

# Modulo 2: Rappresentazione del Testo e Word Embeddings

Corso di Natural Language Processing

# Indice dei Contenuti

- Introduzione alla rappresentazione del testo
- Rappresentazioni tradizionali del testo
- Word Embeddings: rappresentare le parole come vettori
- Visualizzazione e interpretazione dei word embeddings
- Applicazioni dei word embeddings
- Sentence Embeddings e Document Embeddings
- Sfide e limitazioni dei word embeddings
- Implementazione pratica dei word embeddings
- Conclusione

# Introduzione alla rappresentazione del testo

# Cos'è la rappresentazione del testo? 🤔

- Processo di **trasformazione** del linguaggio naturale in un formato elaborabile dai computer
- Passaggio fondamentale: da testo comprensibile agli umani a rappresentazioni numeriche/vettoriali
- I computer non "comprendono" naturalmente il linguaggio umano

## Importanza della rappresentazione

- La **qualità della rappresentazione** influenza direttamente l'efficacia dei sistemi NLP
- Una rappresentazione inadeguata limita anche gli algoritmi più sofisticati
- Una rappresentazione efficace permette buoni risultati anche con modelli semplici

"La rappresentazione è più importante dell'algoritmo" - dominio dell'apprendimento automatico

# Rappresentazioni tradizionali del testo

# One-Hot Encoding ⚡

- Ogni parola è un vettore con dimensione = dimensione del vocabolario
- Un 1 nella posizione corrispondente alla parola, 0 in tutte le altre

## Limitazioni:

- Alta dimensionalità (vettori enormi per vocabolari realistici)
- Nessuna informazione semantica (tutte le parole equidistanti)
- Nessuna generalizzazione

# Bag-of-Words (BoW)

- Rappresenta un documento come vettore di conteggi di parole
- Ignora l'ordine delle parole, considera solo la frequenza

## Esempio:

"Il gatto insegue il topo" → [2, 1, 1, 1, 0, 0, ...]  
(il, gatto, insegue, topo, ...)

## Limitazioni:

- Perdita dell'ordine delle parole
- Perdita di contesto
- Alta dimensionalità



# Term Frequency-Inverse Document Frequency (TF-IDF)

Formula:  $TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$

- **TF(t, d)**: frequenza del termine t nel documento d
- **IDF(t, D)**: logaritmo del rapporto tra numero totale di documenti e numero di documenti contenenti il termine t

## Vantaggi:

- Migliore capacità discriminativa rispetto a BoW
- Penalizza parole comuni e valorizza parole distintive

## N-grams

- Estensione del BoW che considera sequenze di N parole consecutive
- Cattura parzialmente l'ordine locale delle parole

### Esempio (bi-grams):

"Il gatto insegue il topo" → ["il gatto", "gatto insegue", "insegue il", "il topo"]

### Limitazioni:

- Dimensionalità aumenta esponenzialmente con N
- Sparsità dei dati

## Applicazione Aziendale: Analisi dei Brevetti con TF-IDF

Le aziende tecnologiche utilizzano TF-IDF per analizzare brevetti:

- Estrazione dei termini più distintivi da migliaia di documenti brevettuali
- Identificazione di tecnologie emergenti in specifici settori
- Monitoraggio delle attività di brevettazione dei concorrenti
- Valutazione di potenziali acquisizioni basata su portfolio brevetti

**Esempio:** IBM utilizza analisi TF-IDF avanzata per monitorare il panorama brevettuale in settori strategici come l'intelligenza artificiale

# Word Embeddings: Rappresentare le Parole come Vettori

## Cos'è un word embedding? 🔍

- Rappresentazione vettoriale di una parola in uno spazio multidimensionale continuo
- Ogni parola → vettore di numeri reali (tipicamente 100-300 dimensioni)
- La posizione nello spazio vettoriale cattura relazioni semantiche e sintattiche

### Differenze dalle rappresentazioni one-hot:

- **Densi** vs sparsi
- **Semanticamente significativi** vs arbitrari
- **Dimensionalità ridotta** vs alta dimensionalità

## Proprietà sorprendenti dei word embeddings ✨

I word embeddings catturano analogie e relazioni semantiche attraverso operazioni vettoriali:

$$\text{vector}(\text{"re"}) - \text{vector}(\text{"uomo"}) + \text{vector}(\text{"donna"}) \approx \text{vector}(\text{"regina"})$$

Questa proprietà emerge naturalmente durante l'addestramento!

# Principi di funzionamento

- Basati sull'**ipotesi distribuzionale** della semantica:
  - ▮ "You shall know a word by the company it keeps" (J.R. Firth)
- Parole che appaiono in contesti simili tendono ad avere significati simili
- Addestrati su obiettivi predittivi:
  - Predire una parola dato il suo contesto (CBOW)
  - Predire il contesto data una parola (Skip-gram)
  - Predire la probabilità di co-occorrenza (GloVe)

# Word2Vec

Sviluppato da Google nel 2013, due varianti principali:

- **Continuous Bag of Words (CBOW):**  
Predice una parola target dato il contesto circostante
- **Skip-gram:**  
Predice il contesto circostante data una parola target

Utilizza una rete neurale shallow con un singolo strato nascosto.



# GloVe (Global Vectors)

Sviluppato da Stanford nel 2014:

- Combina vantaggi dei metodi basati su contesto locale con statistiche globali di co-occorrenza
- Costruisce una matrice di co-occorrenza delle parole
- Addestra un modello per predire il logaritmo delle probabilità di co-occorrenza

## Vantaggi:

- Cattura sia informazioni locali che globali
- Migliore performance su alcune relazioni semantiche

# FastText

Sviluppato da Facebook AI Research nel 2016:

- Estende Word2Vec considerando i caratteri n-gram all'interno delle parole
- Apprende vettori per frammenti di parole (n-gram di caratteri)
- Rappresenta una parola come somma dei vettori dei suoi n-gram

## Vantaggi:

- Gestione di parole fuori vocabolario (OOV)
- Migliore per lingue morfologicamente ricche (italiano, tedesco, finlandese)

# Visualizzazione e interpretazione dei word embeddings

# Tecniche di riduzione della dimensionalità

Per visualizzare gli embeddings (da 100-300D a 2-3D):

- **Principal Component Analysis (PCA):**  
Identifica direzioni di massima varianza
- **t-Distributed Stochastic Neighbor Embedding (t-SNE):**  
Preserva relazioni di vicinanza locale
- **Uniform Manifold Approximation and Projection (UMAP):**  
Preserva sia struttura locale che globale

## Interpretazione delle dimensioni

- Le singole dimensioni non hanno generalmente un'interpretazione semantica chiara
- È possibile identificare **direzioni significative**:
  - Direzione di genere: "uomo"-"donna", "re"-"regina"
  - Direzione di formalità/informalità
  - Direzione di positività/negatività

## Valutazione degli embeddings

- **Test di analogia:** "a sta a b come c sta a ?"
- **Test di similarità:** correlazione tra similarità coseno e giudizi umani
- **Valutazione estrinseca:** performance in compiti downstream (classificazione, NER)

Diversi embeddings possono eccellere in diversi tipi di valutazione

# Applicazione Aziendale: Ricerca Semantica nei Documenti Legali



Gli studi legali utilizzano word embeddings per:

- Ricerca di concetti legali simili anche con terminologia diversa
- Identificazione di precedenti rilevanti basata sulla similarità semantica
- Organizzazione automatica di grandi archivi di documenti legali
- Suggerimento di clausole contrattuali pertinenti
- Vector DB Creazione del contesto nei Retrieval-Augmented Generation

**Esempio:** Analizzare contratti e documenti legali, estraendo informazioni rilevanti da migliaia di documenti in una frazione del tempo

# Applicazioni dei word embeddings



# Miglioramento della ricerca semantica

- Comprensione del significato delle query, non solo parole esatte
- Risultati pertinenti anche con sinonimi o termini correlati

## Applicazioni:

- **E-commerce:** ricerca di prodotti con descrizioni generiche
- **Editoria digitale:** trovare articoli rilevanti basati su concetti

## Categorizzazione avanzata dei documenti

- Maggiore accuratezza nella classificazione dei testi
- Categorizzazione corretta anche con terminologie diverse

### Applicazioni:

- **Settore assicurativo:** categorizzazione automatica delle richieste di risarcimento
- **Analisi social media:** categorizzazione di post per temi, sentiment o intenti

## Sistemi di raccomandazione basati su contenuto 👍

- Rappresentazioni vettoriali di documenti, prodotti o contenuti
- Calcolo di similarità semantiche per raccomandazioni pertinenti

### Applicazioni:

- **Media e intrattenimento:** analisi di descrizioni e metadati per raccomandazioni
- **Recruiting:** matching tra curriculum e offerte di lavoro

## Analisi del sentiment più accurata 😊😡

- Migliore generalizzazione per riconoscere il tono emotivo
- Riconoscimento del sentiment anche con parole non presenti nel dataset

### Applicazioni:

- **Settore finanziario:** analisi di notizie economiche e social media
- **Ristorazione e ospitalità:** insights granulari dalle recensioni dei clienti

## Rilevamento di temi emergenti

- Identificazione di temi o problemi emergenti nelle comunicazioni
- Analisi di cluster di parole nello spazio degli embeddings

### Applicazioni:

- **Settore farmaceutico:** identificazione precoce di effetti collaterali
- **Marketing:** identificazione di nuove tendenze di consumo

# Sentence Embeddings e Document Embeddings

# Da word embeddings a sentence embeddings

Approcci semplici:

- Media o somma pesata dei word embeddings delle parole

Approcci più sofisticati:

- **Smooth Inverse Frequency (SIF)**: media pesata con pesi inversamente proporzionali alla frequenza
- **Doc2Vec**: estensione di Word2Vec per documenti
- **Universal Sentence Encoder (USE)**: modello pre-addestrato per similarità semantica

# Modelli avanzati per sentence embeddings 🚀

Modelli basati su architetture Transformer:

- **BERT Sentence Embeddings**: rappresentazione del token [CLS] o media dei token
- **Sentence-BERT (SBERT)**: modificazione di BERT ottimizzata per sentence embeddings
- **SimCSE**: utilizza tecniche di apprendimento contrastivo

Questi modelli catturano molto meglio il significato complessivo delle frasi



# Applicazioni dei sentence embeddings

- Clustering semantico di documenti
- Rilevamento di duplicati e near-duplicates
- Sistemi di risposta a domande
- Riassunto estrattivo

## Settori di applicazione:

- Bancario: analisi e categorizzazione di comunicazioni con clienti
- Ricerca accademica: scoperta di letteratura rilevante

## Sfide e limitazioni dei word embeddings ⚠

## Polisemia e ambiguità

- Word embeddings tradizionali assegnano un **singolo vettore** a ogni parola
- Problematico per parole polisemiche con significati diversi in contesti diversi

### Esempio:

"calcio" → sport, elemento chimico, azione di colpire

### Impatto:

- Rappresentazione "media" non ottimale per nessun significato specifico
- Problemi in applicazioni come traduzione automatica

## Bias e stereotipi

- Gli embeddings ereditano bias presenti nei dati di addestramento
- Associazioni problematiche: professioni-genere, etnia-attributi, ecc.

### Impatto:

- Propagazione e amplificazione di bias nelle applicazioni
- Risultati potenzialmente discriminatori (es. screening CV)

## Dipendenza dalla qualità e quantità dei dati

- Qualità degli embeddings dipende fortemente dai dati di addestramento
- Embeddings generici potrebbero non catturare terminologia di dominio

### Problematiche:

- Termini specialistici o acronimi mal rappresentati
- Lingue con risorse limitate hanno embeddings di qualità inferiore

## Mancanza di comprensione profonda 🧠

- Catturano relazioni semantiche superficiali, ma non vera "comprensione"
- Non catturano relazioni logiche complesse o conoscenza del mondo reale

### **Esempio:**

Riconoscere che "Parigi" e "Francia" sono correlati  $\neq$  comprendere che Parigi è la capitale della Francia

## Evoluzione verso embeddings contestuali

Per superare queste limitazioni → embeddings contestuali:

- La rappresentazione di una parola dipende dal contesto specifico
- Modelli come ELMo, BERT e GPT generano embeddings dinamici

### Esempio:

In BERT, "calcio" ha rappresentazioni diverse in:

- "lo gioco a calcio"
- "Giocando mi hanno dato un calcio"

# Implementazione pratica dei word embeddings



## Scelta del modello di embedding

Fattori da considerare:

- **Dominio applicativo:** embeddings specialistici vs generici
- **Lingua:** disponibilità e qualità per lingue diverse dall'inglese
- **Risorse computazionali:** complessità del modello
- **Compito specifico:** alcuni modelli performano meglio per certi compiti

# Embeddings pre-addestrati vs addestramento custom

## Pre-addestrati:

- Word2Vec su Google News
- GloVe su Wikipedia e Gigaword
- FastText su Wikipedia, Common Crawl

## Addestramento custom vantaggioso quando:

- Terminologia di dominio molto specifica
- Grandi quantità di dati proprietari
- Necessità di ottimizzazione per compiti specifici

## Integrazione nelle pipeline NLP

Modalità di integrazione:

- **Feature engineering:** input per modelli ML tradizionali
- **Layer di embedding in reti neurali:** inizializzazione con embeddings pre-addestrati
- **Calcolo di similarità:** similarità coseno per ricerca o clustering

## Tecniche di debiasing

Per mitigare i bias negli embeddings:

- **Hard debiasing:** neutralizza esplicitamente componenti di bias
- **Soft debiasing:** regolarizzazione durante l'addestramento
- **Augmentation dei dati:** arricchimento con esempi che contrastano stereotipi

**Conclusione** 

## Riepilogo del Modulo

- La rappresentazione del testo è fondamentale per l'NLP
- Le tecniche tradizionali (BoW, TF-IDF) hanno limitazioni significative
- I word embeddings rappresentano le parole come vettori densi che catturano relazioni semantiche
- Word2Vec, GloVe e FastText sono le principali tecniche, ciascuna con punti di forza specifici
- I sentence embeddings estendono il concetto oltre le singole parole
- Le applicazioni spaziano dalla ricerca semantica all'analisi del sentiment
- Nonostante le limitazioni, i word embeddings hanno trasformato l'NLP

## Evoluzione e futuro 🧙

- I word embeddings hanno posto le basi per i modelli contestuali più avanzati
- L'evoluzione verso embeddings contestuali (BERT, GPT) supera molte limitazioni
- I principi fondamentali della rappresentazione vettoriale rimangono centrali

Nel prossimo modulo: classificazione del testo e analisi del sentiment

## Riferimenti e Approfondimenti

- Mikolov, T., et al. (2013). Distributed Representations of Words and Phrases and their Compositionality.
- Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation.
- Bojanowski, P., et al. (2017). Enriching Word Vectors with Subword Information.
- Bolukbasi, T., et al. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.