(De-)indicizzazione e diritto all'oblio al tempo dei *Large Language Model*

Serena Nicolazzo

Ricercatrice, DISIT, Università del Piemonte Orientale
ORCID 0000-0003-2719-9526

Anna Sapienza

Ricercatrice, DISIT, Università del Piemonte Orientale
ORCID 0000-0002-0842-7987

Salvatore Vilella

Assegnista di ricerca, DISIT, Università del Piemonte Orientale
ORCID 0000-0002-7666-0318

Giancarlo Ruffo

Professore ordinario di Informatica, DISIT, Università del Piemonte Orientale
ORCID 0000-0002-3407-9234

DOI: 10.54103/milanoup.273.c591

Abstract: Nell'era digitale, la questione dell'oblio si è affermata come una problematica di crescente rilievo, in particolare in relazione alla gestione dei dati personali e alla loro accessibilità online. Il diritto all'oblio (Right to Be Forgotten, RTBF) consente ai cittadini di richiedere la rimozione dal pubblico accesso di informazioni obsolete o potenzialmente dannose. Tuttavia, l'attuazione concreta di tale diritto pone notevoli difficoltà tecniche per i motori di ricerca. Il presente contributo si propone di offrire a un pubblico non specialista un'introduzione ai concetti fondamentali dell'information retrieval (IR) e della deindicizzazione, strumenti cruciali per comprendere in che modo i motori di ricerca possano effettivamente "dimenticare" determinati contenuti. Verranno illustrati i principali modelli di IR, tra cui il modello booleano, i modelli probabilistici, quelli basati su spazi vettoriali e su embedding, con particolare attenzione al ruolo svolto dai Large Language Models (LLM) nel migliorare le capacità di elaborazione e analisi dei dati. Si discuterà anche di come gli LLM, oltre agli indubbi vantaggi che hanno introdotto, rendano ancora più difficile la rimozione di informazioni eventualmente acquisite dal modello durante la sua fase di addestramento. Infine, si introdurranno i

principi del cosiddetto "Machine Unlearning", una metodologia per rimuovere dati dai modelli di apprendimento automatico, senza per questo far degradare (eccessivamente) le proprietà dei modelli stessi. Questa metodologia, che sta attirando un interesse crescente negli ultimi anni nella comunità scientifica di riferimento, potrebbe diventare lo strumento principale per consentire l'applicazione del diritto all'oblio anche in presenza di LLM. Grazie a questa panoramica, si intende evidenziare le complessità insite nel bilanciamento tra la tutela del diritto all'oblio e le sfide tecniche che i motori di ricerca e gli LLM si trovano ad affrontare nella gestione della visibilità e della disponibilità delle informazioni online.

In the digital age, the issue of forgetting has become a growing concern, particularly in relation to the management of personal data and its online accessibility. The Right to Be Forgotten (RTBF) allows citizens to request the removal of obsolete or potentially harmful information from public access. However, the practical implementation of this right poses significant technical challenges for search engines. This paper aims to provide a non-specialist audience with an introduction to the fundamental concepts of information retrieval (IR) and de-indexing, crucial tools for understanding how search engines can effectively "forget" certain content. The main IR models will be illustrated, including the Boolean model, probabilistic models, models based on vector spaces, and those based on embeddings, with particular attention to the role played by Large Language Models (LLM) in improving data processing and analysis capabilities. We will also discuss how LLMs, in addition to their undeniable advantages, make it even more difficult to remove information acquired by the model during its training phase. Finally, we will introduce the principles of so-called "Machine Unlearning," a methodology for removing data from machine learning models without (excessively) degrading the models' properties. This methodology, which has attracted growing interest in recent years in the relevant scientific community, could become the primary tool for enabling the application of the right to be forgotten even in the presence of LLMs. Through this overview, we intend to highlight the complexities inherent in balancing the protection of the right to be forgotten with the technical challenges that search engines and LLMs face in managing the visibility and availability of information online.

Parole chiave: Diritto all'Oblio; Recupero delle informazioni; Grandi modelli linguistici; Disapprendimento automatico.

Keywords: *RTBF; Information Retrieval; LLM; Machine Unlearning.*

1. Introduzione

La *de-indicizzazione* è il processo di rimozione di specifici contenuti o link dall'indice di un motore di ricerca, impedendo che appaiano nei risultati di ricerca. A differenza dell'eliminazione di contenuti da un sito web, la de-indicizzazione non cancella le informazioni dalla loro fonte originale; piuttosto, assicura che tali informazioni siano meno accessibili tramite i motori di ricerca. Questo processo è spesso richiesto da individui che cercano di limitare la visibilità di determinate informazioni online che potrebbero essere obsolete, irrilevanti o dannose per la loro reputazione.

La de-indicizzazione è strettamente associata al *diritto all'oblio (RTBF)*, un principio legale che consente agli individui di richiedere la rimozione dei dati personali dall'accesso pubblico quando non servono più a uno scopo legittimo. Originario dell'Europa con la storica sentenza Google Spain SL, Google Inc. contro Agencia Española de Protecciòn de Datos nel 2014, il RTBF è ora una pietra miliare dei diritti alla privacy nell'era digitale, definito in legislazioni come il Regolamento Generale sulla Protezione dei Dati (GDPR), Articolo 17. La de-indicizzazione serve come meccanismo pratico per far rispettare il RTBF, bilanciando il diritto individuale alla privacy e il diritto del pubblico all'informazione.

Tuttavia, per comprendere appieno le dinamiche, le possibilità e le sfide tecniche dietro la de-indicizzazione del Web, è importante avere una conoscenza dei principi di base dell'indicizzazione e, più in generale, del *recupero delle informazioni (Information Retrieval, IR)*. L'obiettivo del presente documento è fornire al lettore una panoramica generale e tecnica dell'IR, consentendo riflessioni e un'analisi delle implicazioni della deindicizzazione e del RTBF sugli algoritmi dei motori di ricerca e sulla privacy dei dati.

Il presente documento è organizzato come segue. Nella Sezione 2 forniamo una panoramica tecnica dei modelli di information retrieval più comuni, esaminando le basi dei modelli booleani, dei modelli probabilistici, dello spazio vettoriale e dei modelli basati su embedding. Introdurremo anche il lettore ai Large Language Models (LLM), che rappresentano l'ultima innovazione nell'IR e nel data mining testuale. Infine, nella Sezione 3 concludiamo con una discussione sulle tecniche di de-indicizzazione alla luce degli argomenti tecnici elaborati nelle sezioni precedenti. Questa sezione include anche un'introduzione al "Machine Unlearning" e a come questa metodologia possa essere usata per rimuovere alcune informazioni dagli LLM.

Se il lettore è particolarmente interessato all'IR, suggeriamo vivamente i seguenti riferimenti: [3,6,21,34]. Tuttavia, questo capitolo affronta in modo succinto alcuni degli aspetti più fondamentali delle principali tecnologie IR, sebbene il nostro obiettivo primario sia descrivere come i moderni motori IR gestiscono il modo in cui le informazioni possano essere "dimenticate".

2. Modelli di Information Retrieval

L'information retrieval (IR) è il processo di ricerca di informazioni contenute in grandi collezioni di dati non strutturati o semi-strutturati in risposta a query (interrogazioni) dell'utente. A differenza del recupero di dati strutturati nelle basi di dati, l'IR si occupa principalmente di collezioni di documenti ricchi di testo, come libri, articoli, pagine web, ed informazioni multimediali (aspetto di cui non ci occuperemo in questo capitolo). L'obiettivo di un sistema di IR è soddisfare le esigenze informative dell'utente fornendo risultati pertinenti e classificati per ordine di importanza. Questo campo si trova all'intersezione tra informatica, linguistica e scienze cognitive, attingendo a vari metodi per sviluppare algoritmi in grado di elaborare e interpretare il linguaggio umano, un compito intrinsecamente complesso.

Il processo fondamentale di recupero delle informazioni coinvolge diverse fasi, tra cui la pre-elaborazione dei dati (tokenizzazione[1], stemming[2], rimozione delle stop-word[3]), l'indicizzazione del contenuto per consentire ricerche veloci e la classificazione dei documenti in base alla loro rilevanza per una query. I moderni sistemi di IR utilizzano una varietà di tecniche, come l'espansione della query, il feedback di rilevanza e la ponderazione dei termini, per migliorare la qualità dei risultati di ricerca. Sempre più spesso, l'apprendimento automatico (ML) e l'elaborazione del linguaggio naturale (NLP) vengono integrati nei processi di IR, consentendo ai sistemi di gestire le query degli utenti in modo più intelligente e di adattarsi alle esigenze informative in evoluzione.

I modelli di IR, che forniscono le basi teoriche su come le informazioni vengono recuperate, possono essere suddivisi in tre categorie:

1. *Modelli Booleani*: Questi modelli trattano documenti e query come insiemi di termini e utilizzano criteri di corrispondenza rigorosi basati su operatori logici (AND, OR, NOT) per determinare la rilevanza. Sebbene semplici e intuitivi, i modelli booleani non hanno meccanismi per classificare i risultati o tenere conto delle corrispondenze parziali.
2. *Modelli a Spazio Vettoriale*: Questi modelli rappresentano documenti e query come vettori in uno spazio multidimensionale, dove ogni dimensione corrisponde a un termine. La rilevanza viene calcolata utilizzando misure come la similarità del coseno, che cattura il grado di allineamento tra i vettori di documento e query. Questo approccio consente la corrispondenza

1 Tokenizzazione: la suddivisione di un testo grezzo in unità più piccole, come parole o frasi, dette "token".

2 Stemming: il processo di riduzione della forma inflessa di una parola a un cosiddetto "tema", o forma radice, che spesso corrisponde al "lemma", come viene chiamato in linguistica.

3 Stop-word: parole estremamente frequenti nei testi, come gli articoli, preposizioni e congiunzioni, quindi difficilmente caratterizzanti i singoli documenti.

parziale e la classificazione, ma richiede una ponderazione attenta dei termini per essere efficace.

3. *Modelli Probabilistici*: I modelli probabilistici stimano la probabilità che un documento sia rilevante per una data query, basandosi su probabilità a priori e sulla presenza o assenza di termini. Un esempio preminente è il modello di rilevanza probabilistico, che è alla base di tecniche avanzate come BM25 [30].

Oltre a questi modelli fondamentali, i modelli linguistici profondi di grandi dimensioni, pre-addestrati e basati su transformer, come BERT [12], T5 [29] e GPT [28], si sono dimostrati efficaci per il recupero e la classificazione di passaggi di testo [13,18,24,36].

2.1 Modelli Booleani e Rappresentazioni dei Documenti

Il *modello di IR booleano* è un modello per il recupero delle informazioni in cui possiamo porre qualsiasi query sotto forma di espressione booleana di termini, ovvero in cui i termini sono combinati con gli operatori AND, OR e NOT. Nei modelli booleani, documenti e query sono rappresentati come insiemi di termini e, applicando uno o una combinazione degli operatori logici, il modello restituisce i documenti pertinenti ai termini specificati nella query.

Esistono molti modi possibili per esplorare un documento: il più intuitivo sarebbe una scansione lineare dell'intero documento per recuperare le parole selezionate. Tuttavia, questo è un processo estremamente costoso, in particolare per testi molto lunghi e query complesse. Come risposta a questo problema, si ricorre a diverse tecniche. Una possibile soluzione è quella di costruire una *Matrice Termine-Documento* (Term-Document Matrix - TDM): una rappresentazione tabellare utilizzata per catturare la frequenza o la presenza di termini in una collezione di documenti. In questa matrice ogni riga rappresenta un termine unico dal corpus, mentre ogni colonna corrisponde a un documento. I valori delle celle indicano la frequenza (o presenza binaria) di un termine in un documento.

Sia:

- C un corpus costituito da n documenti: D_1, D_2,..., D_n.
- V il vocabolario dei termini unici nel corpus, costituito da m termini: t_1, t_2, ..., t_m.

La TDM è una matrice $m \times n$ $\mathbf{A} = [a_{ij}]$, dove:

$$a_{ij} = \begin{cases} f(t_i, D_j), & \text{se il termine } t_i \text{ compare nel documento } D_j \\ 0, & \text{altrimenti} \end{cases}$$

Qui, $f(t_i, D_j)$ è una funzione che quantifica l'associazione tra il termine t_i e il documento D_j. Le scelte comuni per $f(t_i, D_j)$ includono:

1. **Frequenza del Termine (TF)**: $f(t_i, D_j)$ = conteggio di t_i in D_j.
2. **Rappresentazione Binaria**: $f(t_i, D_j) = 1$ se t_i compare in D_j; 0 altrimenti.
3. **TF-IDF (Frequenza del Termine-Frequenza Inversa del Documento)**: $f(t_i, D_j)$ = punteggio TF-IDF di t_i in D_j - quantità che definiremo in modo preciso più avanti.

Ad esempio, consideriamo un corpus con C = $\{D_1, D_2, D_3\}$ e vocabolario V = {"mela", "banana", "arancia"}:

- D_1 = "mela banana mela"
- D_2 = "banana arancia"
- D_3 = "arancia mela arancia"

La TDM **A** con le occorrenze dei termini è:

$$A = [\ 2\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 1\ \ 2\ ]$$

Trasformando un corpus di documenti in una matrice, si possono sfruttare tecniche algebriche per analizzarli in modo più efficiente; inoltre, in un corpus reale la matrice sarà probabilmente sparsa, migliorando ulteriormente l'efficienza computazionale.

Un altro metodo molto comune per indicizzare i documenti è l'*indice inverso* (inverted index) che offre vantaggi significativi rispetto alla TDM in termini di efficienza dello spazio, velocità di elaborazione delle query, supporto per query complesse, facilità di aggiornamenti e scalabilità, rendendola la scelta preferita per la struttura dati nei moderni sistemi di recupero delle informazioni.

L'indice inverso associa ogni termine $t_i \in$ V a un insieme di identificatori di documento e, opzionalmente, alle posizioni in cui il termine appare. Formalmente, l'indice inverso è definito come:

$$\text{Index}(t_i) = \{(j, \text{P}_{ij}) \mid t_i \text{ compare in } D_j\},$$

dove j è l'indice di un documento D_j in C e P_{ij} è l'insieme delle posizioni in cui il termine t_i appare nel documento D_j.

Se consideriamo lo stesso corpus C e vocabolario V dell'esempio precedente:

- D_1 = "mela banana mela"
- D_2 = "banana arancia"
- D_3 = "arancia mela arancia"

L'indice inverso per questo corpus è:

– Index("mela") = {(1,{1,3}),(3,{2})}
– Index("banana") = {(1,{2}),(2,{1})}
– Index("arancia") = {(2,{2}),(3,{1,3})}

Qui:

- "mela" compare in D_1 alle posizioni {1,3} e in D_3 alla posizione {2}.
- "banana" compare in D_1 alla posizione {2} e in D_2 alla posizione {1}.
- "arancia" compare in D_2 alla posizione {2} e in D_3 alle posizioni {1,3}.

Come possiamo vedere, mentre la TDM associa direttamente i termini con valori numerici in ogni documento, l'indice inverso inverte la struttura, associando i termini ai documenti (e posizioni) in cui compaiono. Questa struttura è anche particolarmente utile per query di ricerca efficienti ed è un metodo estremamente comune per l'indicizzazione delle pagine web.

2.2 Limitazioni dei Modelli di Query Booleani

I modelli di query booleani, sebbene fondamentali nei sistemi di IR tradizionali, presentano diverse limitazioni:

- **Rigidità delle Query** – Le query booleane sono spesso troppo rigide in quanto corrispondono strettamente alle parole chiave specificate senza considerare la loro rilevanza o contesto all'interno dei documenti. Di conseguenza:
  - I documenti contenenti tutte le parole chiave vengono recuperati, indipendentemente dalle relazioni o dall'importanza di questi termini nel documento.
  - Documenti pertinenti che utilizzano sinonimi o espressioni parafrasate potrebbero essere persi.
- **Problema "Feast or Famine"** – Questo problema si verifica quando le query booleane recuperano o troppi documenti (un banchetto: feast) o nessuno (una carestia: famine). Nello specifico:
  - Query ampie producono un numero eccessivo di risultati, la maggior parte dei quali poco rilevanti.
  - Query ristrette escludono documenti potenzialmente rilevanti.

L'incorporazione della valutazione basata sulla prossimità nelle strutture di indice inverso pùo affrontare parzialmente questo problema classificando i documenti in base alla "vicinanza" ai termini della query.

- **Mancanza di Flessibilità -** I modelli booleani non tengono conto della somiglianza semantica o delle variazioni linguistiche, come:
  - **Sinonimi:** Ad esempio, "auto" e "macchina".
  - **Variazioni Morfologiche:** Ad esempio, "correre" e "correndo".

Tradizionalmente, ciò viene affrontato espandendo il vocabolario della query utilizzando tecniche come thesauri o stemming. Tuttavia, questi metodi spesso non riescono a catturare appieno le relazioni sfumate.

- **Risultati Binari** – I modelli booleani classificano i documenti come:
  - Rilevanti (se corrispondono precisamente alla query), o
  - Non rilevanti (se non corrispondono a nessuna parte della query).

Questa classificazione binaria non consente di creare una classifica dei documenti in base alla loro rilevanza: un punteggio di rilevanza compreso nell'intervallo reale [0,1] è senz'altro preferibile. I sistemi di IR avanzati spesso utilizzano meccanismi di valutazione della rilevanza del documento rispetto alla data query come la Term Frequency-Inverse Document Frequency (TF-IDF) o modelli di apprendimento automatico per superare questa limitazione.

2.3 Modelli a Spazio Vettoriale

Il *modello a spazio vettoriale (Vector Space Model - VSM)* è un altro possibile approccio per rappresentare e confrontare i dati testuali. In questo modello, sia i documenti che le query sono rappresentati come vettori in uno spazio ad alta dimensionalità, dove ogni dimensione corrisponde a un termine unico nel vocabolario. La rilevanza di un documento per una query è determinata dalla somiglianza tra i rispettivi vettori. Sia:

- $C = \{D_1, D_2, ..., D_n\}$ una collezione di n documenti;
- $V = \{t_1, t_2, ..., t_m\}$ il vocabolario dei termini unici nel corpus.

Ogni documento D_j è rappresentato come un vettore:

$$\mathbf{d}_j = [w_{1j}, w_{2j}, ..., w_{mj}]$$

dove w_{ij} è il peso del termine t_i nel documento D_j. I pesi dei termini, w_{ij}, svolgono un ruolo cruciale nel determinare l'efficacia del VSM.

Gli approcci comuni includono:

1. **Frequenza del Termine (TF)** – Riflette quante volte un termine compare in un documento.

$$w_{ij} = \text{TF}(t_i, D_j) = \text{Conteggio di } t_i \text{ in } D_j.$$

2. **TF-IDF (Frequenza del Termine-Frequenza Inversa del Documento)** – Bilancia la frequenza del termine con la rarità di un termine all'interno del corpus.

$$w_{ij} = \text{TF}(t_i, D_j) \times log\left(\frac{n}{DF(t_i)}\right),$$

dove $\text{DF}(t_i)$ è la frequenza del documento di t_i (cioè, il numero di documenti che contengono t_i), e n è il numero totale di documenti nella collezione. Il logaritmo viene applicato per smorzare l'importanza relativa di termini che occorrono molto frequentemente nella collezione.

Infine, la rilevanza di un documento per una query viene calcolata utilizzando misure di similarità. La più ampiamente utilizzata è la *similarità del coseno*, definita come:

$$\text{CosineSimilarity}(\mathbf{d}_j, \mathbf{q}) = \frac{dj \cdot q}{\|dj\| \|q\|},$$

dove:

- $\mathbf{d}_j \cdot \mathbf{q}$ è il prodotto scalare dei vettori,
- $\|\mathbf{d}_j\|$ e $\|\mathbf{q}\|$ sono le loro magnitudini, ovvero le lunghezze dei vettori in uno spazio multi-dimensionale[4].

4 In termini matematici, la magnitudine di un vettore **v** si calcola grazie alla radice quadrata della somma dei quadrati delle sue componenti: **v** = v12+ v22+ vV2 , dove V è la dimensione dello spazio vettoriale

Intuitivamente, la similarità del coseno misura l'angolo tra i vettori in uno spazio |V|-dimensionale, con angoli più piccoli che indicano una maggiore somiglianza tra il documento e la query.

2.4 Modelli Probabilistici

In questa sezione, parliamo brevemente dei modelli probabilistici in ambito di IR, che forniscono un quadro statistico per prevedere la probabilità che un documento sia rilevante per una data query. Questi modelli sono basati sulla teoria della probabilità e utilizzano il principio dell'incertezza per gestire la variabilità e l'ambiguità inerenti al linguaggio naturale. L'idea fondamentale è *classificare* i documenti in base alla loro *probabilità di rilevanza* per una query.

2.4.1 Il Modello di Rilevanza Probabilistico

Il modello di rilevanza probabilistico (Probabilistic Relevance Model - PRM) assume che esista un insieme ideale di documenti rilevanti per una query. L'obiettivo è classificare i documenti in ordine della loro probabilità di appartenenza a questo insieme. Data una query Q e un documento D_j, il punteggio di rilevanza viene calcolato come la probabilità $P(R = 1 \mid D_j,Q)$, dove R è una variabile binaria che indica la rilevanza.

Usando il teorema di Bayes:

$$P(R = 1 \mid D_j, Q) = \frac{P(Q \mid D_j,R=1)\cdot P(R=1 \mid D_j)}{P(Q)}.$$

Poiché $P(Q)$ è costante per una data query, può essere omesso quando si classificano i documenti. Ciò porta alla funzione di classificazione:

$$\text{Score}(D_j) \propto P(Q \mid D_j,R = 1) \cdot P(R = 1 \mid D_j).$$

Per stimare $P(Q \mid D_j,R = 1)$, il PRM tipicamente assume l'indipendenza tra i termini della query:

$$P(Q \mid D_j,R = 1) = \prod_{t\in Q} \quad P(Dj, R = 1).$$

Per implementazioni pratiche, vengono fatte ipotesi semplificative sulla distribuzione dei termini nei documenti rilevanti e non rilevanti.

2.5 BM25 e Ponderazione dei Termini

Come esempio di modello probabilistico, analizziamo l'algoritmo *BM25* [30]. Basandosi sul modello di rilevanza probabilistico, BM25 introduce una funzione di classificazione ampiamente utilizzata nei moderni sistemi di IR. BM25 affronta due aspetti importanti:

1. **Saturazione della Frequenza del Termine -** A differenza di TF-IDF, dove la frequenza del termine aumenta linearmente, BM25 applica una funzione di saturazione per limitare l'impatto di termini molto frequenti.
2. **Normalizzazione della Lunghezza del Documento -** I documenti più lunghi vengono penalizzati per prevenire punteggi artificialmente elevati dovuti alle occorrenze dei termini.

Il punteggio BM25 per un documento D_j e una query Q è dato da:

$$\text{BM25}(D_j,Q) = \sum_{t \in Q} \quad IDF(t) \cdot \frac{TF(t,D_j)\cdot(k_1+1)}{TF(t,D_j)+k_1\cdot\left(1-b+b\cdot\frac{len(Dj)}{avglen}\right)},$$

dove:

- IDF(t) è la frequenza inversa del documento del termine t;
- TF(t,D_j) è la frequenza del termine t nel documento D_j;
- k_1 e b sono parametri di tuning che controllano la saturazione della frequenza del termine e la normalizzazione della lunghezza del documento;
- len(D_j) è la lunghezza del documento D_j, e avglen è la lunghezza media dei documenti nella collezione.

2.6 Embedding di Documenti e Parole

Mentre i modelli a spazio vettoriale tradizionali rappresentano documenti e query come vettori sparsi basati su frequenze di termini grezze o pesi TF-IDF, questo approccio spesso non riesce a catturare le relazioni semantiche tra i termini. Per affrontare questa limitazione, i moderni sistemi di information retrieval si affidano sempre più agli *embedding*, rappresentazioni vettoriali dense che codificano il significato semantico in uno spazio continuo a bassa dimensionalità.

2.7 Embedding di Parole

Gli *embedding di parole* sono rappresentazioni vettoriali dense di parole, dove parole semanticamente simili sono rappresentate in punti vicini nello spazio adottato. Questi embedding sono tipicamente "appresi" utilizzando modelli basati su reti neurali addestrati su grandi corpora di testo. I metodi più tradizionali per generare embedding di parole includono:

1. **Word2Vec** [22] - Utilizza una rete neurale poco profonda per generare embedding tramite due approcci principali:
   - *Continuous Bag of Words (CBOW)* - Predice una parola dato il suo contesto circostante.
   - *Skip-gram:* Predice le parole del contesto data una parola target.
2. **GloVe (Global Vectors for Word Representation)** [25] - Combina statistiche di co-occorrenza globale delle parole con il contesto locale per produrre embedding.

3. **FastText** [4]: - Estende Word2Vec rappresentando le parole come una collezione di n-grammi di caratteri, consentendo una migliore gestione delle parole fuori vocabolario e delle variazioni morfologiche.

In questi modelli, ogni parola o termine t_i è rappresentata come un vettore $\mathbf{v}_i$ in uno spazio continuo, tale che la similarità del coseno tra due vettori approssima la vicinanza semantica delle parole corrispondenti. Ad esempio:

$$\text{CosineSimilarity}(\mathbf{v}_{\text{re}}, \mathbf{v}_{\text{regina}}) > \text{CosineSimilarity}(\mathbf{v}_{\text{re}}, \mathbf{v}_{\text{auto}}).$$

2.8 Embedding di Documenti

Mentre gli embedding di parole catturano le relazioni semantiche a livello di parola, gli *embedding di documenti* forniscono rappresentazioni vettoriali per interi documenti. Questi embedding aggregano informazioni da singoli embedding di parole o frasi per catturare il contenuto semantico complessivo di un documento. Le tecniche per generare embedding di documenti includono:

1. **Media degli Embedding di Parole** – Un approccio semplice prevede che l'embedding di un documento venga calcolato come la media dei suoi embedding di parole costituenti. Sebbene computazionalmente efficiente, questo metodo potrebbe non riuscire a catturare relazioni e contesti complessi.
2. **Doc2Vec** [16] – Estende Word2Vec introducendo vettori specifici per i documenti. Due approcci principali sono utilizzati:
   - *Distributed Memory (DM)* – Apprende le rappresentazioni vettoriali dei documenti prevedendo le parole basandosi sul vettore del documento e sui vettori delle parole circostanti.
   - *Distributed Bag of Words (DBOW)* – Apprende i vettori dei documenti prevedendo parole specifiche del documento senza usare il contesto.
3. **Transformers** – I modelli linguistici pre-addestrati come BERT [12], T5 [29] e GPT [28] producono potenti embedding contestuali per i documenti. Questi modelli codificano i documenti utilizzando meccanismi di auto-attenzione, consentendo loro di catturare relazioni intricate tra parole e frasi. Per il recupero dei documenti, gli embedding sono tipicamente generati prendendo l'output di un token speciale, come [CLS] in BERT, o raggruppando gli embedding di output di tutti i token.

Nel complesso, gli embedding di documenti e parole hanno rivoluzionato l'information retrieval consentendo la comprensione semantica e migliorando l'accuratezza del recupero dell'informazione più rilevante. Sono stati impiegati per eseguire compiti diversi, come:

- **Ricerca Semantica –** Gli embedding consentono ai sistemi di IR di recuperare documenti che sono semanticamente correlati alla query, anche se non condividono termini esatti. Ad esempio, una query su "fonti di

energia rinnovabile" può recuperare documenti che discutono di "impianti solari" o "fattoria eolica", anche se i termini precisi della query sono assenti dai documenti.

- **Classificazione e Punteggio di Similarità -** Gli embedding densi consentono una classificazione più accurata della rilevanza documento-query, sfruttando metriche come la similarità del coseno o modelli di classificazione basati su reti neurali.
- **Clustering e Classificazione -** Gli embedding facilitano il clustering dei documenti e la modellazione degli argomenti, raggruppando documenti semanticamente simili.

Una rappresentazione geometrica dello spazio semantico consente ai sistemi automatizzati di avere una comprensione più profonda del contesto testuale, migliorando le loro prestazioni e l'esperienza dell'utente.

2.9 Large Language Models

I *Large Language Models (LLM)* sono il punto d'incontro tra i modelli a spazio vettoriale e le reti neurali profonde. Come possiamo vedere nella Fig. 1 nella rassegna di Yang et al. [35] e nella Fig. 1 di questo capitolo, negli ultimi anni gli LLM hanno acquisito un enorme slancio in termini di popolarità, diventando progressivamente più complessi man mano che vengono integrati alle nostre attività quotidiane. Sono basati su reti neurali avanzate progettate per elaborare e generare testo simile a quello umano, imparando da dataset su larga scala e sfruttando architetture sofisticate come i transformer. Il concetto centrale alla base di questi modelli è la loro capacità di rappresentare e manipolare informazioni in uno *spazio di parametri* ad alta dimensionalità, denotato matematicamente come $\theta \in R^d$, dove d è il numero di parametri. Questi parametri, che possono essere centinaia di miliardi, codificano i pesi e i bias della rete neurale, consentendole di modellare schemi complessi nel linguaggio naturale.

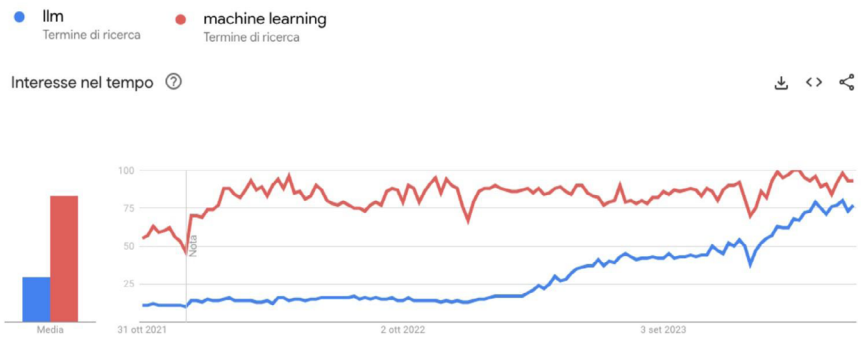

Figura 1: Interesse nel tempo delle query *machine learning* e *llm* su Google. Possiamo notare come gli LLM guadagnino popolarità e si avvicinino al livello della query *machine learning* nel tempo.

Gli LLM, proprio come qualsiasi altro modello di "deep learning" supervisionato, subiscono una fase di addestramento iniziale e una seconda fase di perfezionamento (fine-tuning). Come spiegano i ricercatori di Google Inc. nella loro *Introduzione agli LLM* [11], queste fasi possono essere visualizzate intuitivamente come i passaggi necessari per addestrare un cane da servizio speciale, come nella Fig. 2. In una prima fase, sarà necessario insegnare ai cani le basi come sedersi, stare fermi o rispondere a comandi, abilità che sono ampiamente applicabili e costituiscono le fondamenta del loro addestramento. Questo è analogo alla fase di training iniziale degli LLM, dove il modello apprende schemi e strutture generali del linguaggio da vasti dataset.

Nella seconda fase di perfezionamento, l'attenzione si sposta sull'insegnamento al cane di compiti specifici, come guidare una persona con problemi di vista o rilevare condizioni mediche specifiche. Allo stesso modo, il fine-tuning di un LLM implica la specializzazione del modello per applicazioni particolari, come rispondere a domande specifiche di un dominio o generare contenuti adattati a un determinato pubblico.

2.10 Fase di Training

La fase di addestramento ottimizza i parametri del modello θ per minimizzare una funzione di perdita (loss function) $L(\theta)$, tipicamente definita su un ampio corpus di testo. Ad esempio, nei modelli autoregressivi come GPT, l'obiettivo è massimizzare la probabilità del token successivo t_i dati i token precedenti $t_1,t_2,...,t_{i-1}$. Questo è espresso come:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log P_\theta(t_i|t_1, t_2, \ldots, t_{i-1})$$

Figura 2: Un confronto intuitivo tra i passaggi necessari per addestrare un cane da servizio speciale e le fasi di training e fine-tuning di un LLM [11].

dove N è il numero totale di token nel corpus. La probabilità $P_0(t_i | t_1,t_2,...,t_{i-1})$ è tipicamente calcolata utilizzando l'architettura transformer [32], che impiega meccanismi di auto-attenzione multi-testa:

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

dove Q, K e V sono matrici di query, chiave e valore derivate dagli embedding di input, e d_k è la dimensionalità delle chiavi. I livelli impilati del transformer consentono al modello di catturare dipendenze sia locali che globali lungo la sequenza.

2.11 Fase di Fine-Tuning

Dopo il pre-training, il modello può essere sottoposto a fine-tuning per compiti specifici. Ciò comporta un ulteriore aggiornamento di θ utilizzando un dataset più piccolo specifico del dominio D_{task}. Ad esempio, nel fine-tuning supervisionato, la funzione di perdita viene modificata per riflettere l'obiettivo specifico del compito:

$$L_{\text{task}}(\theta) = -\sum_{(x,y)\in D_{task}} \quad logP_\theta(x),$$

dove x e y sono coppie input-output (ad esempio, domande e risposte). Tecniche come il Reinforcement Learning with Human Feedback (RLHF) [9] raffinano il modello ottimizzando le "ricompense" (reward) $R(\theta)$ derivate dalle preferenze umane.

2.12 LLM Iniziali e Moderni

L'evoluzione degli LLM è iniziata con modelli come GPT di OpenAI [27], che hanno introdotto il pre-training non supervisionato su grandi corpus seguito dal fine-tuning. GPT-2 [28] ha dimensionato questo approccio con 1,5 miliardi di parametri, mentre GPT3 [5] si è espanso a 175 miliardi di parametri, consentendo l'apprendimento anche a partire da pochi esempi. Questi progressi si basano su strategie di parallelizzazione efficienti, come il parallelismo di modello e dati, per gestire le immense richieste computazionali.

Modelli moderni come GPT-4 e PaLM 2 di Google [1] spingono ulteriormente i limiti, incorporando perfezionamenti architettonici e tecniche di training avanzate. Ad esempio, questi modelli impiegano spesso tecniche come la regolarizzazione dropout, la normalizzazione di livello e gli schemi di tasso di apprendimento adattivi per stabilizzare il training e migliorare la generalizzazione. La dimensione di questi modelli è spesso misurata in notazione scientifica, con conteggi di parametri che raggiungono 10^{11} ed oltre.

3. (De)-Indicizzazione e implicazioni sul RTBF

Avendo stabilito una base tecnica sui modelli di Information Retrieval, possiamo ora riassumerli e metterli in relazione con la questione dell'indicizzazione (e de-indicizzazione) dei contenuti dal Web. L'emergere degli embedding neurali ha portato significativi progressi nel campo del recupero delle informazioni, in particolare nel modo in cui i documenti vengono rappresentati, recuperati e analizzati. A differenza dei modelli tradizionali come il *bag-of-words*, che riducono i documenti a collezioni non ordinate di frequenze di parole e non riescono a catturare il significato contestuale o semantico, gli embedding neurali forniscono una rappresentazione robusta dei documenti all'interno di uno spazio vettoriale continuo. Questo cambiamento consente ai sistemi di IR di comprendere meglio il contenuto e il contesto dei documenti, sbloccando nuove possibilità per il recupero e l'analisi semantica.

Come già descritto precedentemente, gli *embedding di documenti* codificano il significato e le relazioni di parole e frasi, consentendo una rappresentazione più profonda e sfumata del contenuto testuale. Proiettando i documenti in spazi ad alta dimensionalità, gli embedding catturano somiglianze semantiche che non sono facilmente discernibili utilizzando approcci basati su parole chiave. Questa capacità è particolarmente preziosa per superare i limiti delle tecniche IR tradizionali, che spesso si basano sulla corrispondenza esatta delle parole chiave e faticano con sinonimi, parafrasi o variazioni nel linguaggio.

Un altro grande vantaggio degli embedding di documenti risiede nella loro capacità di calcolare la somiglianza semantica tra i documenti sfruttando metriche di distanza, come la similarità del coseno, nello spazio degli embedding, come abbiamo descritto in precedenza. Tale meccanismo consente il recupero di documenti semanticamente correlati, anche quando le loro parole chiave differiscono o sono disposte in un ordine non corrispondente. Ad esempio, due documenti che trattano lo stesso argomento, ma espressi in modi unici, possono comunque essere identificati come strettamente correlati nello spazio degli embedding, migliorando notevolmente l'accuratezza e la rilevanza del recupero

Inoltre, gli embedding neurali mostrano impressionanti capacità di generalizzazione, in particolare quando addestrati su grandi corpus. Questi modelli possono applicare efficacemente le relazioni semantiche apprese a documenti nuovi e non visti, rendendoli altamente adattabili a collezioni di testo di dimensione variabile ed in continua crescita come quelle presenti sul Web. Questa adattabilità al ridimensionamento (scalability) garantisce che i sistemi di IR possano tenere il passo con la rapida evoluzione dei contenuti, fornendo prestazioni costanti e un'ampia copertura anche quando emergono nuovi argomenti.

Tuttavia, come spesso accade, i vantaggi recano con loro anche delle problematiche. Controllare e gestire la visibilità delle informazioni sul Web è un compito complesso. La rimozione completa di un documento è spesso tecnicamente

infattibile, richiedendo interventi come modifiche al DNS o modifiche alle policy di routing di basso livello, che possono essere aggirate. Una soluzione più pratica comporta la de-indicizzazione di documenti specifici dai principali motori di ricerca. Questo approccio non rimuove fisicamente il contenuto dal Web; invece, riduce la visibilità dei documenti mirati e interrompe le loro relazioni contestuali nello spazio degli embedding. Ad esempio, consideriamo il caso di un individuo (il signor John Smith) che abbia deciso di esercitare il suo RTBF chiedendo ai principali fornitori di motori di ricerca di de-indicizzare il contenuto relativo a un caso giudiziario in cui era coinvolto. Nello specifico, chiederà di rimuovere tutti i documenti che lo collegano alla questione in discussione, che indicheremo con il token "X". Se i documenti che collegano "Smith" e "X" vengono de-indicizzati, la loro associazione semantica si indebolisce; d'altra parte, tuttavia, altre associazioni, come quella tra "Smith" e un altro token "Y", potrebbero rafforzarsi. Questo aggiustamento dinamico delle relazioni nello spazio degli embedding ha profonde implicazioni su come le informazioni vengono rappresentate e recuperate: non sappiamo se la nuova associazione con "Y" sia, secondo il signor Smith, più o meno desiderabile, e non è facile prevedere le conseguenze che tale operazione possa avere sullo spazio degli embedding su scala più ampia. Esistono già casi che sono, apparentemente, correlati a casi di rimozione di informazioni di individui dagli LLM. Un esempio recente è il caso di David Mayer, un prompt con questo nome ha fatto sì che ChatGPT producesse un messaggio di errore senza alcuna ulteriore spiegazione[5]. Il problema ha portato a varie teorie tra gli utenti, inclusi problemi di privacy e il diritto all'oblio. Alcuni hanno ipotizzato che lo stesso David Mayer potesse aver richiesto che le sue informazioni fossero rimosse dalle risposte di ChatGPT, sebbene ciò sia stato successivamente etichettato da OpenAI come un'incomprensione del problema tecnico in questione[6]. È opportuno, inoltre, specificare che la risposta restituita da ChatGPT in data 17/9/2025 sia più articolata e venga specificato che il sig. David Mayer de Rothschild, che su Wikipedia viene descritto come un famoso ambientalista inglese, non abbia fatto richiesta specifica ad OpenAI perché le informazioni sul suo conto venissero rimosse. Anche se questo caso potrebbe non essere correlato alla questione specifica della rimozione della conoscenza, evidenzia come aziende private che gestiscono i sistemi di AI generativa dovrebbero adottare metodi sofisticati per affrontare il problema, perché modifiche "brutali" al software (patch hard-coded) e a posteriori potrebbero causare problemi di comunicazione tra la base utenti.

5 https://shorturl.at/nisj8.

6 https://shorturl.at/KREbP.

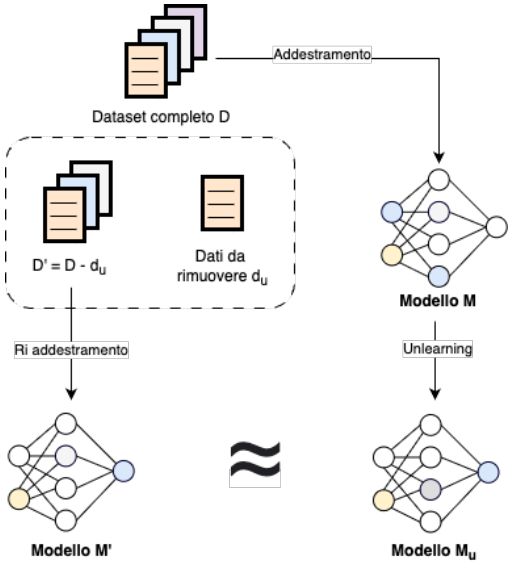

Figura 3: Machine Unlearning workflow

Queste considerazioni inevitabilmente impattano sull'oblio e sul RTBF, avendo implicazioni sugli individui e sulla memoria collettiva. Esistono diversi studi che hanno esplorato, da un punto di vista empirico e matematico, le dinamiche della memoria e dell'attenzione collettiva. Ad esempio, in [7] gli autori differenziano tra memoria comunicativa e culturale, testando una funzione di decadimento bi-esponenziale per l'attenzione in vari domini culturali, implicando che l'attenzione collettiva iniziale (che riflette la memoria comunicativa) diminuisce rapidamente, seguita da un declino più lento (che riflette la memoria culturale). Un approccio empirico a questo argomento potrebbe essere utile per quantificare l'impatto che queste nuove forme di recupero delle informazioni, come i motori di ricerca basati su LLM, possono avere sulla memoria collettiva e sul RTBF. Sebbene ci siano chiari aspetti positivi, come un recupero e una condivisione rapidi e (si spera) accurati delle informazioni, che sono dinamicamente contestualizzati in base alle interazioni dell'utente, ci sono anche aspetti negativi. Molti di essi sono comuni anche ad altre forme moderne di recupero delle informazioni, come l'eccessiva dipendenza dalla tecnologia specifica, o un'omogeneizzazione della conoscenza che può verificarsi se gli LLM tendono a privilegiare narrazioni popolari o ampiamente accettate, mettendo in secondo piano le prospettive minoritarie. Ciò che è nuovo, invece, è la profonda complessità delle dinamiche dell'oblio delle macchine, che potrebbe andare in due direzioni: o in un oblio involontario della conoscenza (che è in parte ciò che accade nel processo di omogeneizzazione sopra menzionato) o nell'impossibilità di un oblio completo, a causa del profondo intreccio di concetti nello spazio degli embedding.

3.1 L'Oblio come Problema di *Machine Unlearning*

In contrapposizione (o in complementarità) alla de-indicizzazione a posteriori, possiamo definire il problema del *machine unlearning* (MU) o apprendimento inverso, ovvero il processo di rimozione selettiva dell'influenza di specifici dati usati in fase di addestramento da un modello di machine learning esistente. L'obiettivo del machine unlearning è consentire a un modello di comportarsi come se non avesse mai ricevuto in input determinati dati, affrontando così molti problemi di privacy, di fairness e migliorando l'adattabilità del modello.

Nel machine unlearning, idealmente, il modello dimentica completamente i dati che hanno contribuito al suo addestramento, eliminando di fatto il loro impatto. In questo modo, se lo stesso dato viene reintrodotto in futuro, il sistema lo elabora come se fosse del tutto nuovo, senza alcuna conoscenza residua derivante dall'apprendimento precedente.

Formalmente, si consideri $D = \{(x_1,y_1),\cdots,(x_n,y_n)\}$ come il dataset originale e completo di addestramento, dove x_i rappresenta le caratteristiche di input e y_i l'etichetta corrispondente, e sia M il modello di apprendimento automatico addestrato su D. Si definisca d_u come l'insieme dei dati da rimuovere, tale che $D' \subseteq D$ e $D' = D - d_u$. Una volta avviato il processo di unlearning, il modello deve cancellare tutte le informazioni relative a d_u.

Il modello risultante, denotato con M_u, dovrebbe essere indistinguibile da un modello di riferimento M' addestrato da zero utilizzando solo D'. La Figura 3 illustra il processo di MU, che rimuove specifici punti dati da un modello di apprendimento automatico già addestrato, preservandone al contempo la funzionalità complessiva.

Un interessante rapporto dello Stanford AI Lab [19] riassume le prospettive passate, presenti e future del machine unlearning, descrivendo diversi approcci che consentono la rimozione di informazioni irrilevanti o obsolete, aiutando i modelli a mantenere le loro prestazioni nel tempo. Sebbene non sia un compito banale, diventa ancora più impegnativo nel contesto dei modelli di deep learning e degli LLM (anche a causa degli attacchi informatici [23]). È stato infatti dimostrato come essi tendano a mantenere là conoscenza anche dopo i tentativi di "unlearning", complicando gli sforzi per rimuovere capacità dannose o indesiderate instillate durante il pre-training [8,10].

Le tecniche di machine unlearning possono essere classificate in base all'esattezza dell'"unlearning" ottenuto nelle seguenti due tipologie principali [17,31]:

- **Unlearning Esatto** – Gli algoritmi di unlearning perfetto o esatto mirano a produrre un modello identico a quello che sarebbe stato ottenuto addestrandolo da zero su un dataset da cui è stato escluso uno specifico punto dati da dimenticare. Questo rappresenta lo scenario ideale, ma raggiungere tale precisione è un compito estremamente complesso. Di conseguenza, al momento, l'unico vero metodo di unlearning esatto è il riaddestramento completo del modello.

- **Unlearning Approssimato** – L'unlearning approssimato rappresenta un'alternativa più efficiente in termini di costi ed è particolarmente utile per algoritmi di apprendimento automatico complessi e adattivi, nei quali è spesso impraticabile ricostruire con precisione la sequenza e l'impatto dei singoli dati.

Una classificazione aggiuntiva delle tecniche di MU si basa sul livello di garanzia che i dati specifici siano stati effettivamente cancellati dal modello. In base a questo criterio, è possibile distinguere i seguenti tipi di metodi:

- **MU Certificato** – Fornisce una garanzia formale che un modello da cui sono stati rimossi dei dati sia indistinguibile da un modello che non ha mai osservato quei dati fin dall'inizio [14]. Anche se questo metodo offre forti garanzie in termini di privacy e sicurezza, il riaddestramento richiesto è molto dispendioso in termini di risorse e può risultare impraticabile per modelli di grandi dimensioni.
- **MU Empirico** – Non fornisce garanzie teoriche formali in merito alla sicurezza o correttezza dell'unlearning, ma ne valuta l'efficacia tramite osservazioni sperimentali e valutazioni empiriche. Grazie alla sua maggiore fattibilità pratica e al minor carico computazionale, il MU empirico è ampiamente adottato in applicazioni reali.

Un potenziale approccio per bilanciare i benefici degli embedding neurali con la necessità di una maggiore interpretabilità e controllo è rappresentato dalla *Generazione Potenziata da Recupero dati (Retrieval-Augmented Generation, RAG)* [2]. I sistemi RAG combinano i punti di forza del recupero semantico e dei modelli generativi, consentendo loro di fornire risposte accurate e contestualmente consapevoli, mitigando alcuni dei rischi associati agli spazi di embedding opachi. Integrando recupero e generazione in un quadro unificato, RAG offre un percorso per migliorare la precisione e la rilevanza dei sistemi di IR mantenendo un certo grado di supervisione e adattabilità. Date le sue caratteristiche, RAG viene esplorato come una potenziale soluzione che consente l'oblio simulato senza interazione diretta con il modello stesso [15,33]. Questo approccio affronta alcune delle significative limitazioni degli LLM tradizionali, come la loro tendenza a produrre informazioni imprecise o obsolete, comunemente denominate "allucinazioni". Sfruttando fonti di dati esterne, RAG consente ai modelli di generare output più accurati e contestualmente rilevanti. In generale, un framework RAG consiste di diversi componenti interconnessi che lavorano insieme per recuperare informazioni pertinenti e generare risposte:

- **Input Query Utente** – Il processo inizia quando un utente invia una query. Questo input viene proiettato in uno spazio di embedding LLM per catturare il significato semantico della query.

- **Sistema di Recupero**
  - **Recupero Documenti** – Il sistema di recupero scansiona *basi di conoscenza esterne* (ad esempio, database o collezioni di documenti) per recuperare blocchi di testo rilevanti. Questo può coinvolgere metodi tradizionali basati su parole chiave (recupero sparso) o moderne tecniche di recupero denso che utilizzano gli embedding per la ricerca di somiglianza.
  - **Classificazione e Filtraggio** – Dopo aver recuperato i potenziali documenti, il sistema li classifica in base alla rilevanza, selezionando tipicamente gli N documenti più pertinenti per ulteriore elaborazione.
- **Generazione di Embedding Contestuali** – Ogni documento recuperato viene anche convertito in un embedding per garantire che il modello generativo possa incorporare efficacemente questa informazione durante la generazione della risposta.
- **Meccanismo di Fusione** – I documenti recuperati vengono fusi con la query originale tramite metodi di fusione precoce o tardiva. Nella fusione precoce, sia la query che i documenti vengono immessi nel modello generativo contemporaneamente; nella fusione tardiva, i documenti recuperati raffinano l'output del modello dopo la generazione iniziale [26].
- **Generazione della Risposta** – L'LLM genera una risposta coerente basata sull'input aumentato, sfruttando sia la sua conoscenza preesistente che le informazioni appena recuperate.

Un framework basato su RAG può facilitare il machine unlearning in diversi modi, per esempio aggiornando in tempo reale la base di conoscenza (knowledge base) senza riaddestrare l'intero modello. Quando dati specifici devono essere dimenticati, possono essere rimossi dalla knowledge base esterna. Questa azione simula efficacemente l'"unlearning" assicurando che future query non recuperino o si basino su questi dati. Gestendo una knowledge base esterna, RAG supporta l'apprendimento continuo consentendo al contempo l'oblio selettivo di informazioni dannose o obsolete. La dipendenza di RAG da un sistema di recupero esterno consente aggiornamenti più efficienti, dove solo la knowledge base deve essere modificata piuttosto che il modello stesso.

Queste considerazioni tecniche si intersecano con questioni etiche e di governance critiche. Il potere di decidere quali documenti vengono de-indicizzati, e quindi resi meno visibili, spetta prevalentemente a pochi attori dominanti nell'ecosistema IR, anche se è regolamentato in misura diversa dalle autorità centrali. Questa concentrazione di influenza potrebbe sollevare preoccupazioni sulla trasparenza, la responsabilità e il potenziale abuso. Inoltre, la fase di addestramento degli embedding neurali svolge un ruolo decisivo nel modellare le relazioni semantiche codificate nello spazio vettoriale adottato. Tuttavia, come abbiamo descritto, questa fase è intrinsecamente opaca, rendendo difficile garantire l'equità nei sistemi di IR risultanti.

Come anche molte fonti autorevoli prevedono [20], è molto probabile che, considerate le sue interessanti sfide tecniche e le sue cruciali e multiformi implicazioni sulla società, l'oblio sarà un argomento di ricerca di spicco nell'informatica e, più in generale, nella ricerca legata all'IA nei prossimi anni.

Riferimenti bibliografici

1. Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.
2. Amazon Web Services (AWS). What is retrieval-augmented generation (rag)?, 2024. Accessed: 2024-12-12.
3. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley, Reading, MA, 1999.
4. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.
5. Tom B. Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
6. Stefan Buttcher, Charles Clarke, and Gordon Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, Cambridge, MA, 2nd edition, 2016.
7. Cristian Candia, C. Jara-Figueroa, Carlos Rodriguez-Sickert, Albert-L´aszlò Baraba´si, and Cèsar A Hidalgo. The universal decay of collective memory and attention. *Nature human behaviour*, 3(1):82–91, 2019.
8. Stephen Casper. Deep forgetting & unlearning for safely-scoped llms, 2023. Accessed: 2025-01-03.
9. Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
10. Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *arXiv preprint arXiv:2205.09357*, 2022. Accessed: 2025-01-03.
11. Google Developers. Introduction to large language models, 2024. Accessed: 202412-10.
12. Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
13. Cicero dos Santos, Xiaofei Ma, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. Beyond [cls] through ranking by generation. In *Proceedings*

of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1722–1727, 2020.

14. Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models, 2023.
15. Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. Learn to unlearn for deep neural networks: Minimizing unlearning interference with gradient projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024. Accessed: 2025-01-03.
16. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.
17. Na Li, Chunyi Zhou, Yansong Gao, Hui Chen, Zhi Zhang, Boyu Kuang, and Anmin Fu. Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.
18. Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained transformers for text ranking: BERT and beyond. *arXiv preprint*, arXiv:2010.06467, 2020.
19. Ken Ziyu Liu. Unlearning in ai and machine learning, 2024. Accessed: 2025-01-03.
20. Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787*, 2024.
21. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schu¨tze. *Introduction to Information Retrieval*. MIT Press, Cambridge, MA, 1st edition, 2008.
22. Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781, 2013.
23. Serena Nicolazzo, Antonino Nocera, et al. How secure is forgetting? linking machine unlearning to machine learning attacks. *arXiv preprint arXiv:2503.20257*, 2025.
24. Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *arXiv preprint*, arXiv:1901.04085, 2019.
25. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
26. Zackary Rackauckas. RAG-Fusion: a New Take on Retrieval-Augmented Generation. *arXiv preprint arXiv:2402.03367*, 2024. Accessed: 2025-01-03.
27. Alec Radford. Improving language understanding by generative pre-training. *Open AI*, 2018.

28. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
29. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
30. Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333– 389, 2009.
31. Siva Sai, Uday Mittal, Vinay Chamola, Kaizhu Huang, Indro Spinelli, Simone Scardapane, Zhiyuan Tan, and Amir Hussain. Machine un-learning: an overview of techniques, applications, and future directions. *Cognitive Computation*, 16(2):482–506, 2024.
32. A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
33. Shang Wang, Tianqing Zhu, Dayong Ye, and Wanlei Zhou. When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge? *arXiv preprint arXiv:2410.15267*, 2024. Accessed: 2025-01-03.
34. Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, CA, 1999.
35. Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Trans. Knowl. Discov. Data*, 18(6), April 2024.
36. Shengyao Zhuang, Hang Li, and Guido Zuccon. Deep query likelihood model for information retrieval. In *Proceedings of the 43rd European Conference on Information Retrieval (ECIR)*, 2021.