si correlano con differenza di distribuzione.

Harris non ha mai enunciato il termine semantica distribuzionale nei suoi lavori, tuttavia, tutto questo è coerente con una definizione di semantica distribuzionale del significato.

Perché Harris propone questa visione distribuzionalista?

- Harris come Bloomfield ha qualcosa in comune con l'approccio antipsicologico della linguistica.

L'idea è che la linguistica non può essere fondata su concetti come intendere il significato dal punto di vista psicologico. L'approccio distribuzionale diventa un metodo per fondare l'analisi semantica su forti basi empiriche.

ChatGPT usa la coerenza semantica.

Si comporta in maniera abbastanza simile all'umano.

Quello che noi osserviamo è come il computer cerca di utilizzare delle parole in modo più simile a quello umano.

Per Harris si vuole dare alla semantica una base scientifica senza poter quindi entrar nella mente del parlante e comprenderne il lato psicologico della rappresentazione del significato.

### **Rappresentazione di tipo sintagmatico: co-occorrenze**

Si differenziano dal tipo paradigmatico.

I tipi semanticamente sono quindi classi di elementi che condividono proprietà che gli permettono di combinarsi tra di loro.

I Language Models fanno esattamente questo.

Le **relazioni sintagmatiche** (anche chiamate sintagmatiche di vicinato o associate) hanno le seguenti caratteristiche:

- Sono parole che co-occorrono vicine tra loro
- Sono relazioni combinatorie (frasi, enunciati, etc.)
- Vedi le co-occorrenze di primo ordine:

I	drink	coffee
you	sip	tea
they	gulp	cocoa

Le **relazioni paradigmatiche** hanno invece le seguenti caratteristiche:

- Sono parole che non co-occorrono ma invece co-occorrono con gli stessi vicini sintagmatici
- Sono relazioni sostituzionali
- Vedi le co-occorrenze di secondo ordine:

I	drink	coffee
you	sip	tea
they	gulp	cocoa

### **I pionieri della semantica distribuzionale**

#### **Paul L. Garvin**

Uno dei maggiori esperti di traduzione automatica degli anni 60.

Lui pubblica un articolo dove fa vedere come la nascente informatica può fornire un contributo all'analisi del linguaggio.

La semantica distribuzionale e quindi i vari elementi condividono proprietà relative al contesto linguistico.  
È possibile raggruppare tutte quelle unità linguistiche e queste possono formare un insieme semantico.

Da sempre nella linguistica computazionale c'è stata la domanda di come fornire informazioni semantiche al computer, chiaramente glielo possiamo dire noi che caffè tè e acqua formano la classe dei liquidi ma questo ovviamente non è efficiente.

Bisogna quindi far sì che il modello impari dai testi.

#### **John R. Firth**

Un altro pioniere degli approcci distribuzionali è John Firth che è il **padre della nozione di collocazione**.

Noi possiamo comprendere il significato delle parole dai "giochi linguistici" ovvero dall'uso quotidiano delle parole nel linguaggio.

Ci sono delle parole che vengono utilizzate in maniera più tipica insieme ad altre parole.

Le **collocazioni** sono co-occorrenze di tipo sintagmatico di unità lessicali nel medesimo contesto linguistico:

*“Le collocazioni di una data parola sono affermazioni dei luoghi abituali o consueti di quella parola nell’ordine collocazionale [...] è un ordine di aspettativa reciproca. Le parole sono reciprocamente aspettate e reciprocamente comprese”* (Firth 1957: 12).

Esempi: *estinguere un debito, indire le elezioni, acceso dibattito, a caro prezzo, ecc.*

Il significato di un lessema è definito dalle sue collocazioni, altri lessemi che hanno relazioni sintagmatiche con esso (**Contextual View of Meaning**).

Noi quindi possiamo comprendere le parole dal modo in cui vengono usate nel contesto linguistico – Wittgenstein filosofo della lingua.

Lo studio delle collocazioni e gli strumenti utilizzati li ritroviamo nella semantica distribuzionale.

Qui i significati delle parole esistono solo in relazione al contesto, rovesciamento del pensiero che i significati delle parole siano context-free.

Sia Harris che Firth sono lontani dal tipo cognitivistico della lingua. La mente è qualcosa di non definibile. Questa cosa ha attirato però molti psicologi e scienziati cognitivistici come Miller:

*“Un linguista definisce la distribuzione di una parola come l’elenco dei contesti in cui la parola può essere sostituita; la somiglianza distribuzionale di due parole è quindi la misura in cui possono essere sostituite negli stessi contesti. [...] Diversi psicologi hanno inventato o adattato variazioni su questo tema distribuzionale come metodo empirico per investigare le somiglianze semantiche.”* (Miller 1967)

Loro possono quindi definire scientificamente il concetto di similarità semantica.

### **Significati e i loro “osservabili”**

Ma come è possibile quindi misturare la similarità semantica dal punto di vista cognitivo?

In qualche modo si può usare la semantica distribuzionale per misurare la similarità tra parole – Miller.

Vedremo quindi come la semantica distribuzionale è molto usata anche per studi cognitivi.

La similarità distribuzionale può quindi diventare un modo per trovare degli “osservabili” nel significato.

Chiaramente dico osservabili perché il significato non è osservabile direttamente, e sono:

- I giudizi dei soggetti (ad es. raccolti in studi);
- Il comportamento dei soggetti in compiti psicolinguistici (ad es. tempi di reazione, tempi di lettura, etc.);
- Il modo in cui le espressioni linguistiche sono usate nel contesto.

Possiamo tenere di conto del giudizio delle persone sulla similarità ma questo è costoso (tempo, remunerazione per l’indagine, etc.).

Possiamo quindi in questo modo avere degli indizi sul funzionamento di queste parole: non sappiamo come è collocato il significato nella mente dei soggetti e quindi utilizziamo tecniche di questo tipo per comprendere come viene percepito il significato.

Noi rappresentiamo semanticamente parole vicine perché le abbiamo viste essere utilizzate in contesti simili.

**L’approccio distribuzionale diventa un modo per studiare il significato in maniera empiricamente fondata.**

L’approccio distribuzionale è anche un modo in cui le mie rappresentazioni semantiche si formano.

“La rappresentazione contestuale di una parola è la conoscenza di come quella parola viene usata nei diversi contesti linguistici, dunque, due parole sono simili se le loro rappresentazioni contestuali sono simili” – Miller.

### **Dalle distribuzioni linguistiche al significato**

Noi quindi ci creiamo delle rappresentazioni mentali in base a come percepiamo le parole in diversi contesti linguistici. Come le parole vengono usate determinano anche la percezione semantica delle parole nella nostra mente:

es. *He ate the gorp with the fork* oppure *The little, spotty gorp was jumping and barking behind the tree*

**Syntactic bootstrapping:** teoria in cui si studia come i bambini imparano il significato delle parole. Prima imparano le parole relative all’ambiente e poi quando imparano il linguaggio utilizzano i significati imparati per imparare nuovi tipi di verbi.

Quando impariamo nuovi termini li impariamo grazie alla nostra esperienza extra linguistica, sensoriale, ma soprattutto dall’input linguistico, dalle esperienze linguistiche ovvero dal fatto che noi sentiamo parlare della realtà e quindi dal linguaggio che tratta della realtà.

Nella visione esternalista ci si discosta completamente da questo metodo: linguaggio e mondo sono separati, qui non è assolutamente così.

In realtà il linguaggio costruisce il mondo. Le rappresentazioni semantiche, quindi, contengono sia info sul mondo esterno ma anche le info sul rapporto tra le parole e le strutture linguistiche.

Quindi visto che sia noi che ChatGPT impariamo dal linguaggio possiamo dire che in parte siamo simili.

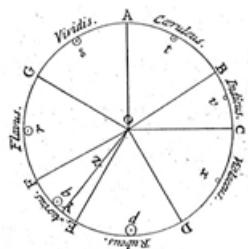
Le espressioni linguistiche diventano testimonianza del **propagation of grounding**.

Ovvero io non ho mai visto un elemento x ma se questo elemento x ricorre insieme a certi altri elementi linguistici, in base agli elementi linguistici contestuali io capisco cosa sia quell’elemento e me lo posso immaginare propagando la mia esperienza sensoriale.

*“Dato un insieme di concetti per i quali abbiamo un ancoraggio diretto nelle esperienze sensoriali-motorie, le co-occorrenze linguistiche determinano l’ancoraggio indiretto di un numero molto maggiore di concetti”* (Vincent-Lamarre et al., 2016; Günther et al., 2020)

### **Blindness and language**

In Marmor (1978), i giudizi di somiglianza da parte dei soggetti ciechi congeniti riguardo ai termini del colore approssimavano la ruota dei colori di Newton.



Molto studiato il fatto che non vedenti congeniti hanno delle conoscenze sui colori che sono estremamente simili a quelli umani. Sanno riconoscere che arancione e rosso sono più simili tra verde e blu.

Kelli, una bambina cieca di 5 anni, aveva acquisito i termini comuni dei colori e i verbi della percezione visiva (come “guarda” e “vedi”), insieme ai loro vincoli semantici, ad esempio i termini dei colori si applicano solo agli oggetti con estensione spazio-temporale, come cani e macchine.

## The redundancy hypothesis

Come è possibile che allora i non vedenti riescano a capire?

L'idea è che l'informazione senso-motoria è **codificata in maniera ridondante** anche nel linguaggio e nell'esperienza percettiva; il linguaggio quindi **si è adattato** a parlare anche della realtà dunque molta dell'informazione codificata senso-motoria appare nelle nostre esperienze linguistiche.

**L'ipotesi della ridondanza predice che la mancanza di esperienze visive possa essere compensata con le informazioni acquisite attraverso il linguaggio.**

Per questo anche ChatGPT è così avanti poiché le strutture linguistiche sono presenti nel linguaggio e nei testi su cui impara.

## **Lezione 11 – 22/04**

Nella lezione precedente abbiamo parlato dell'ipotesi distribuzionale (**Weak Distributional Hypothesis**).

Possiamo dire che la semantica distribuzionale è un metodo empirico per l'analisi semantica.

Si tratta di una metodologia dove abbiamo l'idea che indipendentemente dal mondo psicologico della comprensione del significato è ovvio pensare che il significato sta dentro la distribuzione delle parole all'interno del contesto linguistico.

Sfruttiamo il fatto che le parole vengono usate in modo diverso a seconda delle proprietà semantiche per dedurre queste proprietà semantiche (parole che appaiono in contesti simili tendono ad avere significati simili).

C'è un altro approccio della semantica distribuzionale:

### Strong Distributional Hypothesis

Ipotesi cognitiva sulla forma e le origini delle rappresentazioni semantiche.

La distribuzione delle parole nel contesto ha uno specifico ruolo causale sul modo in cui si formano le rappresentazioni semantiche per quella parola.

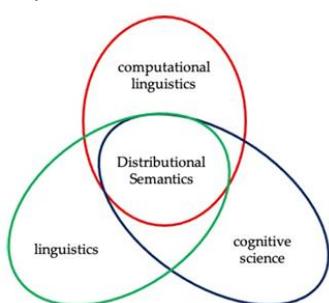
Le proprietà distribuzionali delle parole nei contesti linguistici sono un fattore esplicativo della competenza semantica umana.

Il significato nasce andando a registrare le proprietà distribuzionali delle parole.

Il modo in cui sentiamo parlare di qualcosa influenza il nostro punto di vista semantico su quella determinata cosa.

Dire “*banda di scienziati*” anziché “*gruppo*” associa una connotazione negativa.

Le parole condizionano in modo creativo la nostra percezione.



Le distribuzioni semantiche sono al centro di molti studi linguistici.

## Distribuzionale semantics: main assumptions

L'ipotesi fondante della semantica distribuzionale è **l'ipotesi di correlazione tra proprietà semantiche** delle parole e uso di queste nel linguaggio. L'ipotesi distribuzionale è principalmente una congettura sulla

similarità semantica, modellata come una funzione della similarità distribuzionale: **i lessemi che hanno proprietà distribuzionali simili hanno significati simili.**

Il focus dell'ipotesi distribuzionale è sempre stato il lessico, proprio perché si basa su una congettura sul significato lessicale delle parole; oggi la situazione non è più totalmente così. La semantica distribuzionale si basa su una visione contestuale e basata sull'uso del significato: il significato del lessema è dato da come quel lessema viene usato nei contesti linguistici.

**Allucinazioni semantiche** -> modelli come ChatGPT dicono cose che si avvicinano al vero ma non sono del tutto vere.

Abbiamo dei modelli computazionali oggigiorno che sono capaci di costruire significati anche di cose mai viste prima, questi modelli recenti sono non compostionali.

Tutte le proprietà semantiche codificate nei vettori sono i modi in cui le parole sono utilizzate nei diversi contesti.

### **Distributional Semantics Models**

I modelli semantici distribuzionali sono modelli computazionali che costruiscono le rappresentazioni vettoriali delle espressioni linguistiche codificando aspetti del significato.

Un DSM è caratterizzato da un **insieme di parole target ( $T$ )** ovvero il vocabolario del modello ed un **insieme di contesti linguistici ( $C$ )**. Quello che fa il modello è **assegnare a ciascun elemento target  $T$ , un vettore distribuzionale  $n$ -dimensionale, codificando il valore con le sue co-occorrenze nei contesti  $C$ .**

Si tratta di una procedura in tre fasi:

- Estrarre dai corpora le co-occorrenze degli elementi lessicali dai contesti linguistici (generalmente vengono usati i training corpus)
- Rappresentare gli elementi lessicali in maniera geometrica mediante vettori distribuzionali costruiti a partire da (in funzione delle) loro statistiche di co-occorrenza.
- Misurare la similarità semantica tramite la **distributional vector similarity**, calcolando quindi la distanza tra vettori distribuzionali.

Esistono tanti tipi di DSMs: word space models, semantic space models, (semantic/distributional) vector space models, geometrical (semantic) models, context-theoretic semantic models, statistical semantic models or corpus-based semantic models, etc.

### **Corpus selection**

Essendo questi modelli data-driven ovvero modelli basati sui dati è chiaro che il tipo di rappresentazione che i modelli sviluppano dipende dal corpus. Di conseguenza **la scelta del training corpus è fondamentale**. Se uso un corpus meno formale i modelli apprenderanno un significato linguistico meno formale, dipende tutto dal tipo di testo che utilizziamo nell'addestramento.

Oggi tendiamo a inserire qualsiasi informazione dentro un modello, questo procedimento può portare a risultati erronei: dando in input info sbagliate, il modello impara su dati inesatti e produce poi risultati sbagliati.

L'altro elemento fondamentale è la **dimensione dei corpora**.

Solitamente più è grande il corpus di addestramento e migliore è il modello.

### **Corpus Processing**

La prima cosa da fare è quella **di selezionare il corpus e tokenizzarlo**.

Fino a poco tempo fa il testo di partenza veniva anche pre-processato mediante: POS-tagging, lemmatizzazione, dependency parsing.

Oggi non vengono più eseguite a mano dal linguista, ma comunque sia avvengono automaticamente.

Il compromesso tra un'analisi linguistica più approfondita e la necessità di risorse specifiche per la lingua può introdurre errori in ciascuna fase dell'analisi e richiedere la regolazione di più parametri. La strategia di elaborazione del corpus influisce sulla selezione dei target e dei contesti.

### **The development of distribuzionale semantics**

La semantica distribuzionale nasce come modello d'impatto negli anni 90.

Possiamo individuare **tre generazioni**:

- 1- **Prima generazione - Count models:** si chiamano così perché il modello impara a costruire i vettori distribuzionali contando quante volte una parola target sta in un certo contesto. Questi conteggi vengono poi rappresentati con una matrice di co-occorrenza.  
Questo modello è stato dominante dagli anni 90 fino al 2013.
- 2- **Seconda generazione:** nel 2013 nascono i primi modelli basati su reti neurali, in particolare nasce **Word2Vec**.  
Questi **modelli** vengono chiamati **predittivi**, i modelli imparano a predire dato un contesto ovvero vengono addestrati con un Language Model di tipo neurale. Qui parliamo di **embedding**.  
Questi elementi neurali tuttavia sono molto piccoli, hanno pochi elementi.
- 3- **Terza generazione:** nel 2019 compare **BERT**, in questa fase, si tratta sempre di reti neurali ma hanno una caratteristica fondamentale: se prima i vettori erano composti di parole tipo ovvero di lessemi, con i modelli di terza generazione i modelli imparano una rappresentazione semantica sotto forma di vettore per ogni occorrenza di parola in un contesto. Questi modelli, quindi, sono modelli neurali profondi di predizione che assegnano a ciascun token di parola in un contesto di frase specifico una rappresentazione distribuzionale unica.

I vettori di I e II generazione sostanzialmente imparavano un vettore e basta valido per tutti i contesti del lessema, qui invece abbiamo un vettore diverso per ogni contesto.

Model name	References
Hyperspace Analogue of Language (HAL)	Lund and Burgess (1996)
Latent Semantic Analysis (LSA)	Landauer and Dumais (1997)
Random Indexing (RI)	Kanerva et al. (2000)
Dependency Vectors (DV)	Padó and Lapata (2007)
Topic Models	Griffiths et al. (2007)
Random Indexing with permutations	Sahlgren et al. (2008)
Distributional Memory (DM)	Baroni and Lenci (2010)
word2vec (CBOW, Skip-Gram)	Mikolov et al. (2013a, 2013b)
Global Vectors (GloVe)	Pennington et al. (2014)
FastText	Bojanowski et al. (2017)
Bidirectional Encoder Representations from Transformers (BERT)	Devlin et al. (2019)

### **Classical DSMs (I gen)**

Questi sono ancora utili per rappresentare e capire il meccanismo dei vettori distribuzionali.

Si tratta di informazione distribuzionale rappresentata come matrici di co-occorrenza.

Questo tipo di approccio nasce nell'ambito dell'information retrieval.

- 1- Le co-occorrenze tra elementi lessicali target e contesti linguistici sono estratte da un corpus testuale. L'insieme dei target contiene tipi lessicali selezionati tra i lessemi la cui frequenza è al di

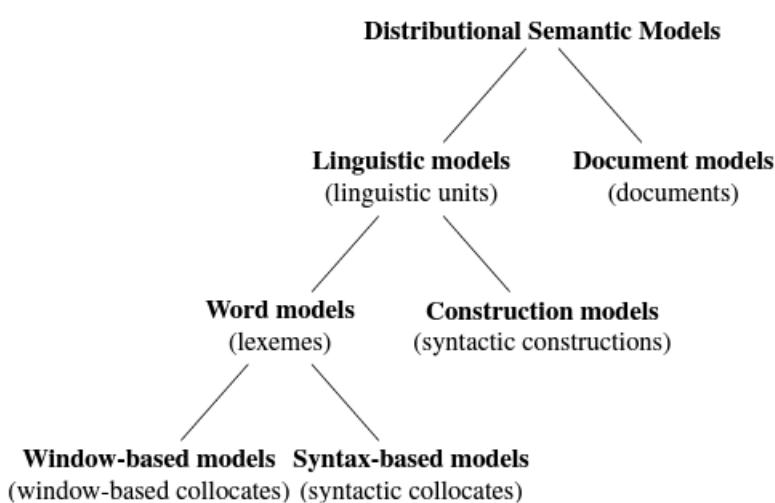
sopra di una soglia determinata empiricamente. I target possono essere lessemi morfologicamente flessi, lemmi o lemmi disambiguati dal punto di vista del POS (part of speech).

- 2- La distribuzione degli elementi lessicali è rappresentata con una matrice di co-occorrenza, le cui righe corrispondono agli elementi lessicali target, le colonne ai contesti e le entrate alla loro frequenza di co-occorrenza o a una funzione di essa per ponderare la rilevanza del contesto.
- 3- La matrice di co-occorrenza è (opzionalmente) mappata su una nuova matrice ridotta di dimensioni latenti.
- 4- La similarità semantica tra lessemi è misurata con la similarità dei loro vettori di riga nella matrice (ridotta).

## Cosa intendiamo per contesto linguistico

### Contesto come lessemi

Nella semantica distribuzionale esistono varie forme di contesti linguistici, il più famoso è quello di rappresentare contesti linguistici con le parole.



Molto importante qui è il **concetto di collocazione**.

I collocati sono le parole contesto di una particolare parola, che ho trovato insieme ad un'altra parola.

Es. *run e panino* non posso trovarle accanto.

**Vocabolario del modello** = insieme delle **parole target** ovvero quelle di cui voglio costruire rappresentazioni distribuzionali e le **parole contesti** che uso per costruire le rappresentazioni.

**Spesso insieme delle parole target e contesto coincidono.** L'insieme delle parole contesto C solitamente viene creato con le n parole più frequenti. Sono possibili però altri criteri.

Il collocato di una parola target è una parola che co-occorre con la parola target.

I modelli differiscono per il tipo di **relazioni di co-occorrenza**:

- **Co-occorrenze Window based:** la parola contesto si trova entro una certa distanza lineare dal target. Window perché viene utilizzata una finestra di contesto di ampiezza n arbitraria determinata a priori.
- **Co-occorrenze sintattiche:** la parola contesto è collegata al target da relazioni sintagmatiche (indipendentemente dalla loro distanza lineare).

La prima tipologia di rappresentazione è migliore della seconda.

Nel secondo caso, infatti, abbiamo bisogno che il nostro corpus sia annotato sintatticamente ed è anche molto più costoso. Anche ChatGPT usa una finestra di contesto molto grande di tipo lineare.

N.B. Contesto = parole che co-occorrono con le target.

## **Lezione 12 – 29/04**

### Contesto come documenti

N.B. Contesto = insieme di documenti con cui una parola ricorre.

La distribuzione di un target  $t$  può essere rappresentata con le sue occorrenze nei documenti.

Per rappresentare il contesto di una parola possiamo intendere l'insieme di documenti in cui la parola ricorre. Due parole saranno più simili tra di loro quanto queste tenderanno a ricorrere in documenti simili.

I documenti sono unità testuali come capitoli di libri, pagine web, articoli di giornale, paragrafi, frasi, turni di dialogo, o semplicemente porzioni di testo di qualsiasi dimensione fissa

Questo tipo di modello fu dominante nella ricerca cognitiva fino agli anni 90.

Firth\_1957. [You shall **know** a word] by the company it keeps!

context types	co-occurrences
window-based collocate	$\langle \text{know}, \text{word} \rangle$
dependency-filtered syntactic collocate	$\langle \text{know}, \text{word} \rangle$
dependency-typed syntactic collocate	$\langle \text{know}, \langle \text{obj}, \text{word} \rangle \rangle$
document	$\langle \text{know}, \text{Firth\_1957} \rangle$

Es. parola target “know” + il suo contesto “word”

2° caso: parole legate da una relazione di tipo sintagmatico

3° caso: contesto + informativo, perché tiene traccia anche del fatto di una dipendenza di tipo oggetto (oltre che sintattica)

4° caso: ho identificato “know” in un determinato documento

### Co-occurrences counts

Si parte da un insieme  $T$  di parole target ed un insieme  $C$  di contesti.

La prima cosa che viene fatta è l'estrazione della distribuzione delle co-occorrenze.

$t$	$c$	$F(t, c)$	$t$	$c$	$F(t, c)$	
bike	buy	9	dog	eat	9	Ciascuna di queste coppie è una co-occorrenza individuale.
bike	get	12	dog	get	10	L'idea è che le parole differenziano per quante volte co-occorrono nel contesto.
bike	park	8	dog	live	7	Questo significa andare a contare quante volte una determinata coppia distribuzionale ricorre all'interno del contesto per ottenere così la <b>frequenza di co-occorrenza</b> e la <b>distribuzione delle parole target rispetto al contesto</b> .
bike	ride	6	dog	tell	1	
car	buy	13	lion	bite	6	
car	drive	8	lion	eat	1	
car	get	15	lion	get	8	
car	park	5	lion	live	3	

### From distributions to matrices

Questo metodo in realtà non è una soluzione ottimale per ragioni che vedremo.

Quello che viene fatto è trasformare le frequenze di co-occorrenza in un peso attraverso specifiche **funzioni di pesatura** che stimano l'importanza di ogni parola contesto  $c$  nel determinare la parola target  $t$ .

Abbiamo quindi una frequenza di peso che prende in input le occorrenze e le trasforma in altri valori numerici più utili matematicamente per valutare il peso.

Dopo aver fatto questo viene creata la **matrice di co-occorrenza con le coppie distribuzionalmente pesate**: la matrice avrà tante righe quante le parole target e tante colonne quanto i contesti.

La matrice è etichettata, ad ogni riga corrisponde una certa parola target.

$$\begin{matrix} & c_1 & c_2 & \dots & c_n \\ t_1 & \left( \begin{array}{cccc} w_{t_1,c_1} & w_{t_1,c_2} & \vdots & w_{t_1,c_n} \\ w_{t_2,c_1} & w_{t_2,c_2} & \vdots & w_{t_2,c_n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{t_m,c_1} & w_{t_m,c_2} & \vdots & w_{t_m,c_n} \end{array} \right) \\ t_2 \\ \vdots \\ t_m \end{matrix}$$

In questo modo noi andiamo a creare uno spazio vettoriale dato dalla matrice di co-occorrenza.

### Weighting contexts

Il modo più semplice per la funzione di pesatura è usare le **raw frequencies** delle coppie di elementi:

$$\begin{matrix} & bite & buy & drive & eat & get & live & park & ride & tell \\ bike & 0 & 9 & 0 & 0 & 12 & 0 & 8 & 6 & 0 \\ car & 0 & 13 & 8 & 0 & 15 & 0 & 5 & 0 & 0 \\ dog & 0 & 0 & 0 & 9 & 10 & 7 & 0 & 0 & 1 \\ lion & 6 & 0 & 0 & 1 & 8 & 3 & 0 & 0 & 0 \end{matrix}$$

Ciascuna riga = vettore distribuzionale

Nell'immagine vediamo come *get* è il contesto più pesato per ogni parola.

Chiaramente per *car* un contesto vero non è *get* ma *park* o *drive* che pesano meno di *get*.

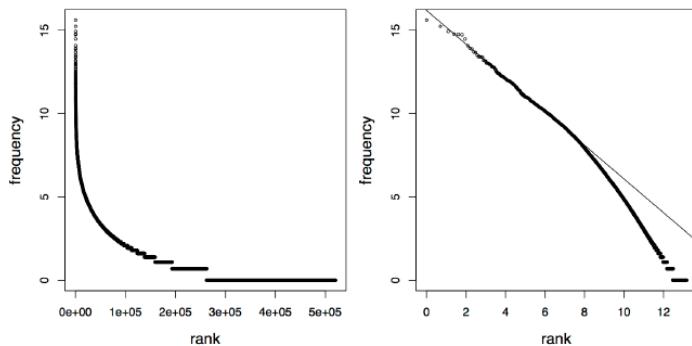
A noi, quindi, interessa il contesto più informativo.

Più una parola tende a ricorrere con altre parole rispetto ad altre, più è informativa per quel determinato contesto. Se invece una parola come *get* è associabile a qualsiasi cosa allora anche se ricorre maggiormente è molto meno informativa.

Ritorna in gioco la **legge di Zipf**: nei testi ci sono parole che tendono a ricorrere spesso.

$$F(w) = \frac{C}{r(w)^a} \quad \log F(w) = \log C - a \log r(w)$$

Ci sono alcune parole con frequenza altissima che tendono a ricorrere con tutto



Lunga coda di parole con bassissima frequenza

### Gli effetti della legge di Zipf per DSMs

Problema **Data sparseness**: poiché le distribuzioni di frequenza delle parole sono caratterizzate da un grande numero di parole con frequenze molto basse, le evidenze distribuzionali sugli elementi lessicali sono sparse.

Quindi: le **parole + informative** hanno la **frequenza + bassa**

Quindi: le frequenze non ci danno una stima affidabile

Le frequenze grezze **non forniscono una stima affidabile** dell'importanza dei fatti distribuzionali.

Es.  $f(\text{dog}, \text{get})$  è molto più alto di  $f(\text{dog}, \text{bark})$ , ma “*bark*” è sicuramente un contesto più importante di “*get*” per caratterizzare il significato di cane.

La distribuzione non uniforme delle parole nei corpora introduce un **bias di frequenza**: le parole con frequenza simile finiscono per avere vettori distribuzionali più simili di quanto non siano in realtà.

**Frequenze bias:** condizionate dal fatto che 2 parole con significati molto lontani ma con frequenze simili rischiano di essere erroneamente considerate simili.

### Misure di associazione (association scores)

Misurano la forza dell'associazione tra il target e il contesto come funzione non solo della loro frequenza congiunta, ma anche della loro distribuzione generale nel corpus di addestramento.

I DSM a matrice che rappresentano i contesti dei lessemi target con i loro collocati utilizzano tipicamente punteggi di associazione derivati dallo studio quantitativo delle collocazioni.

Questo calcolo viene fatto dentro la rete, nei modelli count invece si modificano le semplici frequenze.

Molto importante è il **Pointwise Mutual Information PMI**,

$$\text{PMI}_{(t,c)} = \log_2 \frac{p(t,c)}{p(t)p(c)}$$

$$p(t,c) = \frac{F(t,c)}{F(*,*)} \quad p(t) = \frac{F(t,*)}{F(*,*)} \quad p(c) = \frac{F(*,c)}{F(*,*)}$$

Idea: certe parole tendono a ricorrere più insieme piuttosto che una indipendentemente dall'altra

Se la mutua informazione è negativa, vuol dire che non ho nessuna informazione. Quindi: la MI propone alcuni

contesti e abbassa l'importanza di altri

Notazione:

$F(t,*)$  - total frequency of the distribution pairs with target  $t \in T$ :

$$F(t,*) = \sum_{j=1}^n F(t, c_j)$$

$F(*,c)$  - total frequency of the distribution pairs with context  $c \in C$ :

$$F(*,c) = \sum_{i=1}^m F(t_i, c)$$

$F(*,*)$  - total frequency of distributional pairs:

$$F(*,*) = \sum_{i=1}^m \sum_{j=1}^n F(t_i, c_j)$$

Punteggio di associazione:

	<i>bite</i>	<i>buy</i>	<i>drive</i>	<i>eat</i>	<i>get</i>	<i>live</i>	<i>park</i>	<i>ride</i>	<i>tell</i>
<i>bike</i>	0	9	0	0	12	0	8	6	0
<i>car</i>	0	13	8	0	15	0	5	0	0
<i>dog</i>	0	0	0	9	10	7	0	0	1
<i>lion</i>	6	0	0	1	8	3	0	0	0

$$\text{PMI}_{(\text{dog}, \text{eat})} = \log_2 \frac{p(\text{dog}, \text{eat})}{p(\text{dog})p(\text{eat})} = \log_2 \frac{9/121}{(27/121)(10/121)} = 2.01$$

$$\text{PMI}_{(\text{dog}, \text{get})} = \log_2 \frac{p(\text{dog}, \text{get})}{p(\text{dog})p(\text{get})} = \log_2 \frac{10/121}{(27/121)(45/121)} = -0.01$$

Osservo che il PMI di  $(\text{dog}, \text{get})$  è molto inferiore rispetto a  $(\text{dog}, \text{eat})$ , questo per capire anche il grado di informazione.

Con *get* il risultato è negativo.

Questo significa che non fornisce alcuna informazione.

Infine, sostituiamo nella matrice i valori del PMI al posto di quelli di prima ed osserviamo dei risultati più realistici.

### PPMI (Positive Pointwise Mutual Information)

$$\text{PPMI}_{\langle t,c \rangle} = \begin{cases} \text{PMI}_{\langle t,c \rangle} & \text{if } \text{PMI}_{\langle t,c \rangle} > 0 \\ 0 & \text{otherwise} \end{cases}$$

	<i>bite</i>	<i>buy</i>	<i>drive</i>	<i>eat</i>	<i>get</i>	<i>live</i>	<i>park</i>	<i>ride</i>	<i>tell</i>
<i>bike</i>	0	0.50	0	0	0	0	1.09	1.79	0
<i>car</i>	0	0.80	1.56	0	0	0	0.18	0	0
<i>dog</i>	0	0	0	2.01	0	1.65	0	0	2.16
<i>lion</i>	2.75	0	0	0	0.26	1.01	0	0	0

Vediamo però che non è del tutto preciso, guarda tell con dog.

Non è possibile che sia così alto.

Viene quindi fatta un'operazione ulteriore di raffinamento:

### Explicit vectors

Una **rappresentazione distribuzionale esplicita** è un vettore di caratteristiche n-dimensionale  $u$ , tale che ogni componente del vettore  $u$  corrisponde a un contesto linguistico distinto. I vettori esplicativi sono ad alta dimensionalità e sparsi (hanno un alto numero di valori 0).

Poiché considerano ogni contesto linguistico come una caratteristica distinta, i vettori esplicativi non tengono conto del fatto che i contesti possono a loro volta essere molto simili e fortemente correlati gli uni agli altri.

I vettori esplicativi soffrono del fatto che molte co-occorrenze possibili rimangono non osservate nei corpora, a causa della distribuzione dei dati distorta.

Ci sono delle parole che occorrono raramente ma che stanno insieme, se il corpus è piccolo può portare a risultati falsi e non reali.

Sempre, in questo modo, noi tendiamo a vedere i contesti come uno diverso dall'altro; ortogonali tra loro. Ma in realtà contesti come *bite* e *eat*, *drive* e *park* sono più simili tra loro rispetto a *park* e *eat*.

La maggioranza dei contesti ricorrono con **pochi** parole:

1. I contesti sono pochi
2. Le parole sono selettive (ricorrono solo con determinate parole)
3. Se due parole non sono insieme in un testo, non è detto che non possano stare insieme → alcuni zeri sono tali solo perché non li ho osservati, non perché non possono essere insieme due parole → quindi non potrò mai sapere a cosa è legato uno zero

Per tutti questi motivi solitamente si lavora sui **vettori impliciti**.

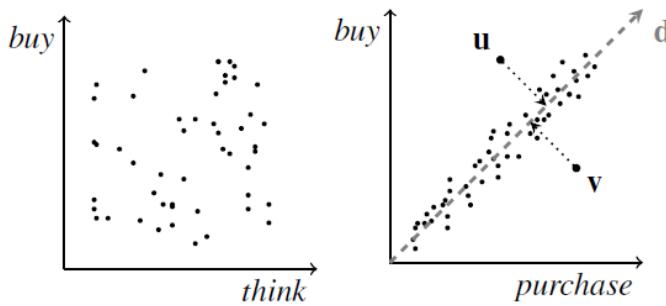
### Implicit vectors

Una rappresentazione distribuzionale implicita è un vettore di caratteristiche k-dimensionale  $u$  tale che:

1. le dimensioni corrispondono a k caratteristiche latenti estratte dalle co-occorrenze
2. il numero di caratteristiche latenti, tipicamente nell'ordine di alcune centinaia, è molto più piccolo rispetto ai n contesti linguistici in C
  - a. le componenti non corrispondono più ai singoli contesti, ma a dimensioni latenti (nascoste), quindi non sono immediatamente etichettabili
3. la maggior parte dei componenti è non nulla (il vettore è denso)

Idea: **compressione** (nel passaggio da vettori esplicativi a quelli impliciti) → estraggo le dimensioni più informative dal dato originario

### Features latenti (imparate dai modelli neurali)



Dimensioni latenti come "buy" e "think" portano a punti distribuiti in maniera abbastanza randomica

Invece, "buy" e "purchase" sono parole diverse ma simili nel contesto, quindi tenderanno a ricorrere in maniera **lineare** → i punti formano una linea → è come se esistesse una dimensione che "nascondono"

Quindi → posso ottenere una sola dimensione, come "parole più comprabili e meno comprabili" (rappresentando i dati lungo la retta **d**) → acquistabile diventa una dimensione semantica.

Se proietto i punti "u" e "v" sulla dimensione d, la distanza tra loro è molto poca (rispetto alla distanza che avevano su "buy" e "purchase"). Infatti, sulla dimensione latente delle parole vengono considerate molto più simili.

Es. posso vedere una parola tante volte con "buy" e mai con "purchase" (es. "cornetto") ma se lo metto sulla dimensione della "comprabilità", la parola "cornetto" non è molto distante da una parola come "casa".

Reti neurali → sono dei "riduttori di dimensionalità" – sistema chiamato anche "feature extraction" (operazione che ci permette di passare da vettore esplicito ad implicito), perché è come se estraessimo delle strutture semantiche "nascoste" nei dati. Questi modelli sono estremamente capaci di ridurre il rumore, smoothing data sparseness.

### **Estrazione delle features**

Quando si passa da vettori esplicativi a impliciti è come se si estraesse dalla distribuzione delle parole l'informazione più rilevante. Così si migliora la qualità degli spazi semantici. Un limite importante delle rappresentazioni implicite è la loro **ridotta interpretabilità**; è difficile, se non impossibile, comprendere la vera natura semantica delle loro dimensioni latenti.

La pratica è conosciuta anche come estrazione di caratteristiche, poiché le dimensioni dello spazio ridotto sono nuove caratteristiche estratte dai dati originali.

L'estrazione delle caratteristiche ha quattro obiettivi principali:

- i) scoprire strutture semantiche latenti nei dati distribuzionali
- ii) ridurre il rumore
- iii) attenuare la data sparseness

### **Vettori impliciti**

Rappresentare le co-occorrenze dei lessemi nei contesti linguistici:

- I vettori esplicativi codificano direttamente le co-occorrenze nei loro componenti.
- I vettori impliciti rappresentano le co-occorrenze in modo indiretto, poiché i loro componenti codificano caratteristiche latenti estratte dai dati distribuzionali.

## Lezione 13 – 30/04

### Riduzione della dimensionalità nei classici DSMs

I DSMs classici mappano la matrice di co-occorrenza su uno spazio semantico latente ridotto con una **funzione di riduzione della matrice**.

### Funzione di riduzione della matrice

La funzione di riduzione della matrice  $R : M_{T \times C} \rightarrow M'_{T \times D}$  trasforma la matrice di co-occorrenza  $M_{T \times C}$ , sparsa e ad alta dimensionalità, in una matrice densa  $M'_{T \times D}$  tale che  $D = \{d_1, \dots, d_k\}$  siano caratteristiche latenti e  $|D| \ll |C|$ .

R di solito consiste in una **fattorizzazione** (o decomposizione) della matrice di co-occorrenza nel prodotto di altre matrici.

Ci sono vari metodi di riduzione, il più popolare è il SVD, che viene dalla PCA (Principal Component Analysis), metodo statistico che individua le principali variazioni.

### SVD (Singular Value Decomposition)

- Le colonne  $d_1, \dots, d_n$  delle matrici  $U$  e  $V$  rappresentano dimensioni latenti nei dati originali, ordinate per l'importanza della varianza che rappresentano.
- I valori singolari in  $\Sigma$  esprimono l'importanza di ciascuna nuova dimensione, dove  $\sigma_i$  indica l'entità della variazione dei dati lungo la dimensione latente  $d_i$ .

$$M_{m \times n} = U_{m \times z} \Sigma_{z \times z} (V_{n \times z})^T$$

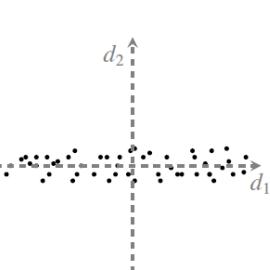
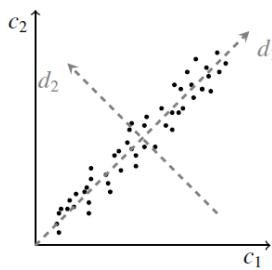
>> **SVD** è un metodo matematico che permette di individuare altre 3 matrici il cui prodotto è  $M$  (la prima moltiplicata per la seconda per la trasposta della terza)

$$M_{m \times n} = U_{m \times z} \Sigma_{z \times z} (V_{n \times z})^T$$

- 1<sup>a</sup> matrice  $U_{mxz} \rightarrow$  le righe sono  $m$  e le colonne sono delle dimensioni latenti ( $d$ )
  - Vettori ortogonali = vettori che nello spazio sono indipendenti l'uno dall'altro
  - Tutte le dimensioni  $d$  saranno indipendenti l'una dall'altra
- 2<sup>a</sup> matrice  $\Sigma_{zxz} \rightarrow$  tutti zeri tranne nella **diagonale** (matrice quadrata)
  - $\{\sigma_1, \dots, \sigma_n\}$ : numeri sulla diagonale ordinati per ordine decrescente (dal più grande al più piccolo da sx a dx) che dicono quanto sono importanti le dimensioni per caratterizzare i dati di partenza
- 3<sup>a</sup> matrice  $(V_{nxz})^T \rightarrow$  righe:  $d$ , colonne: parole target (?)

spiegare varianza = spiegare come variano i dati

## SVD come una rotazione



d1 = asse di variazione principale

SVD = trova nuove dimensioni ortogonali (d1 e d2) e fa un'operazione di **rotazione**, raggruppando i dati originali rispetto alle nuove coordinate (date da d1 e d2)

d1 = asse di variazione principale, perché tutti i dati si distribuiscono lungo d1

varianza dati = spiegata quasi esclusivamente da d1

Cosa ho guadagnato? Ora so quali sono le dimensioni più importanti

Se rappresento i dati solo su d1 perdo un po' di informazioni, ma non le più importanti (perché non mi fa perdere la capacità di discriminare i dati). Quindi, invece di tenere tutte le n dimensioni, tengo solo le prime k e le altre le scarto.

## La matrice ridotta con vettori impliciti

- Eliminando tutti tranne i primi k valori singolari e scartando  $V^T$ , otteniamo una nuova matrice ridotta con dimensioni  $D = \{d_1, \dots, d_k\}$ .
    - I target sono rappresentati in uno spazio dimensionale inferiore che tiene conto della maggior parte della loro variazione originale.
- $M'_{T \times D} = U_{m \times k} \Sigma_{k \times k}$  → matrice con stesso numero di righe ma con un numero di colonne più piccolo, che corrispondono alle dimensioni più importanti.
- In questa operazione ignoro la matrice  $V^T$  ("non ci serve")
  - Le **righe** vettore di  $M'$  sono **embedding** dei target in uno spazio semantico latente.

$$\begin{array}{ccc}
 & d_1, \dots, d_k & \\
 & M'_{T \times D} & \\
 \begin{matrix} t_1 \\ \vdots \\ t_m \end{matrix} & = & \begin{matrix} t_1 \\ \vdots \\ t_m \end{matrix} \quad U \quad \begin{matrix} \sigma_1 \\ \ddots \\ \Sigma \\ \sigma_k \end{matrix} \\
 & d_1, \dots, d_k & \\
 & k &
 \end{array}$$

SVD: garantisce che  $M'$  è la migliore matrice (compressione) che posso ottenere, che mi fa perdere meno informazione possibile (è il meglio che posso fare un uno spazio ridotto)

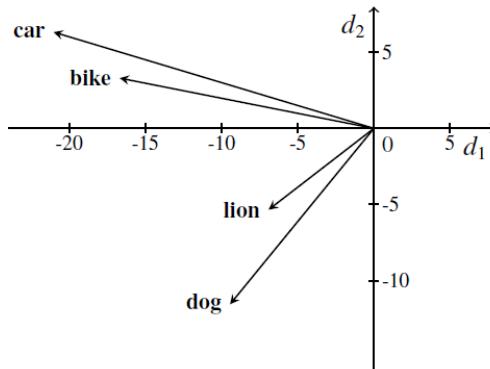
## SVD: un esempio

$$M = \begin{pmatrix} & bite & buy & drive & eat & get & live & park & ride & tell \\ bike & 0 & 9 & 0 & 0 & 12 & 0 & 8 & 6 & 0 \\ car & 0 & 13 & 8 & 0 & 15 & 0 & 5 & 0 & 0 \\ dog & 0 & 0 & 0 & 9 & 10 & 7 & 0 & 0 & 1 \\ lion & 6 & 0 & 0 & 1 & 8 & 3 & 0 & 0 & 0 \end{pmatrix}$$

Partiamo da una matrice esplicita, che con SVD diventa 3 matrici. Voglio tenere solo le prime due dimensioni, quindi rimuovo le altre (avrò solo due colonne, d1 e d2) → troncamento della matrice

Se faccio il prodotto tra U e  $\Sigma$  ottengo  $M'$ , la matrice migliore ottenibile in uno spazio ridotto

$$M' = \begin{pmatrix} & d_1 & d_2 \\ bike & -16.68 & 3.30 \\ car & -21.02 & 4.32 \\ dog & -9.44 & -11.55 \\ lion & -6.91 & -5.34 \end{pmatrix} = U \begin{pmatrix} & d_1 & d_2 \\ bike & -0.57 & 0.24 \\ car & -0.72 & 0.31 \\ dog & -0.32 & -0.83 \\ lion & -0.23 & -0.39 \end{pmatrix} \times \Sigma \begin{pmatrix} 29.28 & 0 \\ 0 & 13.83 \end{pmatrix}$$



↳ Se rappresento le parole nel nuovo spazio ottengo i seguenti vettori (cerco di distillare dai dati delle dimensioni di variazione principali, cioè dimensioni che discriminano meglio i dati)  
scelta numero di dimensioni latenti = iperparametro (è una scelta che faccio io a priori)

### Scegliere le dimensioni latenti

La scelta delle dimensioni latenti ridotte k è normalmente un compromesso tra prestazioni ed efficienza:

**N.B. Più piccolo è k, più efficiente** dal punto di vista computazionale il modello distribuzionale risultante, ma **più grande è k**, più il modello distribuzionale ridotto **assomiglierà all'originale**.

- Landauer & Dumais (1997) sostengono che le prestazioni ottimali si ottengono utilizzando circa 300 dimensioni latenti.
- Lapesa & Evert (2014) sostengono che la dimensione ottimale dei vettori ridotti dipende dal compito semantico e dal dataset, con una configurazione generale raccomandata di 500 dimensioni di caratteristiche latenti.
  - Lapesa, G., e Evert, S. (2014). "A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection". TACL, 2, 531–545.

**Misura similarità semantica = applicabile a qualsiasi tipo di vettori**

Presi due vettori, restituisce un numero reale che indica quanto due vettori sono simili tra loro (se identici è 1)

### Similarità vettoriale

Una misura di similarità vettoriale è una funzione  $S : T \times T \rightarrow \mathbb{R}$  tale che, per ogni coppia di lessimi obiettivo  $u$  e  $v$ ,  $S(u, v)$  è proporzionale al grado di similarità tra  $u$  e  $v$ .  $S$  obbedisce alle seguenti condizioni:

- S1.  $S(u, v) \leq 1$ ;

$$U = \begin{pmatrix} & d_1 & d_2 & d_3 & d_4 \\ bike & -0.57 & 0.24 & -0.78 & -0.06 \\ car & -0.72 & 0.31 & 0.62 & -0.05 \\ dog & -0.32 & -0.83 & 0.01 & -0.45 \\ lion & -0.23 & -0.39 & -0.01 & 0.89 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 29.28 & 0 & 0 & 0 \\ 0 & 13.83 & 0 & 0 \\ 0 & 0 & 7.61 & 0 \\ 0 & 0 & 0 & 6.51 \end{pmatrix}$$

$$V^T = \begin{pmatrix} & bite & buy & drive & eat & get & live & park & ride & tell \\ d_1 & -0.05 & -0.49 & -0.20 & -0.11 & -0.78 & -0.10 & -0.28 & -0.12 & -0.01 \\ d_2 & -0.17 & 0.45 & 0.18 & -0.57 & -0.28 & -0.50 & 0.25 & 0.10 & -0.06 \\ d_3 & -0.01 & 0.13 & 0.65 & 0.01 & -0.01 & 0.01 & -0.41 & -0.62 & 0.00 \\ d_4 & 0.82 & -0.17 & -0.06 & -0.48 & 0.20 & -0.07 & -0.11 & -0.05 & -0.07 \end{pmatrix}$$

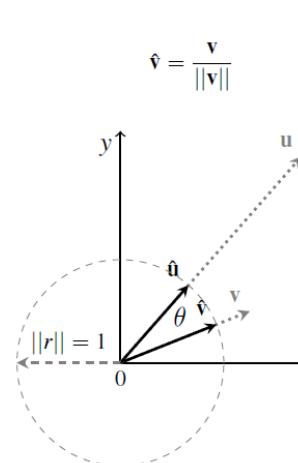
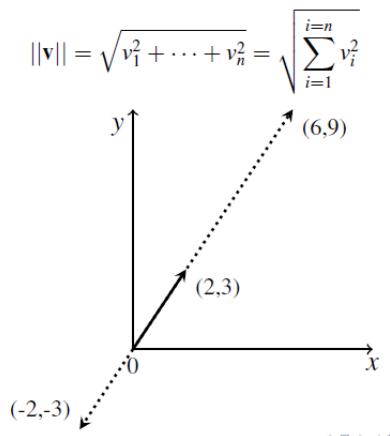
- S2.  $S(u, v) = 1$  se e solo se  $u$  e  $v$  sono identici;
- S3.  $S(u, v) = S(v, u)$  (simmetria).

## Vicini più prossimi (nearest neighbors)

I vicini più prossimi di un obiettivo  $t$  sono gli elementi lessicali con il punteggio di similarità più alto con  $t$ .

## Lunghezza di un vettore

La lunghezza (o norma)  $\|v\|$  di un vettore  $v$  è la lunghezza della sua freccia (**non** è il numero di componenti del vettore)



Ricorda il teorema di Pitagora – generalizzazione in uno spazio a  $n$  dimensioni → vettore = ipotenusa di un triangolo rettangolo

I due vettori sono multipli tra loro, hanno la stessa direzione (primo vettore moltiplicato per uno scalare, ottengo il secondo)

Esempio: "re" è più frequente di "monarca" e anche se le due parole compaiono in contesti molto simili, la freccia di "re" sarà molto più lunga, perché ha valori di frequenza più alti. Quindi, può capitare che nello spazio parole simili come queste siano lontane tra loro, perché condizionate dalla frequenza.

Se invece proietto i punti su una **circonferenza comune** conta solo la loro **direzione** in cui puntano (mentre la lunghezza ora è la stessa). Questa è un'operazione di **normalizzazione** del vettore, cioè, divido il vettore per la sua lunghezza. Quindi tutti i vettori avranno lunghezza 1 (circonferenza di raggio 1) e ora conta solo la direzione.

Se ora conta solo la direzione, posso dire che 2 vettori sono tanto più simili quanto più è simile la direzione in cui puntano, quindi quanto più è piccolo l'angolo  $\theta$  (teta).

La misura del coseno dell'angolo teta, calcolato da **dot product** (prodotto scalare) dei due vettori e il prodotto delle loro dimensioni (delle norme).

## Normalizzazione vettoriale in R

- Computing vector norm
 

```
> c(2,5,7) -> a
> sqrt(sum(a^2)) #computing the vector norm
> 8.831761
```
- Normalizing a vector
 

```
> a/8.831761
> 0.2264554 0.5661385 0.7925939 #normalized vector
```

## Similarità del coseno

**Dot product** = somma delle dimensioni delle diverse componenti  
(somma dei prodotti)

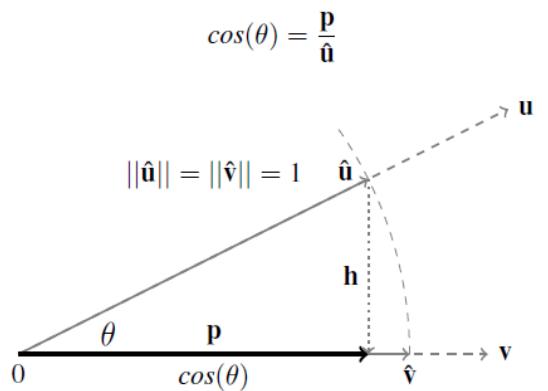
$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + \cdots + u_n v_n = \sum_{i=1}^{i=n} u_i v_i$$

## Coseno

$$\text{sim}_{\cos}(\mathbf{u}, \mathbf{v}) = \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^n u_i v_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}}$$

Se  $\mathbf{u}$  e  $\mathbf{v}$  sono identici  $\rightarrow \cos(0) = 1$

Più si allontanano, più il coseno diminuisce  $\rightarrow \cos(90) = 0 \rightarrow$  vettori ortogonali



## Coseno in R

- Computing the dot product

```
> c(2,5,7) -> a
> c(3,0,9) -> b
> sum(a * b) #dot product
> 69
```

- Computing the cosine

```
> sum(a * b)/((sqrt(sum(a^2))) * (sqrt(sum(b^2))))
> 0.8235321 #cosine similarity between the two vectors
```

Combinazioni neuronali = calcoli di dot product

Parole = non tutte sono simili nella stessa maniera

	bike	car	dog	lion	
bicycle	0.80	van	0.73	cat	0.93
ride_V	0.74	motorcycle	0.70	puppy	0.81
motorbike	0.71	cheap	0.70	kennel	0.75
biker	0.69	motorbike	0.70	groom	0.71
ride_N	0.69	driver	0.69	breeder	0.71
rider	0.68	drive	0.67	bark	0.70
try	0.66	automobile	0.64	rabbit	0.69
one	0.66	hire	0.64	chase	0.69
ready	0.65	vehicle	0.64	kitten	0.67
horse	0.65	motorist	0.62	vet	0.67
				beast	0.52

## Lezione 14 – 6/05

Oggi vediamo come i modelli neurali rappresentano la generazione più recente di semantica distribuzionale.

Iniziamo quindi a vedere come funzionano le reti neurali e come queste possono essere utilizzate per imparare la semantica.

## Modelli neurali per la Semantica Distribuzionale

Le reti neurali hanno la caratteristica che l'informazione e i processi sono gestiti da un'unica rete di unità computazionali che lavorano in parallelo, simile al modo in cui lavorano i neuroni umani (ispirati alla fisiologia del neurone).

Reti neurali artificiali, **parallel distributed processing** (PDP)  $\rightarrow$  distribuzione parallela dei vettori, **connessismo** (termine del passato per indicare lo studio dei modelli neurali), **deep learning** (reti profonde, a molti livelli)

I metodi di Deep learning sono usati per costruire rappresentazioni distribuzionali

Questo tipo di rappresentazioni sono non simboliche perché nei neuroni non ci sono tracce di simboli e i neuroni e le computazioni neurali sono dei processi dove l'informazione si basa su vettore.

## Computazione neurale

Neurone = la più piccola unità computazionale

Trasmette informazioni ad altri neuroni come una funzione della somma dei segnali che riceve in input  
(manipola segnali per manipolare informazioni)

Apprendere cambia la forza delle connessioni tra i neuroni

La rappresentazione della conoscenza e il trattamento è determinato dall'attività distribuita di neuroni interconnessi

Modulazione segnale = amplifica o inibisce (riduce) l'intensità del segnale per regolare (modulare)

l'importanza del segnale

### **Artificial Neural Networks (reti neurali artificiali)**

Alla base della rete neurale c'è l'ispirazione al cervello umano.

Il neurone lo possiamo vedere come **l'unità minore di computazione** che sta nel nostro cervello, esso manipola dei segnali elettrochimici utilizzati per manipolare l'informazione.

Il neurone è un'unità connessa ad altri neuroni, ecco perché connessionismo.

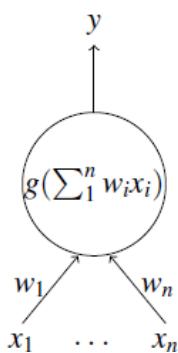
Il neurone riceve l'informazione attraverso segnali elettrici che vengono poi combinati perché provenienti da più neuroni. Poi trasmette un altro segnale ad altri neuroni attraverso l'assone. Riceve input da collegamenti con altre unità pesati, che corrispondono alle connessioni sinaptiche nel cervello.

Un peso è un **meccanismo su base elettrochimica** che modula il segnale nel senso che lo potenzia o lo inibisce, tali operazioni servono al neurone per modulare l'importanza dei segnali provenienti da altri neuroni. Se un segnale viene amplificato ci sarà una maggiore importanza relativa all'attività derivante da quel segnale.

Quando invece parliamo di **reti neurali artificiali** parliamo di modelli computazionali, quindi **esclusivamente matematici**, che semplicemente sono ispirate alla fisiologia del neurone prima descritto.

Il neurone matematico è qualcosa che riceve un input da altri neuroni (cioè, dei valori numerici reali) ciascuna di queste connessioni ha un peso ovvero un numero che serve per quantificare l'importanza del pezzo di input ricevuto.

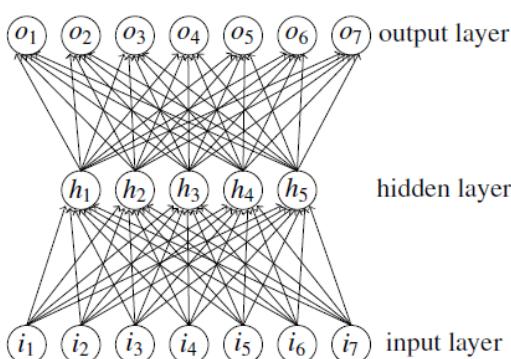
Il neurone integra l'informazione e fa una sommatoria, si tratta di un'operazione lineare.



A seconda della distribuzione dei pesi viene modulato l'input.

Dentro al neurone vi è una funzione di attivazione  $g$  che integra i segnali ricevuti e provoca in output un segnale  $y$ .

La funzione più semplice è quella di **identità**: se ricevo un input di intensità  $x$ , restituisco un output di intensità  $x$ .



Dobbiamo vedere i neuroni come se fossero parte di una rete (neuroni interconnessi). Tipicamente i neuroni in una rete artificiale sono assemblati in livelli (non necessariamente nello stesso numero per ogni livello). I livelli devono essere al minimo 2 (quelli che codificano input e output).

Le reti sofisticate hanno tantissimi livelli, centinaia o migliaia.

Le unità del livello precedente sono connesse a ciascun neurone del livello successivo (**reti fully connected layers**).

Feed-forward neural network with one hidden layer

**Non ci sono connessioni tra i neuroni dello stesso livello.** Ciascun neurone contribuisce in maniera diversa a ciascun neurone del livello successivo in base al **peso del segnale** (ovvero l'arco di collegamento). Questi pesi sono regolati in fase di apprendimento, quando il modello è addestrato a produrre un certo output basato su un certo input.

La freccia indica la direzione dell'informazione (**feed-forward**).

Gli **hidden layers** sono gli strati intermedi. I modelli che hanno molti hidden layers sono descritti come **deep neural networks**.

La cosa fondamentale è la seguente: come nel cervello umano anche nelle reti neurali i pesi possono cambiare soprattutto nella fase di addestramento. Iniziamo con una rete dove i pesi sono assegnati quasi in maniera randomica e poi la rete impara a dosare i pesi in modo da poter svolgere il task ottenendo il risultato ottimale.

In questo modo le reti neurali sono dei **modelli di apprendimento automatico**.

Le reti neurali sono **modelli di tipo ML supervisionato**, hanno bisogno di sapere quale è la risposta giusta per essere addestrati a svolgere un determinato tipo di task.

Due componenti:

- I pesi della rete sono chiamati **parametri**. Questi vengono inizialmente (in fase di addestramento) assegnati in maniera randomica, poi vengono regolati.
- Gli altri sono **iperparametri**: insiemi a priori che escludono vari fattori, come il numero degli strati di rete e la loro dimensionalità, la funzione di attivazione, etc.

In fase di addestramento, la rete riesce ad organizzare esclusivamente i parametri. Il resto (iperparametri) è ciò che noi gli diamo a priori e determina la struttura. La struttura è indipendente dal linguaggio. Per questo, la stessa rete neurale è utilizzabile per riconoscere immagini, linguaggio, suoni.

Dal punto di vista matematico ciascuno strato della rete è un **vettore di numeri reali** la cui dimensionalità corrisponde al **numero di neuroni** nello strato.

Le connessioni tra due strati completamente connessi (fully-connected) sono rappresentate da una **matrice dei pesi W**, in modo che  $w_{i,j}$  sia il peso della connessione dall' i-esimo neurone al j-esimo neurone.

### Rete neurale, vettori e matrici

Tutto questo significa che possiamo vedere la computazione neurale come una semplice operazione di algebra lineare data dalla moltiplicazione del vettore  $m$  per la matrice  $W$ .  $\rightarrow \mathbf{y} = \mathbf{x}W$

Dove  $x$  è un vettore n-dimensionale e  $W$  una matrice dei pesi  $n \times m$ . il risultato è un vettore  $y$  m-dimensionale tale che la j-esima dimensione sia  $\sum_{i=1}^n x_i w_{i,j}$

Cioè, il **dot product**:  $\mathbf{x} \cdot \mathbf{W}_{*,j}$

Tale moltiplicazione trasforma un vettore in un altro vettore ovvero quello del livello successivo.

Il processo di computazione neurale è trasformare vettori in altri vettori attraverso il prodotto matriciale.

In pratica faccio il dot product tra vettore ed elementi delle colonne della matrice e fa la somma tra i risultati.

$$\mathbf{x} = \begin{pmatrix} 3 & 6 & 4 \end{pmatrix} \mathbf{W} = \begin{pmatrix} 0.75 & 0.45 & 0.06 & 0.66 \\ 2.40 & 2.04 & 4.20 & 0.30 \\ 0.00 & 0.12 & 1.72 & 0.24 \end{pmatrix} \mathbf{y} = \mathbf{x}W = \begin{pmatrix} 16.65 & 14.07 & 32.26 & 4.74 \end{pmatrix}$$

Matrice di pesi con: righe = vettore input, colonne = vettore liv. successivo, valori = peso che connette gli el. dei vettori

Le matrici di pesi sono quelle che vengono cambiate durante la fase di addestramento

## Single Layer Network

La rete più semplice possibile è quella che ha due livelli uno di input ed uno di output (**nessun hidden layer**). Questa rete è **feed forward** e si chiama **Perceptron**.

### **Training a single layer network**

L'addestramento avviene in maniera **supervisionata**:

- Il training set D è formato da una coppia input-output  $\langle x, t \rangle$
- Input e output sono essi stessi vettori, le cui dimensioni corrispondono all'input e all'output del singolo neurone

I **pesi sono cambiati proporzionalmente** alla quantità di errore prodotta dalla rete (vengono confrontati output prodotti e attesi)

- Idea: la rete impara **sbagliando** → modificando i pesi in modo proporzionale all'errore che commette

### **Error-driven learning**

- All'inizio, i pesi sono inizializzati randomicamente
- Ad ogni passo di apprendimento (learning step), la rete riceve un input  $x$  e genera un output  $o$  (output osservato)
- Per ogni neurone,  $o$  è confrontato con il target di output di  $x$  nel training set,  $t$
- La differenza tra  $t$  e  $o$  è usata per cambiare i valori dei pesi del neurone
  - Quindi uso l'errore (differenza) tra l'output giusto e il mio output per ricalibrare i pesi
  - In ogni iterazione modifico i pesi in modo proporzionale all'errore, attraverso la **regola delta**, con cui modifico il valore originario di un peso (più è forte l'errore più la modifica è pesante)

## Delta Rule

$\eta$  è una costante chiamata **learning rate** (tasso di apprendimento) e controlla l'effetto dei cambiamenti sui pesi:

- Più il tasso è alto, più i pesi saranno modificati
- Più il tasso è basso, meno i pesi saranno modificati (sarà una modifica granulare)
- Se  $t > o$ ,  $w_i$  è aumentato, altrimenti viene decrementato

$$\Delta w_i = \eta(t - o)x_i$$

### Più l'errore decresce, più piccolo diventa il cambiamento nei pesi

La rete smette di essere addestrata quando non è più in grado di migliorare con la modifica dei pesi (raggiunge una convergenza)

Delta rule è un caso di apprendimento di **discesa del gradiente (Gradient Descent)**.

Il gradient descent è un algoritmo che progressivamente deve portare alla configurazione di pesi che porta alla diminuzione dell'errore. Il gradiente ci dice in che direzione va l'errore (è la derivata dell'errore):

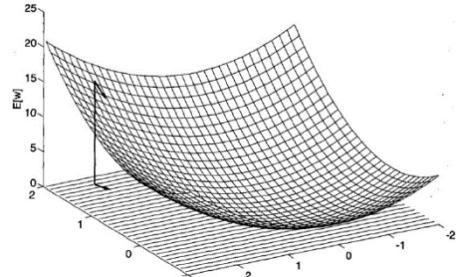
- Esploriamo lo spazio delle ipotesi dei pesi della rete cercando le loro configurazioni ottime
- Impariamo la configurazione ottima dei pesi **minimizzando una funzione di costo (loss function)**. La loss function (come il MSE) è una funzione che misura l'errore compiuto dalla rete
-

## Mean Squared Error (MSE)

$$L_{\text{MSE}}(\mathbf{o}, \mathbf{t}) = \frac{1}{n} \sum_{i=1}^n (t_i - o_i)^2$$

Il **gradiente** (= vettore di derivate parziali della funzione loss) specifica la **direzione** che produce l'aumento più ripido del vettore, mentre la discesa del gradiente cerca la configurazione ottimale della rete spostando i pesi nella direzione opposta a quella di partenza

Minimizzare l'errore = non significa che la rete non sbaglia, ma che non esiste una configurazione che le permette di fare meglio di così (per ridurre la loss)

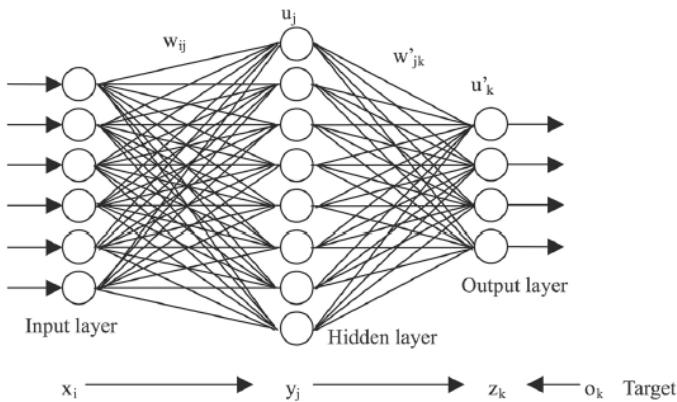


## Discesa del Gradiente Stocastica

- In Batch Gradient Descent, la regola di addestramento calcola gli aggiornamenti del peso dopo aver sommato tutti gli esempi di addestramento in D:
  - il calcolo è molto lento e costoso;
  - l'algoritmo rischia di bloccarsi in minimi locali.
- La **discesa stocastica del gradiente** approssima questa ricerca di discesa del gradiente aggiornando i pesi in modo incrementale, seguendo il calcolo dell'errore per ogni singolo esempio (o dopo un campione casuale).

## Multilayer Feedforward Network

Per avere delle reti potenti abbiamo bisogno di tanti livelli intermedi capaci di ristrutturare l'informazione.

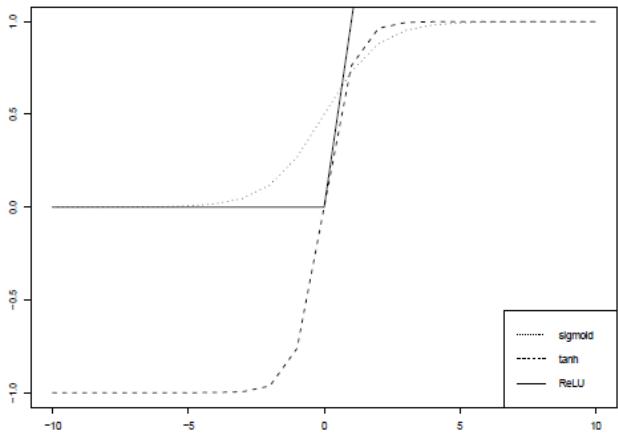


**Le reti multilivello** sono arricchite da **strati nascosti** contenenti **funzioni di attivazione non lineari** (più complesse) che vengono applicate a ogni componente vettoriale

Gli strati nascosti eseguono una trasformazione dei dati in un nuovo spazio di dimensioni latenti.

## Funzione di attivazione non lineare

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \quad \tanh(x) = \frac{\exp(2x) - 1}{\exp(2x) + 1} \quad \text{ReLU}(x) = \max(0, x)$$



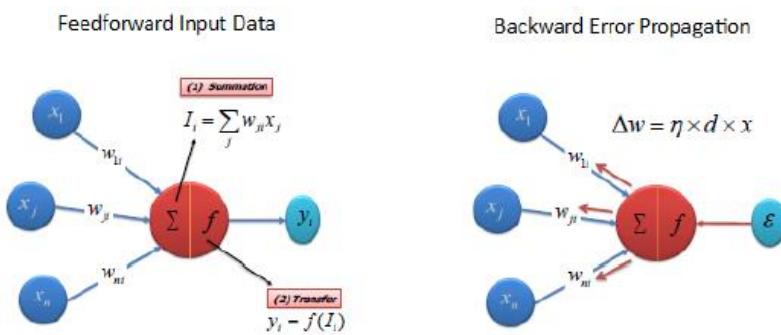
**Sigmoide** = effetto “soglia” → prima il vettore è inerte, poi dopo la soglia il vettore cresce molto velocemente

Funzione non lineare ma continua e derivabile

Prima del 1986 non c’era un modo per addestrare le reti con i livelli nascosti (non ho un qualcosa con cui confrontare il loro input)

Poi: proposta dell’algoritmo di *back-propagation*, perché propaga “all’indietro” l’errore e quindi l’errore in output viene propagato anche per modificare i pesi precedenti

## Addestramento di una MultiLayered Network



L’algoritmo di **Backpropagation** è una generalizzazione del gradiente di discesa stocastica che apprende i pesi di una rete multistrato propagando all’indietro gli errori delle unità di input

## Lezione 15 – 13/05

### Neural Networks and Linguistic Analysis

Le reti neurali sono modelli di apprendimento generici. Dato un input vengono allenate per produrre un certo tipo di output (e soddisfare un certo task).

Le reti neurali (con i metodi odierni) nascono dopo l’invenzione dell’algoritmo di back propagation intorno agli anni ’90.

Le prime ricerche di reti neurali applicate al linguaggio riguardano le scienze cognitive (modelli molto piccoli). Il primo lavoro risale al 1986, con una rete neurale che era in grado di apprendere la morfologia dell’inglese e in particolar modo il past tense: la regola è prendere la radice e aggiungere *ed* e inoltre tenere in considerazione le forme irregolari di variation della radice. L’idea era quella che la morfologia regolare fosse una regola fissa (aggiungere *ed* in fondo) e poi gli altri verbi e forme flesse memorizzabili dal modello.

**Le reti neurali non possiedono regole interne.** Imparano a partire dall’input e sottoposte a vari output corretti imparano progressivamente come rispondere a certi impulsi nella maniera corretta.

Nel caso dell’inglese, la rete neurale esposta progressivamente a coppie di verbi, dopo una serie di iterazioni progressive imparava come generalizzare la coniugazione dei verbi, oltre a quelli posti in fase di apprendimento.

Questo ebbe un input importante nelle scienze cognitive, rendendo possibile la rimozione di regole. Dal dibattito sembrava che le reti neurali potessero essere un’alternativa ai modelli simbolici presenti all’epoca.

Nel 1990 ci fu un primo importante lavoro legato alla sintassi: una rete ricorrente che aveva come obiettivo la capacità di apprendere sequenze di elementi.

Ciò che emerge da questo primo periodo è la possibilità di utilizzo di reti neurali nell'ambito del linguaggio. Dopo questo periodo di “cognitive-science” è comparsa la nuova generazione di reti neurali (quelle del deep learning dei 2000’).

### **Cosa cambia dalla generazione anni '90?**

2010 come anno importante. Ci sono nuovi tipi di reti neurali, che hanno avuto una prima applicazione nell'ambito visuale (image recognition) (e.g. Convolutional Networks, Long Short-Ter Memory, Transformers, etc.).

Iniziano ad esserci esperienze per allenare le reti neurali (reti più grandi e complesse), si va a risolvere in parte il grande **problema del “collo di bottiglia”**.

Reti neurali addestrate su grandi quantità di dati. Ovviamente tutto ciò ha richiesto software e hardware più potenti.

Grande svolta in questo campo, le GPU che hanno potenze di elaborazione talmente elevate da essere impattanti soprattutto nei calcoli matriciali (quindi in parallelo).

### **Neural Language Models (NLMs) and word embedding**

A partire dal 2010 hanno cominciato ad apparire le prime applicazioni di reti neurali per risolvere problematiche inerenti all’analisi del linguaggio: i **Neural Language Models**. Questi danno avvio alla **generazione dei Predict Language Models**, modelli addestrati a **task di predizione**, ovvero un task che data una certa sequenza di parole n1, ..., nn (il **contesto**), determina che la rete deve essere in grado di predire la parola successiva a questa sequenza (il **target**). I Neural Language Models **assegnano quindi una probabilità** ad un certo elemento di poter comparire in una data posizione.

Quello che fa un NLM è imparare a predire una o più parole dato un certo tipo di contesto.

#### **Come fa a fare tutto ciò?**

Apprendimento supervisionato. Il NLM genera all'inizio un output casuale a cui viene dato un giudizio umano; sulla base di questo ricalcola i pesi e si migliora nel generale la risposta. Questo tipo di apprendimento si chiama **self-supervised learning**, poiché non necessitiamo di testo annotato, ma tramite il contesto il NLM genera la risposta che viene corretta progressivamente.

Questo tipo di modello si avvicina molto a ciò che il nostro cervello fa già in automatico: la predizione. Quando viene pronunciata una certa frase, anche se non viene completata in automatico il nostro cervello predice la parola successiva.

Imparando a predire, la rete **impara implicitamente** i vincoli linguistici determinanti per una certa sequenza di elementi. Alla frase *La macchina è rimasta senza benzina*, il NLM avrà una certa conoscenza del mondo, una serie di proprietà morfo-sintattiche e semantiche che riguardano una determinata lingua.

L’idea è che imparando a predire una parola dato un contesto, la rete mantiene una serie di proprietà semantiche relative alla lingua. La rete vedrà che ci sono parole che appaiono in contesti simili e le “raggruppa” tramite rappresentazioni interne che tengono traccia delle posizioni delle parole (**vettori semanticici**). Quindi parole che tendono ad avere contesti simili avranno delle rappresentazioni interne della rete simili (**semantica distribuzionale**).

#### Questa è l’essenza dei modelli predict.

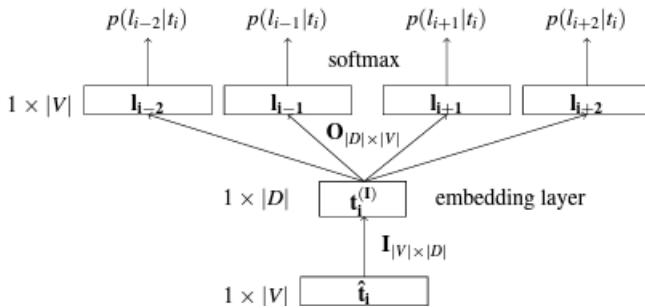
Prima rete neurale che esplicitamente veniva utilizzata per imparare le rappresentazioni distribuzionali, lo **Skip-gram**.

## Skip-Gram

Attualmente vengono creati direttamente degli embedding, cioè dei vettori numerici a basse dimensioni. Questo poiché le reti riducono in automatico le dimensioni.

Word2Vec prima rete neurale che si basava su Skip-gram.

Skip-gram (SG) learns to predict the surrounding context words  $l_{i-n} \dots l_{i-1}$ ,  $l_{i+1} \dots l_{i+n}$  given the target  $t_i$



in grado di predire il contesto (ovvero le parole più probabili) che precedono e succedono la parola target.

La finestra di predizione può essere più larga o più stretta, di solito sono 2 parole per lato.

Il vocabolario dello SG è l'insieme delle parole di contesto, **coincidono completamente**.

Lo strato di input consiste in **vettori one-hot**, cioè vettori di dimensione dell'intero vocabolario. I vettori sono ortogonali tra loro (facendo il coseno tra loro verrà 0).

Ciascuno dei vettori di input viene combinato con una matrice di pesi (embedding layer). La dimensione è qui più piccola, quindi dall'input iniziale otterrò un input ridimensionato.

All'inizio i pesi della matrice sono random, grazie alla fase di addestramento progressivamente questi pesi verranno modificati e quindi resi sempre più precisi.

Fatto tutto il processo, noi vogliamo che gli embedding dei pesi siano rappresentativi della semantica distribuzionale (che quindi grazie a questi, parole come *mela* e *pera* siano percepite come molto più simili rispetto a *mela*, *auto*).

## Skip-gram computation: an example

Assumiamo che la nostra matrice di input abbia dimensione  $I_{4x3}$ : i vettori riga sono embedding di input a 3 dimensioni per ogni parola del vocabolario  $V$ . I pesi sono inizializzati random.

$$I = \begin{matrix} \text{dog}^{(1)} & 0.067 & 0.096 & 0.719 \\ \text{cat}^{(1)} & 0.157 & 0.380 & 0.424 \\ \text{eat}^{(1)} & 0.154 & 0.460 & 0.401 \\ \text{bark}^{(1)} & 0.131 & 0.432 & 0.404 \end{matrix}$$

Adesso ipotizziamo che la nostra parola target sia *dog* e che lo SG debba predire le sue parole contesto. Moltiplichiamo quindi la matrice per il one-hot vector di *dog*.

Il risultato è l'embedding di *dog*.

A questo punto l'embedding di *dog* è moltiplicato per i pesi della matrice  $O_{3x4}$ , i cui vettori colonna sono gli embedding di output per le parole in  $V$ . Questo produce il vettore di output a 4 dimensioni  $I$ . Ciascuna dimensione di  $I$  corrisponde al prodotto scalare tra l'embedding di input del target e l'embedding di output per il lessama nel vocabolario  $V$ .

$$\hat{\text{dog}} = \begin{matrix} \text{dog}^{(1)} & 0.067 & 0.096 & 0.719 \\ \text{cat}^{(1)} & 0.157 & 0.380 & 0.424 \\ \text{eat}^{(1)} & 0.154 & 0.460 & 0.401 \\ \text{bark}^{(1)} & 0.131 & 0.432 & 0.404 \end{matrix} = \text{dog}^{(1)}(0.067 \ 0.096 \ 0.719)$$

$$\hat{\text{dog}} = \begin{matrix} \text{dog}^{(1)} & 0.067 & 0.096 & 0.719 \\ \text{cat}^{(1)} & 0.157 & 0.380 & 0.424 \\ \text{eat}^{(1)} & 0.154 & 0.460 & 0.401 \\ \text{bark}^{(1)} & 0.131 & 0.432 & 0.404 \end{matrix} = \text{dog}^{(1)}(0.067 \ 0.096 \ 0.719)$$

$$\begin{matrix} \text{dog}^{(0)} & \text{cat}^{(0)} & \text{eat}^{(0)} & \text{bark}^{(0)} \\ \text{dog}^{(1)}(0.067 \ 0.096 \ 0.719) & \left( \begin{matrix} 0.236 & 0.220 & 0.424 & 0.064 \\ 0.265 & 0.029 & 0.045 & 0.089 \\ 0.107 & 0.121 & 0.046 & 0.752 \end{matrix} \right) = \\ \mathbf{I} & (0.086 \ 0.075 \ 0.008 \ 0.553) \end{matrix}$$

Quindi il vettore di input viene moltiplicato per la matrice dei pesi.

La matrice di output sarà la trasposta di quella precedente.

Le reti neurali sono spesso utilizzate per eseguire la classificazione multinomiale, ovvero assegnare il vettore di input a una delle n classi possibili. Nel modello Skip-gram, le classi sono le n parole del vocabolario da prevedere. Il vettore di output  $o$  viene trasformato con la funzione softmax.

$$\text{softmax}(o_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)}$$

Per capire quale parola deve predire il sistema, trasforma la matrice dei pesi in matrice di probabilità, questo viene fatto con il vettore softmax.

La funzione softmax è una generalizzazione della funzione sigmoide che trasforma il vettore di output  $o$  in un vettore di valori reali compresi nell'intervallo [0,1] la cui somma è uguale a 1. Questo vettore rappresenta la distribuzione di probabilità sulle n classi possibili  $y_1, \dots, y_n$  dato l'input della rete  $i$ .

La funzione softmax trasforma  $l_j$  in un vettore di probabilità, in modo che ogni dimensione  $k$  corrisponda a  $p(l_j = l_k | \text{dog})$ . Questo vettore rappresenta la previsione della rete del lessema target dato il contesto.

$$\text{softmax}(0.086 \ 0.075 \ 0.008 \ 0.553) = (0.22 \ 0.22 \ 0.21 \ 0.35)$$

La funzione di perdita (loss) viene quindi applicata al vettore di probabilità per ottenere il segnale di errore e aggiornare i parametri della rete con la **back-propagation**

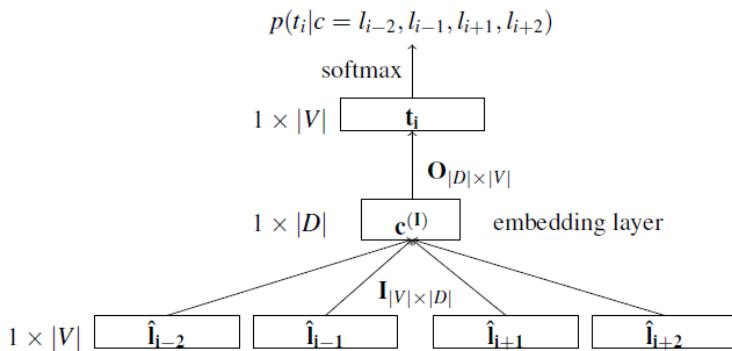
- Ad esempio, supponiamo che la parola corretta che compare a destra di dog sia bark. Allora la funzione di perdita confronta il vettore (0.22 0.22 0.21 0.35) con il vettore *bark* (0 0 0 1)

La differenza tra questi vettori viene usata come il segnale di errore per aggiornare i pesi delle matrici della rete con la back-propagation

### Continuous bag of Words (CBOW)

CBOW impara a prevedere il target  $t_i$  date le parole di contesto  $l_{i-n}, \dots, l_{i-1}$

Parte dai **contesti** a sx e a dx per predire la parola **target** in mezzo



CBOW impara i word embeddings prevedendo i lessemi target in base alle parole della finestra di contesto circostante (la cui dimensione è un iperparametro da impostare)

1. per ogni lessema  $l_j$  nella finestra di contesto che circonda  $t_i$ , il suo vettore di input  $\hat{l}_j$  viene moltiplicato per la matrice di pesi  $I_{|V| \times |D|}$  dello strato input-embedding. Poiché ogni vettore di righe di  $I$  è un embedding di input per una parola in  $V$ , e  $\hat{l}_j$  è un vettore one-hot il risultato della moltiplicazione è l'incorporamento in ingresso di  $l_j$ :

$$I_j^{(I)} = \hat{l}_j I$$

2. Gli embedding di input dei lessemi nella finestra di contesto sono mediati per ottenere l'embedding di input del contesto c:

$$c^I = \frac{1}{2n} \sum_{i-n \leq j \leq i+n, j \neq i} I_j^{(I)}$$

Dal 2018 in poi nascono altri tipi di modelli che vanno a risolvere i problemi di polisemia non analizzabili con modelli come Word2Vec.

Per ottimizzare il training, word2vec usa esempi negativi, come un diverso obiettivo di apprendimento

- per ogni coppia target-contesto  $\langle t, c \rangle$  nei dati di addestramento (ad esempio,  $\langle \text{dog}, \text{bark} \rangle$ ), word2vec genera k coppie  $\langle t, \sim c \rangle$  (ad esempio,  $\langle \text{dog}, \text{sky} \rangle$ ), tali che  $\sim c$  è una parola di **disturbo** campionata casualmente dal vocabolario V
- Queste coppie sono chiamate **esempi negativi** perché sono state generate **casualmente**, invece di essere osservate nel corpus.
- L'obiettivo della rete è trovare i parametri che massimizzano la probabilità delle coppie osservate e minimizzano le probabilità degli esempi negativi.
  - Idea: la rete predice semplicemente se una parola ha una probabilità più alta della probabilità di altri esempi negativi. In questo modo, ad ogni passaggio la rete discrimina gli esempi positivi (che possono stare sul contesto) e negativi, in modo da semplificare la computazione al liv. matematico

## Lezione 16 – 20/05

### Il problema della polisemia

I tradizionali DSMs sono modelli del **lessico**, intesi come depositi di oggetti lessicali fuori dal contesto.

DSMs di solito descrive il significato degli oggetti lessicali con **vettori tipo**.

Questi vettori sono **indipendenti** dal contesto, perché “riassumono” l'intera storia distributiva degli elementi lessicali in un'unica entità.

- **Meaning conflation deficiency** -> come se tutti i sensi fossero “mischiati”.  
I vettori distributivi non sono in grado di discriminare tra i diversi **sensi** delle parole → quindi manca la capacità di affrontare la questione della **polisemia**

Vettore = mescola le informazioni dei vari significati semanticci di una parola

Vicini più vicini prodotti da Skip-Gram per *play*:

*playing, game, play\_N, audition, player, match, sing, star, coach, badminton*

### Co-composizione

Pustejovsky (1995), and many others

Quando le parole sono composte, tendono a influenzare i significati l'una dell'altra → i significati delle parole cambiano con il contesto

- *The horse runs vs. The water runs*
- *“The horse horse-like runs”*
- cf. close to the issue of **Polysemy**

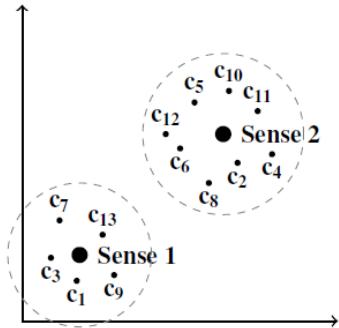
## Sensi come Cluster di contesti

Una famiglia influente di modelli distribuzionali rappresenta sensi di parola come **cluster di contesti simili**

Possiamo rappresentare i sensi di una parola andando a vedere i contesti in cui ricorre

Collezioniamo un insieme di contesti  $\{c_1, \dots, c_n\}$  di un lessema  $l$ , che consiste di frasi nelle quali i token di  $l$  o le finestre di parole che li circondano

- *The bat is mammal.*
- *Bats usually fly at night.*



I contesti sono rappresentati come **vettori**  $\{c_1, \dots, c_n\}$ , detti **vettori di contesto**, che vengono poi raggruppati in un numero predefinito di cluster sulla base della loro somiglianza.

I **vettori di contesto** possono essere semplicemente costruiti facendo la **media** dei vettori delle loro parole (quindi ciascun contesto è rappresentato facendo la media delle parole che lo compongono)

- Clustering: raggruppa vettori di contesti simili in aree dello spazio simili

Ogni senso è rappresentato da un **vettore di senso** o da un **sense embedding** corrispondente al **centroide** di un cluster.

Il centroide di un insieme di vettori  $C = \{u_1, \dots, u_n\}$  è un vettore  $m$  tale che ogni componente  $m_i$  è la media dei valori dell' $i$ -esima componente dei vettori in  $C$ .

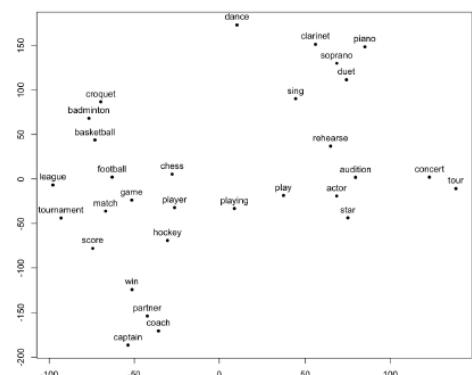
I lessemi sono associati a più vettori di senso, ognuno dei quali rappresenta uno dei suoi usi prototipici.

Pertanto, questi modelli sono chiamati DSM multi-prototipo (multi-prototype DSMs).

## Sensi come cluster dei vicini

Un altro gruppo di metodi distribuzionali sfrutta la rappresentazione dei sensi delle parole come cluster di vicini simili

- Se lo spazio dei sensi è vettoriale lo spazio è continuo, quindi ci permette di catturare la gradualità dei significati delle parole (posso stabilire i rapporti di vicinanza/similarità tra un senso e l'altro)



## Dai type vectors ai token vectors

- Embeddings contestuali (contextual embeddings): ogni singola istanza di una parola è rappresentata con un distinto **vettore token** che codifica gli aspetti del suo contesto
- Gli stati nascosti delle reti neurali addestrate a codificare sequenze linguistiche come vettori forniscono rappresentazioni lessicali sensibili al contesto
- Il vettore nascosto generato da una rete per *cut* in *The butcher cuts the meat* non è identico al vettore generato per lo stesso verbo in *The lumberjack cuts the wood*

Idea: non costruiamo più dei vettori tipo, ma solo i vettori token → a ogni parola nel contesto (frase) siamo in grado di trovare il vettore di quella parola token nel contesto → modelli che imparano vettori context-sensitive (codificano pezzi del contesto).

Se addestriamo una rete neurale ad elaborare “sequenze di elementi” (interi frasi) i vettori degli stati interni finiranno per essere intrinsecamente contestualizzati → costruzione di vettori diversi per ogni occorrenza che ha un significato diverso.

INVECE: in word2vec il modello impara 1 vettore per ogni esempio in cui si trova la parola  
Apertura della strada ai LLM

### Modelli di fondazione, ovvero Large Language Models (LLM)

#### *Caratteristiche comuni*

Reti neurali artificiali profonde (cioè multistrato) pre-addestrate su enormi quantità di dati testuali non etichettati.

Idea: modelli sempre addestrati in un task di predizione

Nella fase di pre-addestramento, la rete acquisisce una grande quantità di conoscenze di base dai corpora testuali addestrandosi come modello linguistico.

Un modello linguistico neurale (NLMs) viene addestrato con un task di predizione di stringhe auto-supervisionato.

I modelli **autoregressivi** (generativi) (ad esempio, **GPT**) prevedono una parola in base al contesto precedente.

- *The dog is chasing a red...*

I modelli di **de-noising\*** – BIDIREZIONALI (ad esempio, **BERT**) predicono una parola mascherata in un contesto bidirezionale (vede il contesto sia a sx che a dx)

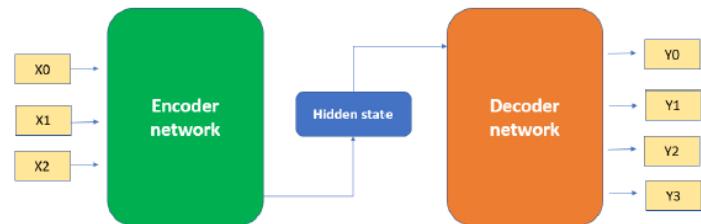
- *The dog is [MASK] a red ball*

### Modelli Encoder Decoder (seq2seq)

L'**Encoder** trasforma una sequenza input in un vettore (o sequenza di vettori).

Il **Decoder** genera una sequenza output dal vettore di input (o sequenza di vettori) → quindi prende il vettore e lo decodifica in un'altra sequenza di parole

Idea: vettore (hidden) che codifica tutte le sequenze di parole



### Architettura

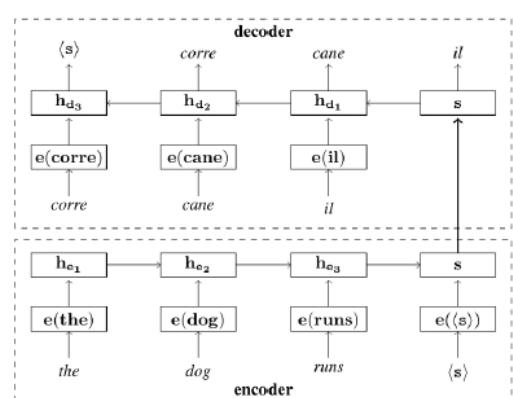
#### Encoding di sequenze con reti ricorrenti (Recurrent Networks)

##### **LSTM (Long Short Term Memory Network)**

Tipo di rete ricorrente nella quale lo strato nascosto riceve informazioni dall'input e dagli strati nascosti precedenti, specializzata nella modellazione di sequenze di parole

- Encoding = fase di creazione della rappresentazione interna
- Decoding = fase di generazione (es. ChatGPT = modello generativo)
- Vettori = intrinsecamente contestualizzati (tengono traccia del modello precedente)

POI: avvento dei transformers (sempre strutture encoder-decoder)



## Encoding Sequences with Transformers

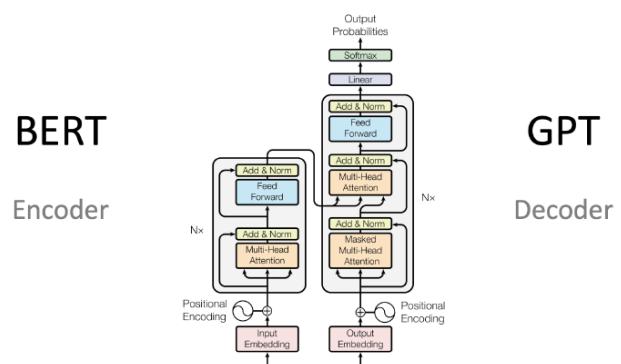
Vaswani et al. (2017). "Attention is all you need", NIPS 2017

Tengono traccia dell'ordine degli elementi ma sfruttano

il meccanismo dell'**attenzione** (e non più delle  
ricorrenze)

Transformer = costituito da vari **blocchi**, dove ognuno è  
una sequenza di neuroni

Si possono prendere anche solo la parte di encoder  
(BERT) o di decoder (GPT)



## Il meccanismo dell'attenzione (Attention Mechanism)

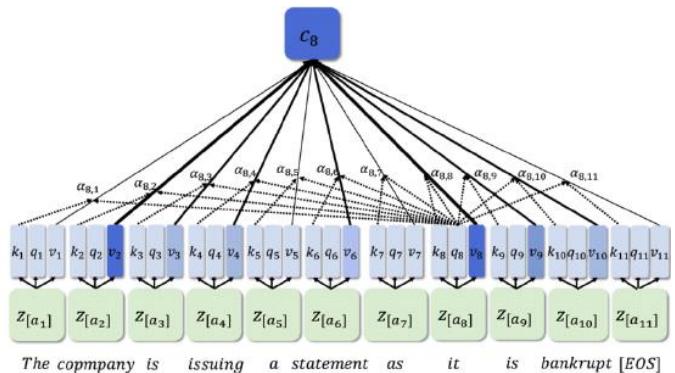
I vettori generati da ogni strato sono le somme pesate (ponderate) sull'intera sequenza di input e perciò sono intrinsecamente contestualizzati:

- sequenza di layers
- Input = intera frase

Di livello in livello imparano a combinare  
ciascun vettore di una parola con ciascun  
vettore di ogni altro livello della rete.

Sono combinazioni di vettori di altre parole.

- Imp: pesi
- Rete: impara che alcune parole sono più  
importanti di altre attraverso i pesi, che  
si modificano nel tempo



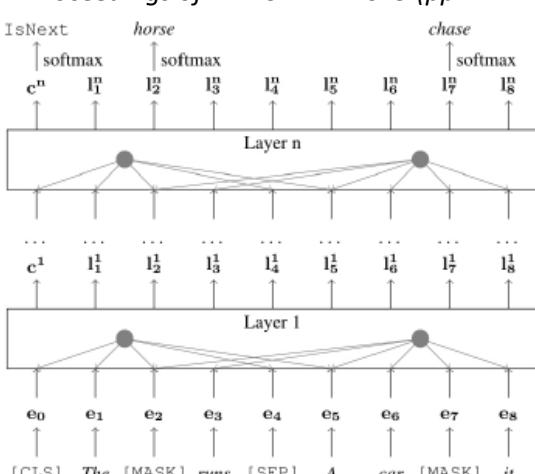
Vanishing gradient → più si allunga il contesto, più diminuisce l'influenza che una parola ha su un'altra  
molto lontana

**INVECE:** il meccanismo dell'**attenzione** lavora in **parallelo**

## BERT

Devlin, J. et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding".

In Proceedings of NAACL-HLT 2019 (pp. 4171-4186)



Solo encoder → task: **mask language modelling** (task di predizione), in cui una frase di input ha delle parole "mascherate" e bisogna capire quali parole sono mascherate

\***de-noising** → è come se dovessimo togliere il noise (rumore), cioè le parole mascherate

BERT ha una nozione di "senso" delle parole, perché nel suo spazio semantico ci sono vari clusters

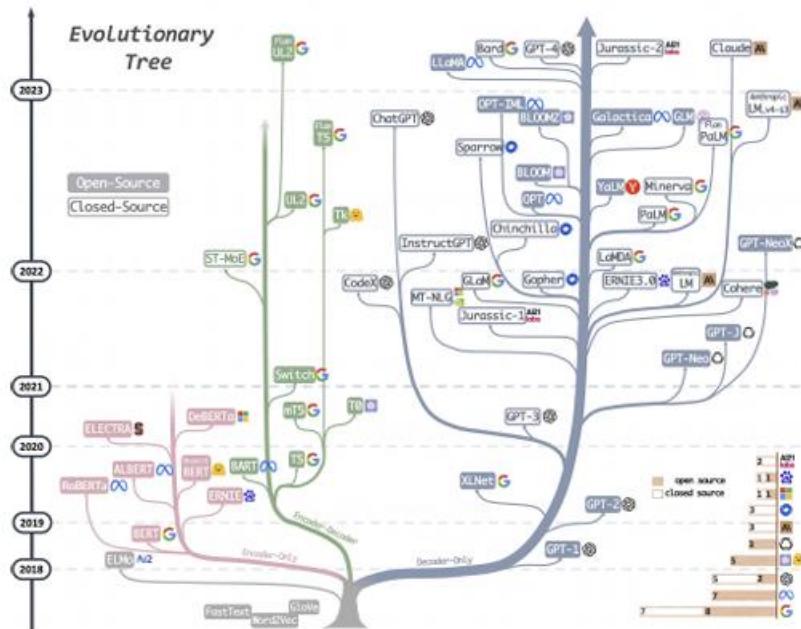
Più saliamo nella rete e più i sensi delle parole diventano sempre più divergenti e separati

Somiglianza del coseno tra le incorporazioni contestuali dei lessemi target in grassetto, generate dal livello 12 di BERTbase

The professor <b>began</b> the conference.	0.81	The professor <b>opened</b> the conference.
The professor <b>unlocked</b> the door.	0.57	
		The professor <b>opened</b> the door.
The professor <b>began</b> the conference.	0.53	
The professor <b>unlocked</b> the door.	0.77	
		The horse <b>runs</b> fast.
The horse <b>gallops</b> fast.	0.73	
The water <b>flows</b> fast.	0.70	
		The water <b>runs</b> fast.
The horse <b>gallops</b> fast.	0.54	
The water <b>flows</b> fast.	0.85	

## Lezione 17 - 27/05

### The Tree of LLMs



Le radici di questo grafico sono interessanti: vediamo anche Word2Vec; quindi, tutti i LLMs (o quasi) derivano dai metodi sviluppati per questo modello.

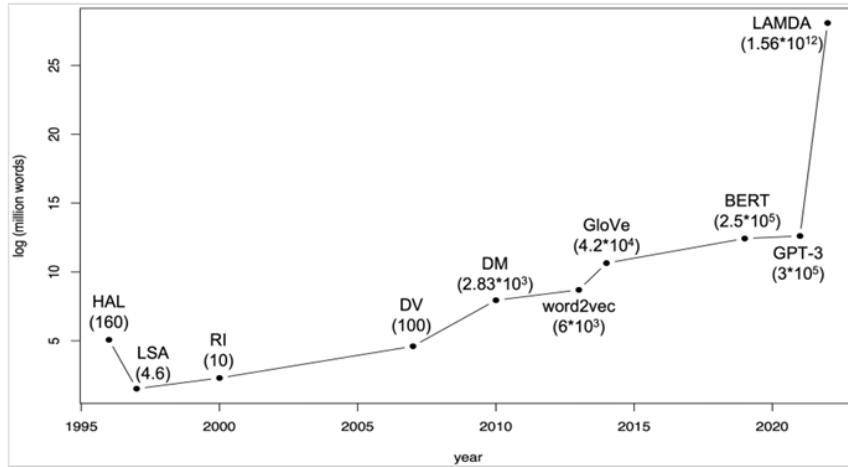
### The Model Size

I modelli distribuzionali sono stati caratterizzati da una crescita esponenziale sia nell'architettura che nel numero dei testi di training.

- BERT Large: 24 livelli e 340 milioni di parametri
- GPT-3: 96 livelli e 175 miliardi di parametri

GPT-3 ha un training set nell'ordine di 499 miliardi di token

## The Growth of Training Corpora



La crescita della curva è esponenziale.

## The Amount of Information LLMs Encode

La semantica distribuzionale del primo periodo era una semantica per lo più del lessico, finalizzata cioè alla creazione di dizionari.

Nei nuovi modelli, questi imparano direttamente dal testo grezzo nei task di predizione molto più nel dettaglio aspetti della sintassi, semantica e addirittura codificano ed imparano molti aspetti della pragmatica. Quindi imparano a comprendere molte più informazioni rispetto ai modelli precedenti.

Questi modelli pur nonostante non siano addestrati a svolgere un determinato task, sembrano capaci di farlo dopo l'addestramento dai testi.

I LLMs come GPT-3 rivelano un'abilità emergente per svolgere task linguistici vari (e.g. traduzione, question-answering, etc.) senza essere esplicitamente addestrati per ciò.

**Esempio ANTHROPIC:** altro importante nome in questo ambito. Cerca di capire quali "concetti" (es. Positività, arroganza) sono associati ai PATH nei LLMs.

## The Way LLMs are Used

Prima i DSMs generavano word embeddings che venivano usati come features nei ML algoritmi supervisionati specializzati per risolvere un determinato task.

Adesso invece questi modelli vengono addestrati (o meglio pre-addestrati) incorporano una grandissima quantità di informazioni. A questo punto i modelli possono essere adattati per risolvere task di tipo diverso (chiamati anche Foundation Models).

Ci sono vari tipi di adattamento:

- **Fine-tuning:** vengono sintonizzati i pesi. Un modello generalista va ad adattare i propri pesi per svolgere vari tipi di task.
- **Prompting:** viene posta la domanda al modello e questo svolge il task richiesto. Es. Gli chiediamo se ci traduce una frase da una lingua ad un'altra.

## Apprendimento contestuale (In-Context Learning)

L'apprendimento contestuale (In-Context Learning o ICL) è una tecnica utilizzata nei modelli di linguaggio avanzati, come GPT, dove il modello viene "incoraggiato" a completare un compito fornendo un contesto o

una sequenza di esempi pertinenti direttamente nell'input (prompt). Invece di addestrare il modello ulteriormente con nuovi dati, si fornisce una serie di esempi nel prompt per guidare la risposta del modello. Questo metodo sfrutta le capacità del modello di comprendere e generalizzare dai pattern presenti nei dati di addestramento originale.

Ci sono due tipologie:

- Zero-shot: il modello predice la risposta avendo a disposizione unicamente una descrizione in linguaggio naturale del task. Non vengono eseguiti aggiornamenti del gradiente.  
Es.



- One-shot: in aggiunta alla descrizione del task, il modello vede un unico esempio del task. Non vengono eseguiti aggiornamenti del gradiente.  
Es.



## Do NLMs work like the human brain?

L'idea fondamentale è che il nostro cervello è costantemente impegnato nel task di predizione, esattamente come fa un modello di questo tipo.

Altra importante cosa: se noi calcoliamo con modelli come GPT l'entropia puntuale di una parola, cioè la **surprisal** questa sarà più alta se la parola predetta è fuori contesto, al contrario più bassa. Se a questi modelli facciamo calcolare la surprisal di una parola dato il contesto linguistico, questi saranno in grado di capire il processing linguistico, la psicolinguistica umana. Quindi quelle rappresentazioni che il modello si crea, sono molto vicine a ciò che capita nella nostra mente.

Gli embedding contestuali (cioè, quelli prodotti da questi modelli) sono in grado di contenere informazioni molto più precise e simili ai procedimenti umani.

In maniera molto ottimistica infatti, gli autori di questo paper, dicono: “DLMs are generative in the narrow linguistic sense of being able to generate new sentences that are **grammatically, semantically, and even pragmatically well-formed at a superficial level**”

## The Pitfalls of Semantic Similarity

Similarità semantica: due parole se sono distribuzionalmente simili allora sono anche semanticamente simili. Nozione di similarità semantica molto complessa però.

Cosa vuol dire?

Banalmente che hanno degli aspetti del significato simili. Però ci sono diverse nozioni di similarità semantica basate sul tipo di significato che noi consideriamo.

In psicologia, la somiglianza gioca un ruolo cruciale nelle teorie della cognizione e nella ricerca empirica sulla memoria semantica e sul lessico mentale. È un ingrediente essenziale della categorizzazione, del recupero della memoria, del ragionamento, dell'induzione, ecc. Le parole sono semanticamente simili nella

misura in cui i loro significati condividono aspetti comuni. Poiché esistono molteplici fonti di tale comunanza, è possibile identificare vari tipi di somiglianza semantica.

E.g. *cavallo* e *cane* sono simili in quanto quadrupedi, animali, ecc... però se prendiamo anche la figura di "auto", a questo punto a cosa è più simile *cavallo*? A *cane* in quanto animale o ad *auto* in quanto anch'esso mezzo di trasporto?

### **Semantic Similarity/Relatedness in Linguistics**

Quindi esistono una serie di problemi alla base e dunque diverse tipologie di similarità semantica.

- **Similarità attributiva:** dati due lessemi a e b, la loro somiglianza attributiva dipende dal grado di corrispondenza tra le caratteristiche di a e le caratteristiche di b. Auto e camion sono simili dal punto di vista attributivo perché hanno diverse caratteristiche in comune (4 ruote, dotati di motore, etc.). La somiglianza attributiva è anche chiamata tassonomica, poiché gli elementi che condividono attributi salienti appartengono alla stessa categoria tassonomica (ad esempio, auto e camion appartengono alla categoria dei veicoli).
- **Similarità relazionale:** due parole sono simili dal punto di vista relazionale quando hanno un legame di qualche tipo tra loro. È il grado di corrispondenza nella relazione che lega le due parole.  
e.g. Macchina e veicolo sono simili basandosi sulla relazione esistente tra loro di iperonimia. Viene chiamata anche similarità analogica poiché rappresenta la similarità come se questa fosse un'analogia: *a* sta a *b* come *c* sta a *d*.  
*E.g. Firenze sta alla Toscana come Roma sta al Lazio.*  
Questo tipo di relazione è fondamentale sotto molti tipi di aspetti. Es nella mente umana è indispensabile il ragionamento logico e questa relazione in qualche modo la riproduce.

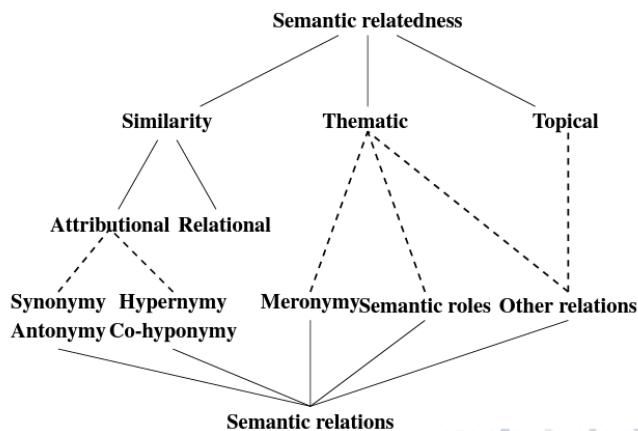
C'è un'espressione che spesso viene confusa con il concetto di similarità semantica:

- **Semantic Relatedness** (si traduce male in italiano): due entità hanno un qualche tipo di relazione semantica tra loro. E.g. Napoli e Campania hanno una relazione semantica (città che sta nella regione), ma non sono semanticamente simili (non sono due città o due regioni); stessa cosa per guidatore ed auto.  
La nozione di semantic relatedness può inoltre avvenire a diversi gradi: un concetto può essere agente del secondo, o viceversa.  
Esiste anche la **Topical Relatedness**: tra lessemi che dipendono dalla stessa sfera di significato o dominio semantico.  
e.g. verbo e fonema sono relazionati (simili) tematicamente perché appartengono al dominio linguistico

Piuttosto che sulle nozioni generali di similarità e correlazione, la ricerca nella semantica lessicale si è invece concentrata **sull'investigazione delle proprietà di specifici tipi di relazioni semantiche**. Le relazioni paradigmatiche sono state considerate il fondamento dell'organizzazione del lessico, sia a livello linguistico che cognitivo.

**Sinonimia** (divano-sofà), **antonimia** (buono-cattivo), **iperonimia** (animale-cane), **co-ponimia** (cane-gatto) e **meronimia** (coda-cane). Sinonimia, iperonimia e co-ponimia possono essere considerate come esempi di **similarità tassonomica**. Gli antonimi sono anche caratterizzati da una forte similarità paradossale di simultanea somiglianza e differenza (Cruse 1986, Lexical Semantics, Cambridge): gli antonimi sono identici in ogni dimensione di significato eccetto una. I **ruoli semantici** (ad esempio, agente, paziente, strumento) sono esempi di **correlazione tematica**.

## Types of Semantic Similarity and Relatedness



Evince il problema della complessità di questa espressione.

Questi modelli alla fine hanno diversi problemi e difficoltà nell'individuazione delle relazioni semantiche molto particolari.

Non esiste un unico concetto di similarità, ma piuttosto una famiglia complessa e multiforme di nozioni distinte ma sovrapposte che differiscono per granularità. I confini tra queste nozioni sono spesso difficili da tracciare, e gli stessi elementi su cui si basano (ad esempio, attributi, relazioni, scenari, argomenti, ecc.) **sono intrinsecamente sfumati** e dipendenti dal contesto. Ciò implica che i nostri giudizi di similarità (correlazione) possono cambiare drasticamente a seconda degli aspetti del significato su cui ci concentriamo. Come affermava Goodman (1972), non esiste una nozione assoluta di similarità. Molte coppie di lessemi possono quindi essere giudicate simili da **più di una prospettiva contemporaneamente**. *Gatto* e *cane* non sono solo simili per attributi, ma condividono anche molte relazioni con altre entità (ad esempio, cani e gatti giocano entrambi con le palle), appaiono spesso insieme negli stessi scenari o eventi (ad esempio, i cani inseguono i gatti), e appartengono allo stesso argomento (cioè, zoologia). Ci sono invece lessemi che soddisfano solo una delle definizioni sopra citate: fede e battesimo sono simili perché appartengono al dominio religioso.

## Relatedness Dataset

Noi vogliamo capire quanto questi modelli sono bravi a captare il significato delle parole.

Uno di questi è **MEN**, un dataset contenente 3000 coppie di parole. Ad ognuna di queste viene assegnato un valore di relatedness da nativi della lingua rispetto ad altre 50 coppie nel dataset.

Ciò che i rater fanno è: gli vengono mostrate due coppie di parole candidate. Il rater deve scegliere la coppia candidata le cui parole sono più correlate nel significato. Es. tra le coppie candidate *portafoglio-luna* e *macchina-automobile*, il rater dovrebbe scegliere la seconda.

## Similarity Dataset

**SimLex-999** contiene 999 coppie di verbi, nomi, aggettivi valutati da 50 soggetti per la loro "genuina" similarità semantica su una scala da 0 a 6 (poi convertita nell'intervallo 0-10). In questo sondaggio viene chiesto di confrontare coppie di parole e di valutare quanto sono simili spostando un cursore.

**SimVerb-3500** è un dataset di 3500 verbi inglesi, ognuno valutato da almeno 10 soggetti per la sua similarità semantica.

Task che ha come target l'analogia, e quindi la similarità relazionale.

## Analogy Tests

Un'analogia si viene a formare quando due coppie di parole sono a livello relazionale simili, ovvero quando possiedono una relazione simile tra loro.

Le analogie sono tipicamente scritte nella forma  $a : b = c : d$ , il che significa che a sta a b come c sta a d (ad esempio, ruota : auto = dito : mano). Il compito di completamento dell'analogia consiste nell'inferire l'elemento mancante in un'analogia incompleta  $a : b = c : ?$

Idea: Prendo 4 parole legate da un rapporto di similarità relazionale (e.g. iperonimi). Il task di analogia prevede che queste parole vengano sostituite da altre e ne venga lasciata una vuota. E.g. Italia : Roma = Svezia : ?

La risposta che il modello assegna deve essere Stoccolma.

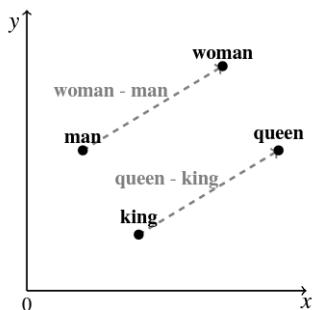
### Esempi di dataset:

- **MSR dataset:** contiene 8000 analogie morfosintattiche;
- **GOOGLE dataset:** contiene nove relazioni morfologiche (ad esempio, plurale) e cinque relazioni semantiche (ad esempio, valuta) con 20-70 coppie di parole uniche per categoria, combinate in tutte le possibili modalità per ottenere 8.869 analogie semantiche e 10.675 analogie sintattiche.
- **BATS dataset:** è un dataset bilanciato che include 40 categorie che coprono morfologia flessionale e derivazionale oltre a diversi tipi di relazioni semantiche, per un totale di 98000 analogie

good : better = heavy : heavier  
walk : walking = code : coding  
cow : cows = car : cars  
sit : sits = say : says  
man : woman = king : queen  
Madrid : Spain = Rome : Italy  
Europe : euro = USA : dollar

## Solving the Analogy Completion Task

Idea: parole che sono legate da relazioni semantiche simili hanno relazioni simili anche nello spazio del vettore.



E.g. parole come man-woman e king-queen, avrebbero una distanza vettoriale lineare misurabile che evince una relazione simile tra le parole (poiché sono lunghe uguali)  
Data l'analogia  $a : b = c : d$ , si assume che il vettore differenza  $b - a$  sia simile al vettore differenza  $d - c$ , poiché le coppie di parole  $a : b$  e  $c : d$  condividono relazioni simili:  $b - a \approx d - c$   
Da questa equazione, possiamo derivare la seguente equivalenza:  $c + b - a \approx d$

Ad esempio, data l'analogia uomo : donna = re : regina, il vettore regina dovrebbe essere molto simile al vettore  $re + donna - uomo$ .

Data l'analogia  $a : b = c : ?$ , il compito di completamento dell'analogia viene risolto cercando il lessema t il cui vettore ha il coseno più alto con  $c + b - a$ .

Relazioni semantiche = codificate da relazioni lineari nello spazio (ma in realtà lo spazio è molto più caotico...)

## Similarità/Correlazione Semantica e DSMs

L'Ipotesi Distribuzionale stessa deve essere **relativizzata**. La verità dell'affermazione che le semeli simili tendono ad apparire in contesti simili dipende infatti dal **tipo** di somiglianza semantica su cui ci concentriamo.

Domande chiave per la semantica distribuzionale:

- Quale tipo di somiglianza è meglio catturato da quale DSM (Modello di Semantica Distribuzionale)?
- In che misura i parametri del modello e i tipi di contesto influenzano lo spazio di somiglianza semantica rappresentato da un particolare modello?

## Static DSMs

Sono stati presi diversi modelli e testati su una grande quantità di dataset specifici (similarità semantica, semantic relatedness, similarità lessicale, task di analogia etc.)

Questi sono alcuni dei risultati:

Model	Context	Vector type	Dimensions
<b>Matrix count models</b>			
PPMI	window.{2,10}; syntax.{typed,filtered}	explicit	10,000
SVD	window.{2,10}; syntax.{typed,filtered}	embedding	300; 2,000
LSA	document	embedding	300; 2,000
LDA	document	embedding	300; 2,000
GloVe	window.{2,10}	embedding	300; 2,000
<b>Random encoding count models</b>			
RI	window.{2,10}	embedding	300; 2,000
RI-perm	window.{2,10}	embedding	300; 2,000
<b>Predict models</b>			
SGNS	window.{2,10}; syntax.{typed,filtered}	embedding	300; 2,000
CBOW	window.{2,10}	embedding	300; 2,000
FastText	window.{2,10}	embedding	300; 2,000

Da questi risultati vediamo molto bene come i punteggi siano alti su dataset sintattici o morfo-sintattici mentre più bassi in quelli semantici. Quindi ci dimostra come lo studio di X riguardante lo studio sulla distanza lineare sia molto più complesso in realtà.

## Type-level Lexical Semantic Tasks

Dataset	Size	Metric	Dataset	Size	Metric
<i>Synonymy</i>					
TOEFL	80	Accuracy	AP	402	Purity
ESL	50	Accuracy	BATTIG	5,231	Purity
<i>Similarity</i>					
RG65	65	Correlation	ESSLLI-2008-1a	44	Purity
RW	2,034	Correlation	ESSLLI-2008-2b	40	Purity
SL-999	999	Correlation	ESSLLI-2008-2c	45	Purity
SV-3500	3,500	Correlation	BLESS	26,554	Purity
WS-353	353	Correlation	<i>Analogy</i>		
WS-SIM	203	Correlation	SAT	374	Accuracy
<i>Relatedness</i>					
WS-REL	252	Correlation	GOOGLE	19,544	Accuracy
MTURK	287	Correlation	SEMEVAL-2012	3,218	Accuracy
MEN	3,000	Correlation	WORDREP	237,409,102	Accuracy
TR9856	9,856	Correlation	BATS	98,000	Accuracy

## Results of Static DSM Evaluation

Dataset	Score	Model	Dataset	Score	Model
<i>Synonymy</i>					
TOEFL	0.92	FastText.w2.2000	AP	0.75	SVD.synt.300
ESL	0.78	SVD.synt.2000	BATTIG	0.48	SGNS.synt.300
<i>Similarity</i>					
RG65	0.87	GloVe.w10.2000	ESSLLI-2008-1a	0.95	SVD.synt.300
RW	0.48	FastText.w2.300	ESSLLI-2008-2b	0.92	SGNS.w2.2000
SL-999	0.49	SVD.synt.2000	ESSLLI-2008-2c	0.75	SGNS.w2.2000
SV-3500	0.41	SVD.synt.2000	BLESS	0.88	SVD.synt.2000
WS-353	0.71	CBOW.w10.300	<i>Analogy</i>		
WS-SIM	0.76	SVD.w2.2000	SAT	0.34	SVD.synt.300
<i>Relatedness</i>					
WS-REL	0.66	CBOW.w10.300	MSR	0.68	FastText.w2.300
MTURK	0.71	FastText.w2.300	GOOGLE	0.76	FastText.w2.300
MEN	0.79	CBOW.w10.300	SEMEVAL-2012	0.38	SVD.synt.300
TR9856	0.17	FastText.w2.300	WORDREP	0.27	FastText.w2.300
			BATS	0.29	FastText.w2.300

## Contextual DSMs

Stessa cosa per BERT, capiamo da questi dati la non super efficienza se utilizzato nel contesto del significato.

Il type embedding di una parola target t viene ottenuta facendo la media delle sue rappresentazioni BERT (Bommasani et al., 2020):

$$t = \text{media}(t_{c1}, \dots, t_{cn})$$

- Ogni contesto ci corrisponde ad una frase s
- S è un esempio random di frase estratta dal corpus ( $|S| = 10$ ), dove appare il target
- $t_{ci}$  è l'embedding del token per t nel contesto ci

## BERT vs Embeddings statici

Lenci et al. (2022), “A comparative evaluation and analysis of three generations of Distributional Semantic Models”, *Language Resources & Evaluation*

Dataset	Static				Dataset	BERT			
	BERT.F4	BERT.L4	BERT.L			F4	L4	L	
<b>Synonymy</b>									
TOEFL	0.92	0.72	0.89	0.82	AP	0.75	0.52	0.63	0.55
ESL	0.78	0.60	0.60	0.64	BATTIG	0.48	0.22	0.40	0.35
<b>Similarity</b>									
RG65	0.87	0.74	0.81	0.78	ESSLLI-2008-1a	0.95	0.68	0.73	0.70
RW	0.48	0.37	0.48	0.36	ESSLLI-2008-2b	0.92	0.82	0.75	0.75
SL-999	0.49	0.49	<b>0.55</b>	<b>0.50</b>	ESSLLI-2008-2c	0.75	0.64	0.62	0.58
SV-3500	0.41	0.34	0.40	0.27	BLESS	0.88	0.60	0.73	0.70
WS-353	0.71	0.61	0.62	0.57	<b>Analogy</b>				
WS-SIM	0.76	0.67	0.70	0.63	SAT	0.34	0.24	0.24	0.21
<b>Relatedness</b>									
WS-REL	0.66	0.56	0.51	0.47	MSR	0.68	<b>0.76</b>	0.69	0.68
MTURK	0.71	0.59	0.56	0.52	GOOGLE	0.76	0.38	0.66	0.64
MEN	0.79	0.70	0.69	0.64	SEMEVAL-2012	0.38	0.33	0.34	0.30
TR9856	0.17	0.13	0.14	0.13	WORDREP	0.27	0.22	<b>0.28</b>	0.22
					BATS	0.29	<b>0.30</b>	<b>0.33</b>	<b>0.34</b>

## Lezione 18 – 28/05

### Compositionally and Distributional Semantics

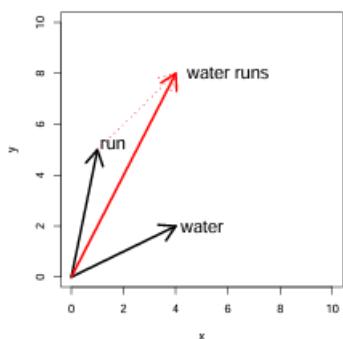
Principio di composizionalità: secondo Frege è grazie a questo che si comprende la generatività e possiamo comprendere il linguaggio. Ma modelli come GPT non seguono necessariamente questo approccio; eppure, sembrano comprendere il linguaggio e lo producono.

Come si può immaginare il processo di costruzione di una rappresentazione linguistica complessa quando noi rappresentiamo gli elementi linguistici come vettori?

Il lessico viene solitamente considerato il “bottleneck” (collo di bottiglia) per la semantica formale, questo perché esistono numerosi problemi nel gestire il comportamento contestuale e vago dei significati lessicali, la somiglianza semantica, le interpretazioni figurative, ecc...

La **composizionalità** è il collo di bottiglia per la semantica distribuzionale.

### Semantic Composition Operations in DSMs



Il **significato distribuzionale** di una frase è la **combinazione dei vettori** che rappresentano le parole e che quindi compongono le frasi.

Il metodo più comune è la somma.

Vector Composition as Vector addition

Se noi abbiamo una struttura linguistica composta da  $a$  e  $b$  che formano un certo sintagma. La rappresentazione di  $p$  è il vettore somma dei vettori di  $a$  e  $b$ .

Semplice vettore di somma (Landauer and Dumais 1997):  $p = a + b$

e.g. *chase cat* = *chase* + *cat*

Il metodo della somma preserva le dimensioni dei vettori componenti; è simile all'**unione**: i coefficienti che sono alti in entrambe i vettori delle parole rimarranno altri nel vettore del risultato.

	hacker	cheese	button
mouse	25	10	17
click	30	0	20
click mouse	55	10	37

Torniamo alla regola del parallelogramma: se usiamo il metodo della somma, questo è matematicamente identico alla **diagonale dei due vettori**.

### Meaning and Structure

Il modello additivo combina i significati delle parole e pertanto la rappresentazione semantica di un'espressione complessa è qualcosa di intermedio nell'interpretazione dei suoi componenti lessicali.

Il significato di *red car* non è la **media** dei significati di *red* e *car*.

Il vettore somma gode della proprietà commutativa e quindi **ignora completamente le strutture sintattiche**. E.g. *man bites dog* e *dog bites man* per questo tipo di modello hanno la stessa rappresentazione semantica.

Il modello additivo non è in grado di affrontare le modulazioni del significato che sono un sottoprodotto del processo di composizione. E.g. *red car* e *red wine* posseggono lo stesso vettore per la parola *red*.

Tutto ciò ha portato al tentativo di trovare modi più sofisticati per combinare i vettori.

## Lexical Function Model (LFM) and adjectival modification

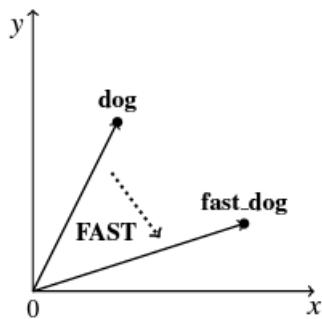
Questo metodo ha un collegamento diretto con la grammatica di Montague poiché propone la rappresentazione lineare-algebrica degli oggetti.

Quindi si suppone una divisione tra funzioni e argomenti. Ricorda: nella grammatica di Montague gli aggettivi sono funzioni che vengono applicate ai nomi per produrre un nome modificato.

Lavorando con vettori, queste funzioni diventano funzioni di matrici. Quindi facciamo il prodotto tra la matrice e il vettore.

$$\begin{pmatrix} 0.5 & 1 \\ 0.8 & 0 \end{pmatrix} \begin{pmatrix} 3 \\ 6 \end{pmatrix} = \begin{pmatrix} (0.5 * 3) + (1 * 6) \\ (0.8 * 3) + (0 * 6) \end{pmatrix} = \begin{pmatrix} 7.5 \\ 2.4 \end{pmatrix}$$

Es. nella frase *fast dog*, *FAST dog* diventa il rapporto tra la matrice di *FAST* e il vettore di *dog*. Le matrici si comportano come funzioni.



Questo metodo però non funzionava perfettamente come descritto dal piano cartesiano, non riusciva a costruire strutture di frasi complesse.

Torniamo alla somma vettoriale....

Il problema principale era che dato che la somma

gode della proprietà commutativa, dog bites man e man bites dog erano rappresentati esattamente allo stesso modo.

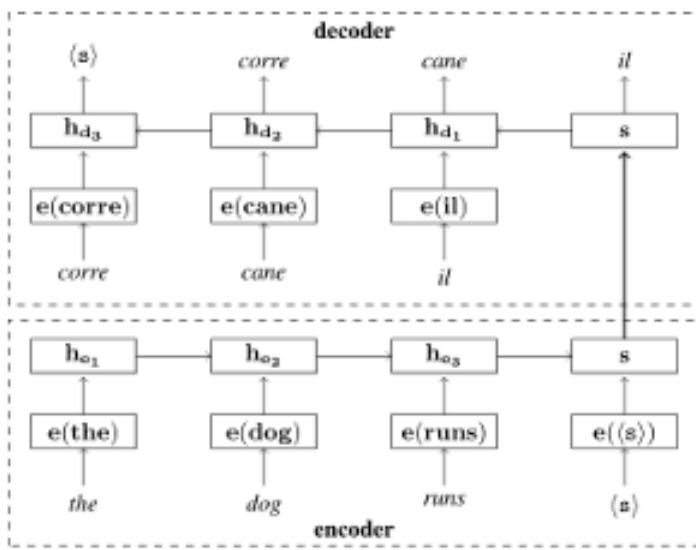
## Encoder – decoder models

## Sentence Embeddings

Idea: data una frase, proviamo a rappresentare la frase come un vettore che tenga conto non solo dei singoli componenti ma anche del loro ordine. Questo vettore di frase prende il nome di **sentence embedding**.

Questo vettore impara con una rete neurale che prende in input i singoli vettori delle singole parole e l'ordine e restituisce un vettore di frase.

Questo è stato possibile grazie ai modelli di encoder-decoder. Il primo approccio è stato possibile grazie alle reti ricorrenti, cioè quelle reti che procedono secondo stati che sono determinati dal vettore dello stato che però tiene conto anche del vettore nascosto dello stato precedente.



Uno dei primi esempi di questo tipo è l'architettura *seq2seq*, una rete di tipo encoder usata per produrre il vettore di rappresentazione della frase  $s_i$ , che viene poi alimentato come input in un'altra rete (il decoder), che lo utilizza per generare la frase  $S_i$ .

Una rete del genere è quindi in grado di distinguere le frasi *man bites dog* da *dog bites man* poiché si avranno delle rappresentazioni di tipo diverso.

Il sentence embedding è esattamente lo stato interno della rete una volta terminato questo

processo che tiene traccia di tutte le diverse informazioni.

**N.B: questo procedimento non è compositivo. Quando io costruisco la rappresentazione semantica della frase, contiene le rappresentazioni semantiche delle parole originali, come se fosse un assemblaggio dei diversi pezzi (parole) che continuo a distinguere nitidamente. Al contrario questi sentence embedding contengono un po' dell'informazione di questi elementi, ma questa è completamente mischiata e non più singolarmente distinguibile.**

Arrivati ad utilizzare questo procedimento però si è capito che lasciava un po' a desiderare. Quando si andavano a testare questi modelli su task effettivi funzionava comunque meglio il metodo dell'addizione vettoriale.

Alla fine dei conti questo sistema complesso della rete non era il vero valore aggiunto, ma questo era dato dai vettori utilizzati come input delle singole parole.

Quindi dal 2019 l'attenzione si è diretta verso un altro fenomeno: imparare i vettori contestuali delle singole parole, cioè, spostarsi sull'idea che la rappresentazione semantica di una frase non sia un vettore che si ottiene incollando i singoli vettori delle parole, ma i vettori delle singole parole sensibili al contesto, quindi **vettori contestuali**.

### The Current Limits of Sentence Embeddings

Si è capito che il vero arricchimento nell'ambito NLP è stato il **cambiare la rappresentazione dei vettori** e non i modelli encoders diversi o migliori.

Quindi la pratica comune utilizzata è la seguente: usare la rappresentazione di una frase come la sequenza dei vettori contestuali di ogni parola token.

E questo è il metodo che ha permesso di notare la differenza e analizzarla in frasi come *man bites dog* e *dog bites man*. Quindi l'informazione contestuale è codificata nelle singole parole stesse.

I modelli recenti (come GPT) non costruiscono una rappresentazione unitaria di una frase, ma il significato della frase nel suo complesso che non è un'unica rappresentazione vettoriale, ma una **sequenza di rappresentazioni vettoriali contestuali**.

Es. BERT: se riceve in input una frase, restituisce i vettori contestuali delle singole parole. Restituisce però anche il CLS che è una sorta di vettore che rappresenta la media pesata di tutti i vettori delle singole parole.

**SentenceBERT** è la versione fine-tuned di BERT che costruisce embeddings di frasi.

## Modelling Eye-Tracking Data with DSMs

Esperimento dell'eye-tracking.

Più alto è il coseno tra gli elementi (quindi elementi maggiormente simili), minore è il tempo di lettura.  
Quindi la correlazione misurata con Spearman risulta negativa.

Risultati Top Models:

GECO			PROVO		
DSM	Context	$\rho$	DSM	Context	$\rho$
BERT (12)	Additive Full	-0.54	BERT (12)	CLS Full	-0.66
BERT (12)	CLS	-0.53	GloVe	Additive Full	-0.65
GloVe	Additive Full	-0.45	BERT (12)	Additive Full	-0.65
SGNS	Additive Full	-0.39	SGNS	Additive Full	-0.60
FastText	Additive Full	-0.39	FastText	Additive Full	-0.57

## The Symbol Grounding Problem in Distributional Semantics

**Aspetto referenziale:** nella semantica formale si parlava dell'estensione del significato, cioè della capacità di identificare un oggetto in base al significato (e.g. saper identificare come è fatto un computer e non solo sapere da cosa è formato, conoscenza fondamentale per l'associazione del significato). La **capacità referenziale è la capacità di connettere il linguaggio al mondo esterno**.

Viceversa, la capacità inferenziale è quella che ci fa rendere in grado di stabilire i legami tra diversi lessemi o concetti (e.g. auto e veicolo).

A questo punto come può avere una competenza referenziale una macchina che impara tutto dai testi? (una macchina di questo tipo può imparare che *cherry* e *red* sono in qualche modo legati, ma non sarà capace di mappare *red* come colore vero e proprio reale)

Problema epistemologico e filosofico che ha attraversato la storia dell'AI, anche definito **The Octopus Argument**. Emily Bender nel 2020 ha mosso delle importanti critiche ai modelli definendoli come dei "pappagalli stocastici", cioè esclusivamente in grado di ripetere ciò che leggevano.

**Octopus Argument:** ci sono due individui A e B posti su due isole diverse sparse nell'oceano. Nel mare c'è una piovra intelligente che riceve input sia da A che B che parlano da soli. La piovra intelligente che ascolta le parole di A e B impara da ciò che dicono. Quindi ad un certo punto A crede di parlare con B e viceversa, ma in realtà è la piovra che ha imparato ascoltando e riesce a rispondere come avrebbero risposto i due parlandosi.

Ad un certo punto A dice a B di costruire una catapulta per lanciare i cocchi (in realtà parla con la piovra che a questo punto non sa rispondere).

**Morale:** i language model sono in grado di rispondere a delle domande ma non avendo la capacità referenziale non sono effettivamente in grado di interagire con il mondo esterno. Ovviamente tutta questa tesi era una critica mossa ai LLMs.

## The Chinese Room Argument

Immaginiamo che c'è una persona chiusa in una stanza che ha a disposizione esclusivamente un manuale di grammatica cinese scritta in cinese. Non ha quindi corrispondenze con un'altra lingua. È un problema simile a quello precedente.

**Morale:** un modello che combina esclusivamente dei simboli non è in grado di riferirsi con quei simboli a degli oggetti reali nel mondo. Altra critica.

Tutte queste critiche sollevano un problema in materia importante:

## The Symbol Grounding Problem in Distributional Semantics

Se un sistema manipola esclusivamente dei simboli, questo non sarà mai in grado di capire effettivamente il significato di questi simboli (non possiedono capacità referenziale).

Con la semantica distribuzionale si ha però un vantaggio: non si rappresentano i significati con dei simboli, ma con dei vettori numerici che codificano il contesto delle parole. Con questi vettori si possono codificare intrinsecamente informazioni di tipo diverso: pixel (immagini), informazioni testuali.

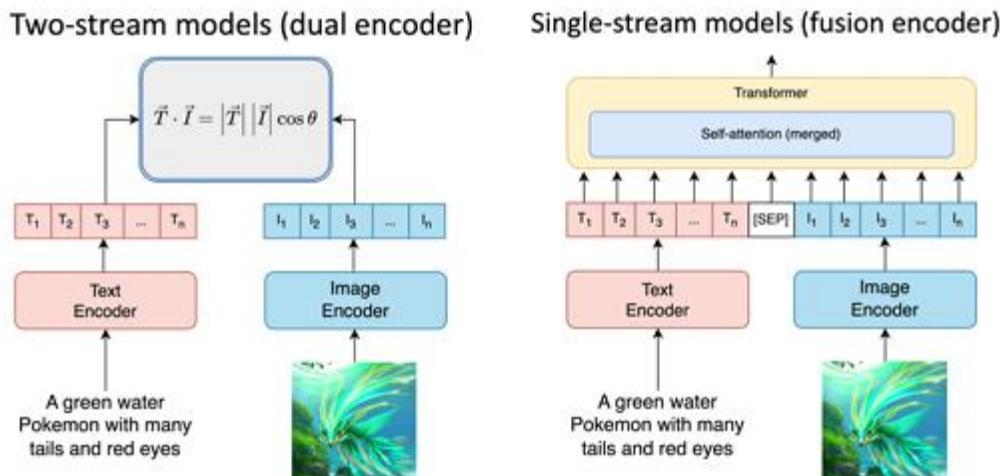
### Visual Embeddings

È vero quindi che questi modelli sono *ungrounded* ovvero non possiedono capacità referenziale, ma grazie a informazioni che codificano l'immagine che quella parola rappresenta questi modelli acquisiscono informazioni visive e **multimodali**.

Quindi se io estraggo informazioni dal testo, e informazioni estratte dall'immagine (e le unisco), posso imparare ad associare un'informazione ad un'immagine (abbinare due informazioni continue).

## Multimodal Language Models (MLMs)

Sono modelli che partono da dati che sono composti da immagini corredate dalle rispettive caption. Grazie ad una rete neurale trasformano queste informazioni in vettori (un vettore per l'immagine e uno per il testo) e poi le combinano in un unico vettore.



Questo procedimento è ciò che permette a modelli come GPT-4 o altri di analizzare o generare immagini.

## Representational Pluralism

Quindi non è vero che questi modelli non possiedono la capacità referenziale. Possono imparare a svilupparla.

Altro problema: anche i sistemi più capaci continuano a confondere spesso l'oggetto con il soggetto. Quindi ancora si ha il problema classico legato a chi rincorre chi nella frase *il cane rincorre il gatto*.

Quindi tanti problemi sono stati risolti, ma tanti altri ancora no.

Questi modelli multimodali ancora non risolvono totalmente il problema del grounding.