

# **Identificazione e disambiguazione della costruzione NPN con BERT**

Caso studio di Scivetti e Schneider sulla costruzione inglese

---

Corso di Semantica a.a. 2025/2026

18 Novembre 2025

Università di Bologna

# Indice

La costruzione NPN

Accedere alla conoscenza di un LLM

Probing

Il Dataset

Training e Test set

Task 1: Identificazione

Task 2: Identificazione (perturbando l'ordine delle parole)

Task 3: Disambiguazione semantica

## La costruzione NPN

---

## Cos'è la costruzione NPN

La costruzione NPN è stata ampiamente studiata in inglese da una prospettiva costruzionista, in particolare da  
**jackendoff2008construction e sommerer2021absent**

Construction Schema

*Noun<sub>1</sub> Preposition Noun<sub>2</sub>*

I due nomi nella costruzione devono avere forma identica, anche per la declinazione del numero. La costruzione non ammette nomi accompagnati da determinati.

# Cos'è la costruzione NPN

La costruzione può comparire in diverse posizioni sintattiche, ad esempio come modificatore avverbiale o come modificatore nominale.

## Esempi

- *I need you to get this word for word.*  
(modificatore avverbiale)
- *There is a rebellious quality to your day to day responses which have not gone unnoticed.*      (modificatore nominale)

# Significato e funzione

Significati delle costruzioni NPN istanziate da *to*

## **SUCCESSION**

Indica una **successione o iterazione**, spesso temporali o spaziali.

## **JUXTAPOSITION**

Indica una **giustapposizione o confronto**, spesso persone o oggetti.

## Obiettivo dello studio

L'obiettivo dello studio è valutare se i Large Language Models, come BERT, siano in grado di rappresentare, almeno in parte, l'unità fondamentale della **Construction Grammar**: la **costruzione**.

In particolare, il case study si concentra sulla costruzione inglese **NPN** instanziata da *to*, scelta per la sua **polisemia** e i molteplici significati contestuali.

## **Accedere alla conoscenza di un LLM**

---

## LLMs come Black Box

I Large Language Models ricevono input e producono output,  
ma il loro funzionamento interno rimane in gran parte opaco.

- **Non osservabile:** non possiamo seguire direttamente come le reti trasformano l'informazione.
- **Non interpretabile:** le rappresentazioni interne sono vettori numerici di grandi dimensioni.

### Non conosciamo

- Quali **rappresentazioni interne** costruiscono?
- Quali **pattern linguistici** apprendono?
- In che modo **utilizzano** queste conoscenze nei task?

# Struttura del Transformer

## Self-Attention

Consente al modello di pesare dinamicamente tutte le parti dell'input.



### Encoder

- Converte il testo di input in **rappresentazioni vettoriali contestuali**.

### Decoder

- Genera testo **un token alla volta**.

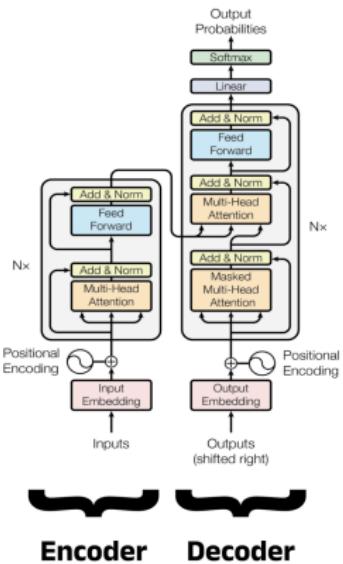
## Tipologie di modelli Transformer

**Encoder-only:** BERT, RoBERTa    **Decoder-only:** GPT, LLaMA

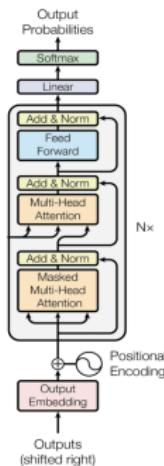
**Encoder-decoder:** T5, BART

# Tipologie di modelli Transformer

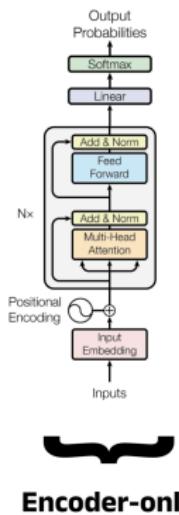
## Transformer



## GPT\*



## BERT\*



Modelli Encoder/Decoder

# BERT: modello encoder-only

## BERT e la costruzione NPN

Il case study del paper indaga la conoscenza che BERT possiede della costruzione **NPN** (Noun–Prep–Noun).

## Obiettivi di pre-training

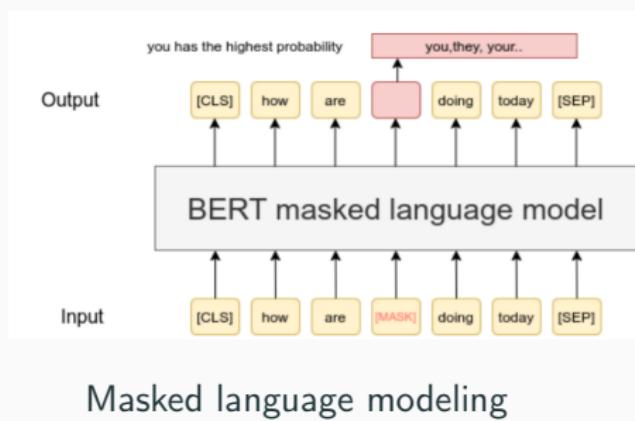
BERT viene addestrato su due task fondamentali:

- **Masked Language Modeling (MLM)**: si maschera un token e il modello deve **ricostruirlo a partire dal contesto**.
- **Next Sentence Prediction (NSP)**: il modello deve **capire se due frasi sono consecutive** nel testo originale.

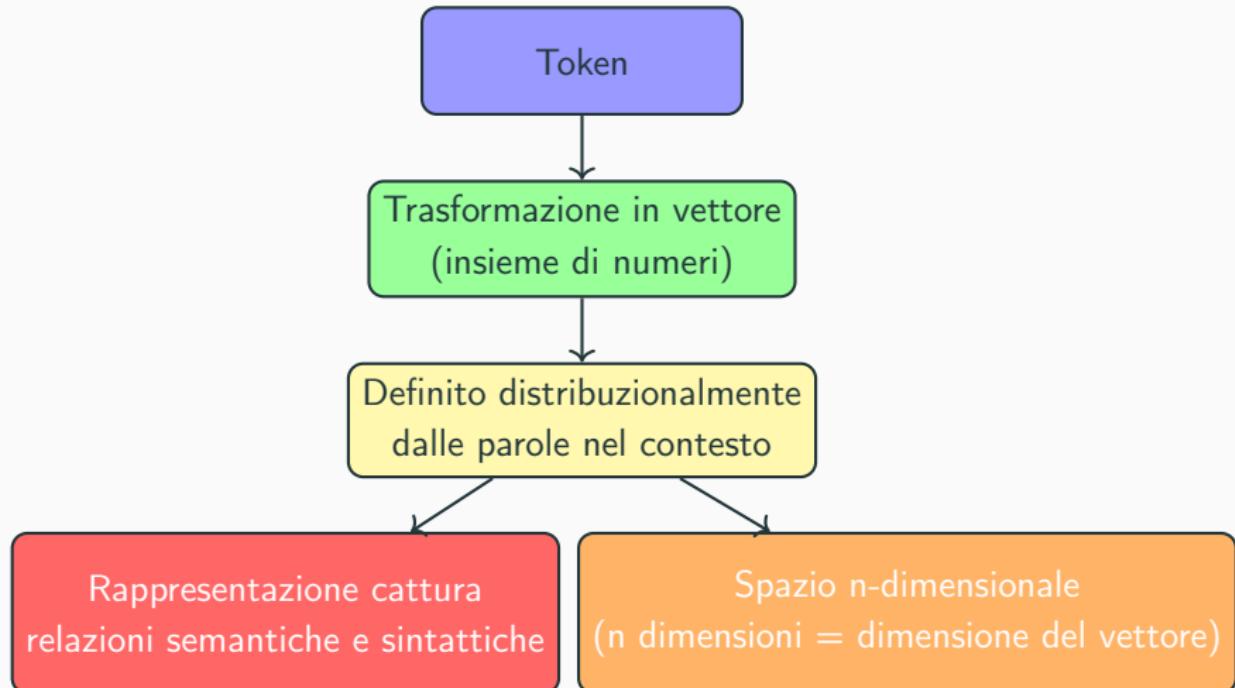
# Punto chiave: rappresentazioni profonde

## Punto chiave

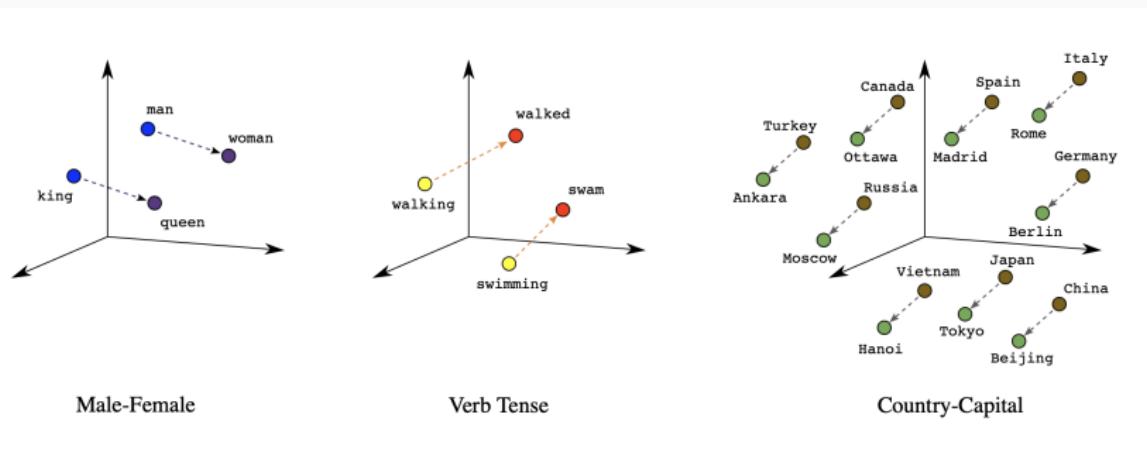
Lo scopo non è risolvere i task in sé, ma insegnare al modello a costruire rappresentazioni linguistiche profonde, cioè vettori contestuali ricchi di informazione sintattica e semantica.



# Che cos'è un vettore?



# Rappresentazione vettoriale



Embedding statici

# Rappresentazione vettoriale in BERT

Fasi principali:

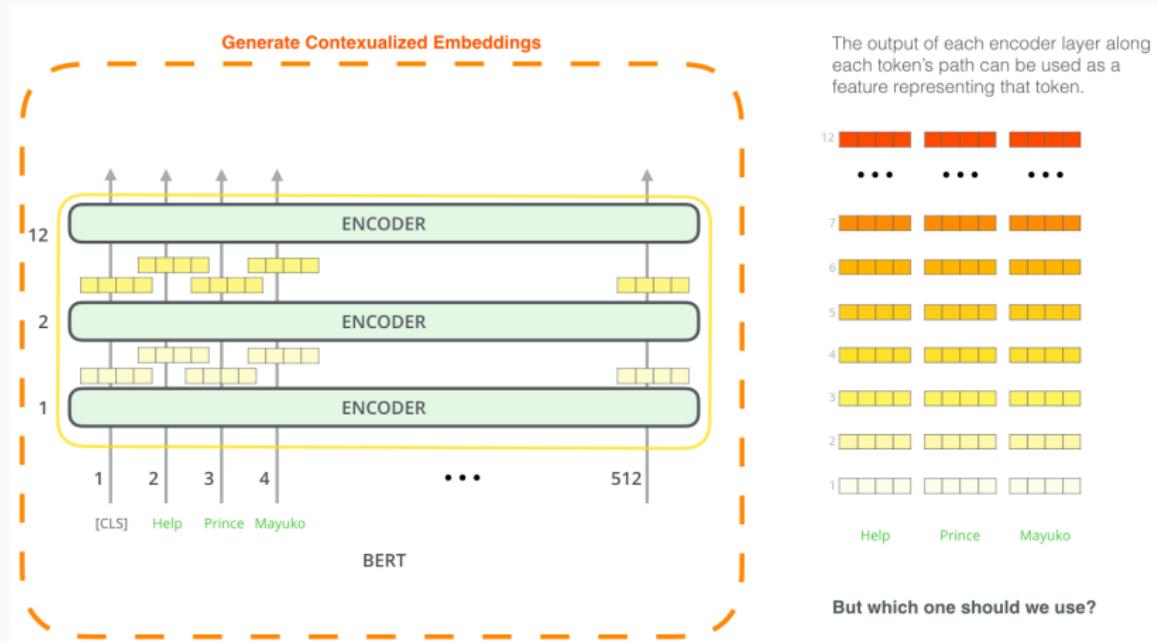
1. **Tokenizzazione**: ogni token viene convertito in un vettore iniziale.
2. **Previsione [MASK]**: cattura informazione linguistica contestuale, propagata attraverso i layer.

Layer dopo layer, il vettore acquisisce informazioni sempre più accurate.

3. **Aggiornamento pesi**: predizione su [MASK]  
Confronto con token corretto (LOSS)  
Aggiustamento dei parametri (BACKPROPAGATION).

Vettori numerici (768d) risultanti dall'addestramento di BERT

# Rappresentazione vettoriale in BERT



Embedding contestuali

# BERT: embedding contestuali

La risoluzione del task **Masked Language Modeling (MLM)** disegna lo **spazio vettoriale**.

- **Rappresentazione contestuale:** ogni token ha un vettore diverso a seconda della frase in cui compare.
- **Contesto bidirezionale:** il modello vede tutti i token della frase e aggiorna gli embedding a ogni layer.

# Probing

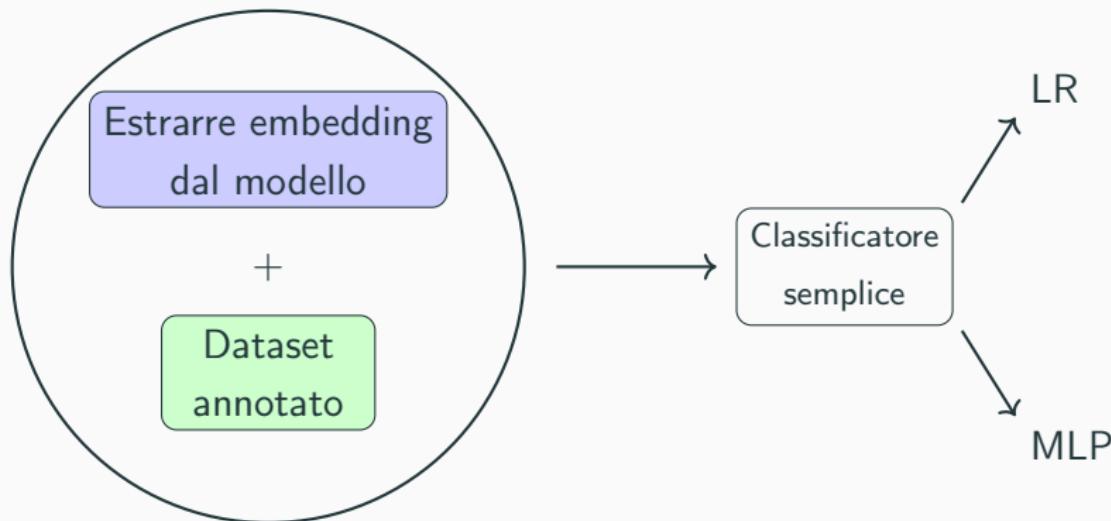
---

# Probing: accesso alle informazioni linguistiche

Vettori numerici complessi → non interpretabili

## Probing

Esaminare i **vettori** per analizzare se contengono determinate conoscenze linguistiche



# Che cos'è un classificatore?

**Definizione:** un classificatore è un modello che assegna un'istanza (nel nostro caso un embedding contestuale) a una **classe** tra un insieme finito di categorie.

**Obiettivo:**

$$f : x \mapsto y \in \{1, \dots, K\}$$

**Fase di Apprendimento (Training)**

Il modello viene addestrato su un **training set**, cioè esempi etichettati:

$$(x_i, y_i)$$

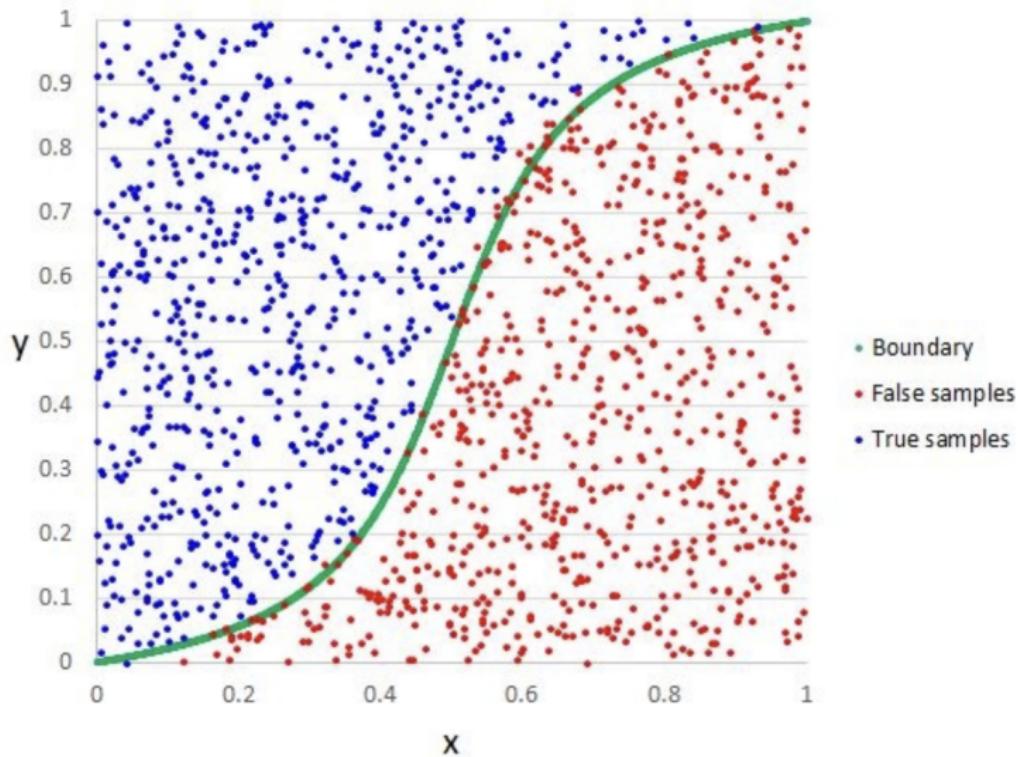
per stimare i parametri che distinguono le classi.

**Fase di Valutazione (Test)**

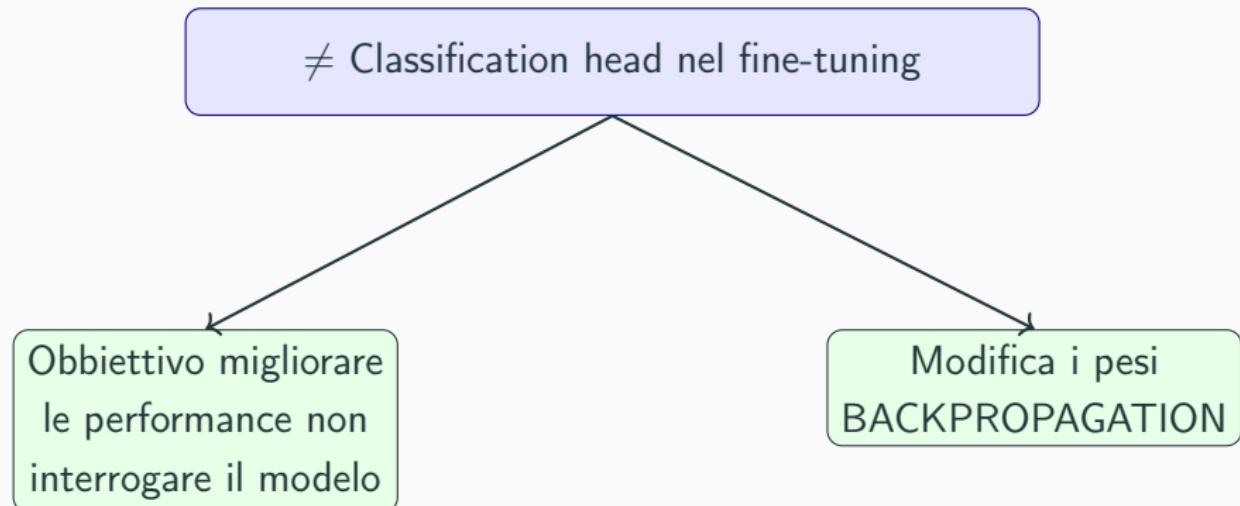
Un **test set** separato misura quanto il modello generalizza su dati mai visti.

# Logistic Regression

Logistic Regression Example



## Differenze rispetto al fine-tuning



## Interpretare le performance del classificatore

Per concludere che le buone performance del classificatore riflettono realmente l'informazione linguistica contenuta nei vettori, è necessario:

- Analizzare le performance degli **embedding per tutti i layer** del modello. → Ci attendiamo incremento e differenza.
- Stabilire una **baseline significativa** per confronto.
- Confrontare i risultati con un **control classifier**.

## II Dataset

---

## Il Dataset

- Estrazione da COCA (Corpus of Contemporary American English)
- Eliminazione dei casi PNPN
- Identificazione dei distrattori
- Annotazione per tutte le istanze delle etichette semantiche

# Affidabilità dell'annotazione

## Doppia annotazione

- Annotato **25%** del dataset
- Accordo grezzo: **84%**
- Cohen's kappa: **0.754** (accordo forte)

## Dimensioni del dataset

- **6599** istanze totali (N-to-N)
- **1885** istanze con doppia annotazione

# Near Minimal Pairs e Distrattori

## NtoN distractors

Oltre alle reali istanze della costruzione NtoN, il corpus contiene anche pattern superficiali **Noun + to + Noun** che non sono costruzioni NtoN. Derivano da contesti sintattici diversi (es. verbi che reggono un oggetto e una PP con *to*): *stick plastic to plastic, time to time travel*, ecc.

Questi casi non esprimono il significato della costruzione ma forniscono utili **eempi negativi** per testare se il modello.

## Near minimal pairs

Poiché condividono la stessa forma superficiale dei veri NtoN, questi distrattori costituiscono **near minimal pairs**: frasi grammaticali, naturali, quasi identiche in superficie, ma con **struttura e significato diversi**.

## Dataset

Nel case study **456** near minimal pairs come distrattori dal corpus.

## Training e Test set

---

## Split training test set

### Evitare overfitting

Max 20 occorrenze per lemma per evitare overfitting e ridurre la sproporzione tra lemmi altamente frequenti e lemmi rari.

### Controllo della generalizzazione

Generazione split casuali di train/test basati sul lemma del nome presente nella costruzione NtoN, in modo che nessun lemma compaia sia nel training set sia nel test set.

### Bilanciamento training

Poiché il numero di distrattori è significativamente inferiore rispetto alle istanze della costruzione, per bilanciare le categorie nel training set è stato utilizzato l'80% dei distrattori, abbinato allo stesso numero di costruzioni. Il test set è quindi composto dal restante 20% dei distrattori, insieme a tutte le costruzioni eccedenti quelle usate per il training.

## Task 1: Identificazione

---

## Task 1: Identificazione

### **Definizione del task**

Distinguere le istanze autentiche della costruzione NtoN dagli esempi negativi del corrispondente pattern distrattore.

### **Control classifier**

Le etichette vengono randomizzate e assegnate in modo deterministico al **word type**.

Le performance dovrebbero attestarsi near chance.

### **Non-contextual baseline (GloVe)**

Valuta la performance basata solo su informazioni lessicali, senza contesto (embedding statici).

Dovrebbero attendersi, in virtù dei campi semantici ricorrenti nella costruzione NtoN come espressioni temporali e parti del corpo, performance non trascurabili.

# A cosa serve la baseline?

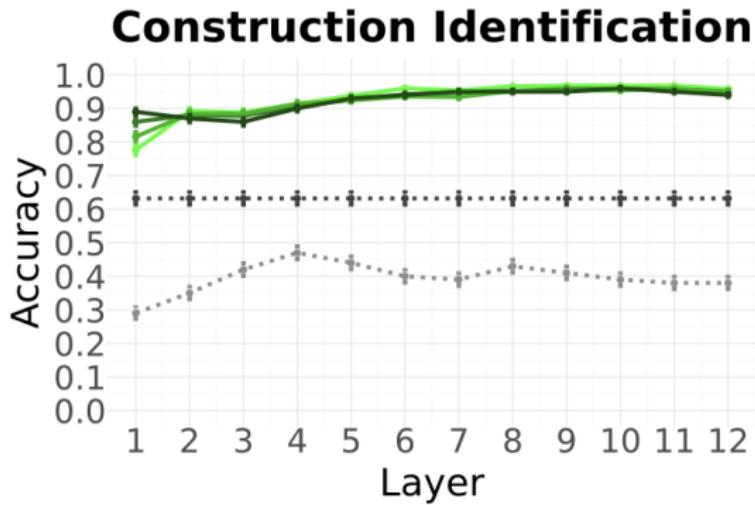
## **Control classifier**

Informa sulla bontà del classificatore, verificare che il classificatore non sia troppo potente.

## **Non-contextual baseline (GloVe)**

Tutto ciò che supera questa performance potrebbe essere attribuito alle informazioni aggiuntive catturate da BERT attraverso il significato contestuale.

## Task 1: Identificazione



This shows that overall, the probing classifier seems to be picking up on some sort of information in BERT which can reliably distinguish the NtoN construction from its near minimal pair NtoN distractor counterparts, beyond what is possible through lexical semantic clues alone.

## **Task 2: Identificazione (perturbando l'ordine delle parole)**

---

## Task 2: Perturbing Word Order

### Obiettivo

- Testare la **robustezza** del classificatore BERT-based.
- Verificare se distingue la **vera costruzione NtoN** da frasi con ordine delle parole **alterato artificialmente**.

### Idea di base

- Se il modello si basa troppo su **indizi lessicali**, classificherà come positive anche frasi non-NtoN con gli stessi nomi.
- Se è sensibile al **pattern N + to + N**, riconoscerà occorrenze perturbate da reali istanze della costruzione.

## Task 2: Perturbing Word Order

### Metodo

- Non viene riaddestrato il probe: vede solo N + to + N corretti.
- Viene manipolato il test set creando 4 ordini delle parole perturbati.

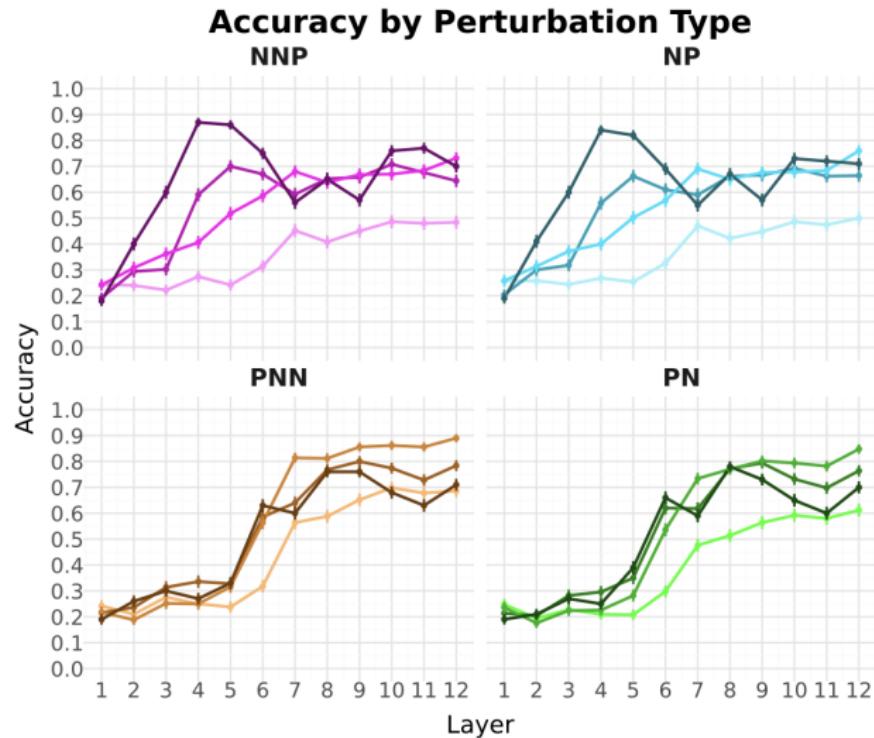
### Tipi di perturbazioni

- PNN: to + N + N
- PN: to + N
- NP: N + to
- NNP: N + N + to

### Scopo finale

- Valutare se il classificatore è sensibile alla **forma** della costruzione.

## Task 2: Perturbing Word Order



## Task 3: Disambiguazione semantica

---

## Task 3: Disambiguazione semantica

**Obiettivo:** Analizzare le etichette semantiche della costruzione NtoN.

- La performance del classificatore è alta nel distinguere NtoN da pattern simili.
- La costruzione NtoN è **ambigua** e può avere significati diversi a seconda del contesto.
- Due significati principali:
  - **SUCCESSION**
  - **JUXTAPOSITION**

## Task 3: Disambiguazione semantica

### Caratteristiche dei sottotipi:

- **SUCCESSION**: frequente con nomi spaziotemporali (es. giorno per giorno, costa a costa)
- **JUXTAPOSITION**: frequente con parti del corpo o esseri umani (es. faccia a faccia, amico ad amico)
- Il significato del sostantivo non è determinante: alcune occorrenze assumono il significato meno comune a seconda del contesto.
- Esistono sostantivi rari per cui non è chiaro quale sottotipo sia più comune.

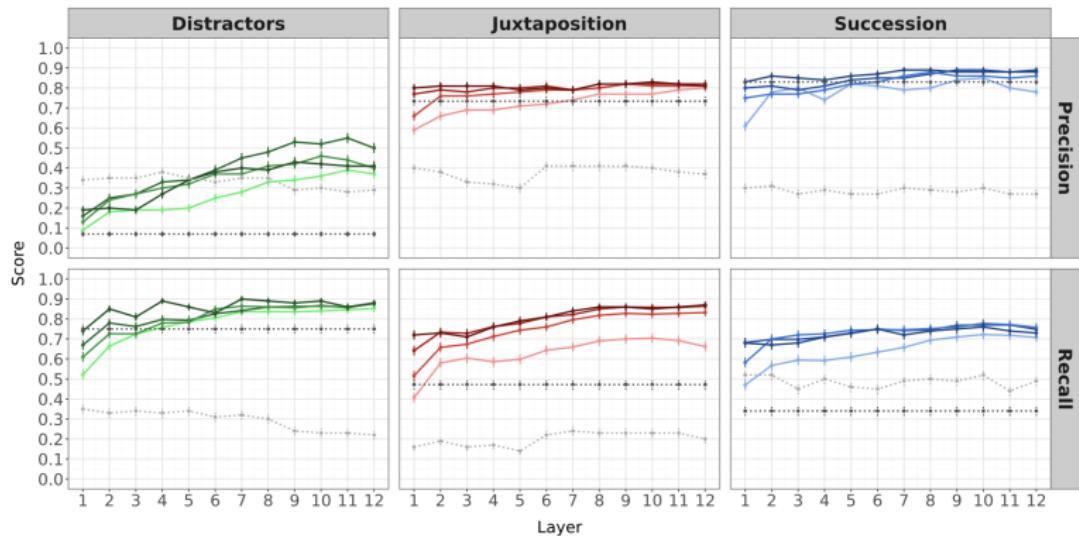
## Task 3: Disambiguazione semantica

### Metodologia:

- Classificatore per distinguere i sottotipi semantici:
  - SUCCESSION
  - JUXTAPOSITION
  - Non-esempi (pattern distrattori)
- Baseline = embedding statici
- Classificatori di controllo: etichette casuali assegnate a ciascun lemma (Hewitt & Liang, 2019)
- Se le probe sono selettive, i classificatori di controllo dovrebbero ottenere 33% di accuratezza.

# ask 3: Disambiguazione semantica

NPN Precision & Recall by Semantic Subtype



## Conclusioni principali

- Creato un nuovo dataset di istanze NtoN estratte da COCA.
- Addestrato un *linear probe* per distinguere la costruzione NtoN dai suoi quasi-minimal pair e da versioni con ordine delle parole perturbato.
- I probe mostrano alta accuratezza già nei layer medi di BERT e riconoscono in modo robusto la costruzione anche con pochi esempi di training.
- BERT è in grado di disambiguare i principali sottotipi semantici (SUCCESSION vs. JUXTAPOSITION)
- Le performance restano stabili anche con forti riduzioni dei dati: indizio che la conoscenza costruzionale è codificata latentamente nel modello.
- Evidenza a supporto dell'ipotesi che gli LLMs acquisiscano informazioni costruzionali.