

# Università della Calabria

Dipartimento di Matematica e Informatica

---



Corso di Laurea Magistrale in Informatica

Tesi di Laurea

## **GAEN: Generative AI for Enhanced Narrative - Rilevamento di Spoiler Mediante Modelli Generativi Avanzati**

Relatori:

Prof. Gianluigi Greco

Prof. Kazumi Saito

Candidato:

Alessandro Fazio

Matricola 242422

---

Anno Accademico 2024/2025



# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Contesto . . . . .	3
1.2	Motivazioni . . . . .	3
1.3	Obiettivi . . . . .	4
1.4	Struttura della tesi . . . . .	4
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Approcci al problema . . . . .	5
2.1.1	Approccio basato su regole . . . . .	5
2.1.2	Approccio basato su machine learning . . . . .	6
2.2	Approccio proposto . . . . .	6
<b>3</b>	<b>Lavori correlati</b>	<b>8</b>
3.1	Contesto della Ricerca . . . . .	8
3.2	LLM, modelli di linguaggio . . . . .	8
3.2.1	Attention is all you need . . . . .	8
3.3	Embeddings e Latent Space . . . . .	8
3.3.1	Vector Database . . . . .	8
<b>4</b>	<b>Design e implementazione</b>	<b>9</b>
<b>5</b>	<b>Valutazione</b>	<b>10</b>
<b>6</b>	<b>Conclusioni</b>	<b>11</b>

# Capitolo 1

## Introduzione

### 1.1 Contesto

Nell'era digitale, l'accesso immediato a contenuti multimediali ha trasformato radicalmente il modo in cui fruiamo di film, serie TV, libri e videogiochi. Tuttavia, questa abbondanza di informazioni porta con sé una sfida: la crescente esposizione a spoiler. Gli spoiler, rivelando anticipatamente elementi cruciali della trama, possono compromettere l'esperienza di fruizione, generando frustrazione e delusione negli utenti.

Il problema degli spoiler è particolarmente rilevante in contesti online, dove le discussioni e le recensioni proliferano su forum, social media e piattaforme di streaming. La mancanza di strumenti efficaci per il rilevamento automatico di spoiler rende difficile per gli utenti proteggersi da tali rivelazioni indesiderate.

Questa tesi si propone di esplorare l'utilizzo di Large Language Models (LLM) per affrontare la sfida del rilevamento di spoiler. L'obiettivo è sviluppare un sistema in grado di identificare automaticamente gli spoiler in testi di varia natura, sfruttando le capacità di comprensione del linguaggio naturale degli LLM.

### 1.2 Motivazioni

La motivazione principale di questa ricerca risiede nella volontà di contribuire allo sviluppo di strumenti che migliorino l'esperienza di fruizione dei contenuti multimediali. A tale scopo, è stato realizzato un sistema di rilevamento di spoiler integrato in una estensione per browser, che offre agli utenti un controllo maggiore sulla propria esposizione agli spoiler.

Questa ricerca intende esplorare il potenziale degli LLM in un'applicazione pratica e rilevante, provando a contribuire alla comprensione delle loro capacità e limitazioni nel contesto del rilevamento di informazioni specifiche in testi complessi.

## 1.3 Obiettivi

Gli obiettivi di questa tesi sono i seguenti:

- Realizzare un sistema di rilevamento di spoiler basato su Large Language Models.
- Valutare l'efficacia del sistema sviluppato nel rilevamento di spoiler in testi di varia natura.
- Integrare il sistema in un'estensione per browser e valutarne l'utilità e l'usabilità.

## 1.4 Struttura della tesi

Il resto della tesi è organizzato come segue:

- Nel Capitolo 2 vengono presentati i concetti e le tecnologie alla base della ricerca, con particolare attenzione ai Large Language Models.
- Nel Capitolo 3 vengono esaminati i lavori correlati, con un focus sulle ricerche relative al rilevamento di spoiler.
- Nel Capitolo 4 viene descritto il sistema di rilevamento di spoiler sviluppato, con particolare attenzione alla progettazione e all'implementazione.
- Nel Capitolo 5 vengono presentati i risultati dell'analisi sperimentale condotta per valutare l'efficacia del sistema.
- Nel Capitolo 6 vengono riassunti i risultati ottenuti e vengono discusse le possibili direzioni future di ricerca.

# Capitolo 2

## Background

### 2.1 Approcci al problema

Di seguito vengono presentati alcuni approcci al problema di rilevazione di spoiler in un testo.

#### 2.1.1 Approccio basato su regole

Un approccio molto semplice consiste nell'identificare alcune parole chiave che sono tipicamente presenti in un testo che contiene spoiler. Ad esempio, le parole “spoiler” e “spoiler alert” sono spesso utilizzate per avvertire il lettore che il testo successivo contiene informazioni che potrebbero rovinargli la visione di un film o la lettura di un libro. Altri esempi di parole chiave potrebbero essere i nomi di personaggi o luoghi chiave della trama. Questo approccio è molto semplice e può essere implementato con poche righe di codice, ma ha il difetto di essere molto limitato e di non essere in grado di rilevare spoiler più sottili o nascosti.

Questo approccio è inoltre soggetto a molti falsi positivi, ovvero a casi in cui il testo viene erroneamente classificato come contenente spoiler quando in realtà non lo contiene. Ad esempio, una recensione di un film potrebbe contenere il nome di un personaggio chiave della trama senza rivelare informazioni cruciali sulla storia. In questo caso, il testo non dovrebbe essere classificato come contenente spoiler, ma l'approccio basato su regole potrebbe erroneamente classificarlo come contenente spoiler.

Oltre ai falsi positivi, questo approccio è anche soggetto a falsi negativi, ovvero a casi in cui il testo contiene spoiler ma non viene classificato come tale. Ad esempio, una recensione di un film potrebbe contenere informazioni cruciali sulla trama senza utilizzare le parole chiave tipicamente associate

ai spoiler, o in casi anche più semplificati, potrebbe essere scritta in una lingua diversa da quella in cui sono state definite le parole chiave, o potrebbe contenere errori di battitura o di ortografia che impediscono al sistema di riconoscere le parole chiave.

Per questi motivi, l'approccio basato su regole non promette risultati soddisfacenti e si è deciso di non adottarlo per questo progetto.

## **2.1.2 Approccio basato su machine learning**

Un approccio più sofisticato e promettente è quello basato su machine learning. In questo approccio, si addestra un modello di machine learning su un insieme di dati di addestramento etichettati, ovvero un insieme di testi già classificati come contenenti spoiler o non contenenti spoiler. Il modello di machine learning impara a riconoscere i pattern nei dati di addestramento e a classificare i testi in base a tali pattern. Una volta addestrato, il modello può essere utilizzato per classificare nuovi testi come contenenti spoiler o non contenenti spoiler.

Questo approccio ha il vantaggio di essere molto più flessibile e potente rispetto all'approccio basato su regole, in quanto il modello di machine learning è in grado di riconoscere pattern complessi e sottili nei dati di addestramento e di generalizzare tali pattern ai nuovi dati. Inoltre, il modello di machine learning è in grado di adattarsi automaticamente ai nuovi dati, senza la necessità di modificare manualmente le regole o i parametri del sistema.

Tuttavia, l'approccio basato su machine learning ha anche alcuni svantaggi. In primo luogo, richiede un insieme di dati di addestramento etichettati, che possono essere costosi e laboriosi da ottenere.

Il secondo svantaggio è che il modello di machine learning è difficile da realizzare e richiede competenze avanzate per la sua implementazione. Inoltre, il modello deve essere addestrato su un insieme di dati di addestramento rappresentativo e bilanciato, altrimenti potrebbe essere soggetto a overfitting o a underfitting.

Generalizzando, l'approccio basato su machine learning è più complesso e richiede più risorse rispetto all'approccio basato su regole, ma promette risultati migliori e più accurati.

## **2.2 Approccio proposto**

Per affrontare il problema della rilevazione di spoiler in un testo, si propone di utilizzare un approccio basato su machine learning.

L'intuizione alla base di questo approccio è che i LLM (Large Language Model) sono in grado di catturare le relazioni semantiche e sintattiche tra le parole e i concetti all'interno di un testo, e quindi di riconoscere i pattern che caratterizzano i testi contenenti spoiler.

In particolare, si propone di utilizzare un modello di LLM pre-addestrato con lo scopo di estrarre lo spoiler da un testo piuttosto che classificare l'intero testo come contenente spoiler o non contenente spoiler, nel caso in cui il testo contenga spoiler.

Utilizzare un modello di LLM pre-addestrato ha il vantaggio di non richiedere un insieme di dati di addestramento etichettati, in quanto il modello è già stato addestrato su un vasto insieme di dati e ha imparato a riconoscere i pattern nei testi in modo automatico e generale.

A prescindere dal modello di LLM utilizzato, si propone di utilizzare un approccio basato su RAG (Retrieval-Augmented Generation) per arricchire il testo con informazioni realitive al contesto in cui è stato scritto. Ciò permette di migliorare la qualità delle predizioni del modello di LLM, in quanto il contesto può influenzare il significato delle parole e dei concetti all'interno del testo.

La difficoltà principale di questo approccio è che i modelli di LLM sono molto complessi e richiedono risorse computazionali e di memoria considerevoli per essere addestrati e utilizzati. Inoltre, i modelli di LLM sono soggetti a fenomeni di allucinazione e di bias, che possono influenzare le predizioni del modello e ridurre l'accuratezza.

Tuttavia, nonostante queste difficoltà, gli LLM si sono dimostrati molto efficaci in una varietà di task di NLP (Natural Language Processing) e promettono risultati soddisfacenti per il problema della rilevazione di spoiler in un testo.

I dettagli dell'approccio proposto verranno discussi nei seguenti capitoli.



# Capitolo 3

## Lavori correlati

### 3.1 Contesto della Ricerca

In questo capitolo verranno presentati alcuni lavori correlati all'oggetto di questa tesi. In particolare saranno introdotti alcuni concetti chiave e verranno discussi in modo da poter meglio contestualizzare il lavoro svolto.

### 3.2 LLM, modelli di linguaggio

#### 3.2.1 Attention is all you need

### 3.3 Embeddings e Latent Space

#### 3.3.1 Vector Database

## Capitolo 4

# Design e implementazione

# Capitolo 5

## Valutazione

## Capitolo 6

## Conclusioni