Università della Calabria

Dipartimento di Economia Statistica e Finanza

Corso di Laurea in **Data Science per le Strategie Aziendali**

Text Summarization su CNN/DailyMail: Con BART, Lead-3 e TF-IDF

Professore

Andrea Tagarelli
Domenico Mandaglio

Studenti

Pierfrancesco Lindia	**256641**
Cristian Tedesco	**269279**
Nabil Larhram	**269686**

Anno Accademico 2025 / 2026

Indice

1 Introduzione

La *text summarization* mira a produrre una rappresentazione sintetica di un documento preservandone i contenuti informativi principali. Nel contesto informativo contemporaneo — caratterizzato da elevati volumi di notizie, report e documenti testuali — la summarization costituisce uno strumento utile per supportare attività di *decision-making*, analisi di mercato e monitoraggio informativo, riducendo tempi di lettura e carico cognitivo.

In letteratura si distinguono due principali famiglie di approcci: (i) metodi *extractive*, che selezionano porzioni del testo originale (tipicamente frasi) ritenute salienti, e (ii) metodi *abstractive*, che generano un riassunto riformulando l'informazione tramite modelli neurali, con potenziale maggiore fluidità e capacità di compressione, a fronte di una maggiore complessità computazionale e del rischio di introdurre dettagli non strettamente presenti nel testo sorgente.

Questo lavoro presenta un workflow end-to-end per la summarization sul benchmark **CNN/DailyMail** (v3.0.0), con l'obiettivo di condurre un confronto sistematico tra:

- un approccio *abstractive* basato su un modello Transformer encoder–decoder (**BART**), adattato al task tramite **fine-tuning supervisionato** su coppie *(articolo, highlights)*;
- due baseline *extractive* semplici e interpretabili: **Lead-3**, che restituisce le prime tre frasi dell'articolo, e **TF–IDF + similarità coseno**, che seleziona le frasi lessicalmente più rappresentative del documento.

Dal punto di vista sperimentale, l'addestramento viene effettuato su un subset controllato dello split *train* (20 000 esempi) e monitorato su un subset di *validation* (1 000 esempi). La valutazione comparativa finale è condotta su un campione fissato e riproducibile di *validation* ($N = 500$), selezionato tramite *shuffle* con seed costante. La qualità dei riassunti è misurata quantitativamente mediante le metriche **ROUGE** (ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum), che stimano l'overlap lessicale tra riassunti generati e riassunti di riferimento (*highlights*). A tali risultati si affianca un'analisi qualitativa su esempi selezionati, utile a evidenziare aspetti non completamente catturati da ROUGE, quali coerenza discorsiva, ridondanza e differenze stilistiche tra metodi extractive e abstractive.

Infine, per garantire robustezza operativa su documenti lunghi e rispettare i vincoli di input del modello, la pipeline include una strategia di **gestione degli input lunghi** basata su *chunking* token-based e generazione **gerarchica** (riassunti locali per chunk seguiti da un riassunto finale), insieme a un controllo **dinamico** della lunghezza dell'output. Nel complesso, il progetto evidenzia i trade-off tra semplicità e interpretabilità degli approcci estrattivi e la capacità di sintesi del modello neurale fine-tunato, fornendo un quadro sperimentale riproducibile e motivato.

2 Text Summarization

La *text summarization* consiste nel generare una versione sintetica di un documento testuale, preservando i contenuti informativi principali e riducendo ridondanza e

dettagli secondari. In generale, un buon riassunto dovrebbe bilanciare tre proprietà: (i) **copertura** dei punti chiave, (ii) **concisione** (compressione dell'informazione) e (iii) **leggibilità** (coerenza e scorrevolezza). In ambito applicativo, la summarization è utilizzata per facilitare la fruizione rapida di notizie, report, documenti aziendali e collezioni testuali, supportando processi decisionali e attività di monitoraggio informativo.

Nel caso specifico dei news articles, come nel dataset CNN/DailyMail, il documento sorgente contiene molti dettagli, mentre il riassunto di riferimento mira a mantenere solo gli elementi essenziali (chi, cosa, dove, quando e perché), spesso in forma molto compatta. Questo setting rende la summarization un problema supervisionato naturale, in cui si dispone di coppie *(testo, riassunto)* per addestrare o adattare modelli.

2.1 Approcci Extractive vs Abstractive

Gli approcci alla summarization si distinguono comunemente in *extractive* e *abstractive*.

Summarization estrattiva (extractive): Un metodo estrattivo costruisce il riassunto **selezionando** segmenti del documento originale (tipicamente frasi) e concatenandoli. Il vantaggio principale è la **fedeltà** al testo sorgente: non avvengono parafrasi e quindi è ridotto il rischio di introdurre informazioni non presenti nel documento. Inoltre, questi metodi risultano spesso **interpretabili**, poiché le frasi selezionate possono essere motivate con criteri espliciti (posizione nel testo, pesi lessicali, similarità, ecc.). Di contro, il riassunto può risultare meno fluido, ridondante o incoerente quando le frasi selezionate non sono contigue o non si integrano bene tra loro.

Summarization astrattiva (abstractive): Un metodo astrattivo genera un riassunto **nuovo**, riformulando l'informazione presente nel documento. I modelli neurali moderni (in particolare i Transformer) possono produrre output più **compatti** e **scorrevoli**, effettuando compressione semantica e parafrasi. Tuttavia, tali modelli possono talvolta introdurre dettagli non esplicitamente presenti nel testo sorgente (*hallucination*), con potenziale perdita di fedeltà fattuale. Inoltre, l'approccio astrattivo è più oneroso dal punto di vista computazionale e richiede procedure di addestramento/ottimizzazione più complesse rispetto alle baseline estrattive.

2.2 Modelli Transformer encoder–decoder

Molti sistemi astrattivi di summarization si basano su modelli *seq2seq* (sequence-to-sequence), che apprendono una mappatura da una sequenza di input (testo sorgente) a una sequenza di output (riassunto). I Transformer rappresentano lo standard attuale per questi compiti, grazie all'uso del meccanismo di **self-attention**, che consente di modellare dipendenze a lungo raggio in modo efficiente.

Nel caso encoder–decoder:

- l'**encoder** legge l'input e costruisce rappresentazioni contestualizzate dei token;

- il **decoder** genera l'output autoregressivamente, un token alla volta, utilizzando sia self-attention sui token già generati sia **cross-attention** sulle rappresentazioni prodotte dall'encoder.

Questo schema è particolarmente adatto alla summarization, poiché consente al decoder di "attendere" alle parti più rilevanti dell'input durante la generazione.

2.3 BART: denoising pre-training per seq2seq

BART (*Bidirectional and Auto-Regressive Transformers*) è un modello encoder–decoder che combina un encoder bidirezionale (stile BERT) con un decoder autoregressivo (stile GPT). L'idea centrale è un pre-training *denoising*: il modello viene addestrato a ricostruire un testo originale a partire da una versione corrotta (ad esempio con masking, permutazioni o eliminazioni). Questa strategia rende BART efficace in compiti generativi condizionati, tra cui summarization, traduzione e semplificazione.

Nel nostro progetto, BART viene utilizzato come base architetturale per la summarization astrattiva: a partire dal modello pre-addestrato, eseguiamo un adattamento supervisionato sul dataset considerato (fine-tuning), in modo da specializzarlo alla produzione di riassunti coerenti con lo stile e la distribuzione dei riferimenti.

2.4 Fine-tuning supervisionato

Il *fine-tuning* è una procedura in cui un modello pre-addestrato viene ulteriormente addestrato su un dataset specifico per un task target. Nel caso della summarization supervisionata, ogni esempio consiste in una coppia (x, y) dove x è il documento (articolo) e y è il riassunto di riferimento.

L'obiettivo di training è massimizzare la probabilità del riassunto y condizionata al documento x. In pratica si minimizza la **cross-entropy** token-level (equivalente alla massimizzazione della log-verosimiglianza):

$$\mathcal{L} = -\sum_{t=1}^{|y|} \log p_\theta(y_t \mid y_{<t}, x), \tag{1}$$

dove y_t è il token t-esimo del riassunto e $y_{<t}$ indica i token precedenti già generati. Il fine-tuning consente al modello di apprendere:

- quali informazioni selezionare come più rilevanti (contenuto);
- come comprimerle nello spazio di output (sintesi);
- lo stile tipico dei riassunti del dataset (forma).

Dal punto di vista pratico, il fine-tuning richiede scelte di iperparametri (learning rate, batch size effettivo, epoche) e una gestione adeguata dei vincoli di lunghezza dell'input. In presenza di documenti lunghi, è comune adottare strategie come truncation o segmentazione in chunk per rispettare il limite massimo di token del modello.

2.5 Baseline estrattiva Lead-3

La prima baseline è **Lead-3**, una strategia molto semplice ma spesso competitiva nel dominio news. L'idea si basa sulla struttura a *piramide invertita* tipica degli articoli giornalistici: le informazioni essenziali tendono a comparire nelle prime frasi.

Operativamente, Lead-3 costruisce il riassunto concatenando le prime tre frasi del documento. Il metodo è:

- **deterministico** e computazionalmente leggero;
- ad **alta fedeltà** (tutto l'output proviene dal testo sorgente);
- facilmente **interpretabile**.

Il limite principale è che non esegue vera compressione semantica: se le frasi iniziali contengono dettagli secondari o se l'articolo non segue la struttura classica, la copertura dei punti chiave può risultare subottimale.

2.6 Baseline estrattiva TF–IDF + similarità coseno

La seconda baseline estrattiva utilizza una misura di **salienza lessicale** per selezionare frasi rappresentative. L'approccio si basa su TF–IDF, una tecnica classica che pesa i termini in base alla loro importanza nel documento: parole frequenti nel testo ma rare nel linguaggio generale tendono ad avere peso maggiore.

In pratica:

1. si segmenta l'articolo in frasi;
2. ogni frase viene rappresentata tramite vettori TF–IDF;
3. si selezionano le frasi più simili (tramite similarità coseno) alla rappresentazione complessiva del documento.

Il riassunto ottenuto è estrattivo e interpretabile, perché è composto da frasi effettivamente presenti nel testo e motivate da un criterio quantitativo. Tuttavia, poiché la selezione privilegia la centralità lessicale, può capitare che le frasi scelte siano ridondanti o poco coese se non sono contigue o se enfatizzano aspetti non centrali nella struttura narrativa dell'articolo.

2.7 Metriche di valutazione: ROUGE

Per confrontare in modo quantitativo i diversi metodi di summarization utilizziamo ROUGE, una famiglia di metriche che misura quanto un riassunto generato sia simile a un riassunto di riferimento in termini di **contenuto lessicale condiviso**. L'idea di base è semplice: se due riassunti comunicano informazioni simili, allora tenderanno a condividere parole e sequenze di parole.

Nel dettaglio, consideriamo le varianti più comuni:

- **ROUGE-1**, basata sulla sovrapposizione di singole parole (unigrammi), utile per stimare la copertura dei concetti principali;

- **ROUGE-2**, basata sulla sovrapposizione di coppie di parole consecutive (bigrammi), più sensibile alla qualità locale della formulazione e all'ordine delle parole;
- **ROUGE-L** e **ROUGE-Lsum**, che sfruttano la *Longest Common Subsequence* per catturare similarità di struttura preservando parzialmente l'ordine, risultando adatte anche a riassunti composti da più frasi.

ROUGE è particolarmente adatta in un contesto sperimentale perché è **facile da calcolare**, consente confronti diretti tra sistemi e rappresenta uno standard consolidato nei benchmark di summarization. Nel nostro progetto, l'uso congiunto delle diverse varianti permette di osservare aspetti complementari: ROUGE-1 è più legata alla copertura lessicale, mentre ROUGE-2 e ROUGE-L/Lsum sono più indicative di coerenza e somiglianza strutturale.

Limiti di ROUGE. È importante sottolineare che ROUGE non misura direttamente la qualità del riassunto in senso umano: essendo basata su overlap, può penalizzare **parafrasi corrette** che esprimono le stesse informazioni con parole diverse. Inoltre, non valuta in modo esplicito proprietà cruciali come **leggibilità**, **coerenza discorsiva** e **accuratezza fattuale**. Un ulteriore aspetto rilevante è che, nel dominio news, metodi estrattivi (ad es. Lead-3) possono ottenere valori competitivi perché i riferimenti condividono spesso espressioni con le prime frasi dell'articolo.

Per questi motivi, oltre ai punteggi ROUGE, viene condotta una valutazione qualitativa su un piccolo numero di esempi, confrontando visivamente output e riferimento. Questa fase è utile per interpretare i risultati quantitativi e osservare fenomeni non pienamente catturati da metriche automatiche, quali coerenza discorsiva, ridondanza e differenze stilistiche tra approcci estrattivi e astrattivi.

3 Dataset e preprocessing

Per gli esperimenti utilizziamo il benchmark **CNN/DailyMail** (v3.0.0), ampiamente impiegato nella letteratura sulla summarization. Il dataset è composto da articoli giornalistici (testo sorgente) e dai corrispondenti riassunti redazionali sotto forma di *highlights*, utilizzati come riferimento (*gold summary*) per l'addestramento e la valutazione.

Ogni istanza è rappresentata dai campi:

- `article`: testo completo dell'articolo;
- `highlights`: riassunto di riferimento;
- `id`: identificativo univoco.

Il dataset è suddiviso in tre split standard: `train`, `validation` e `test`. Nel progetto adottiamo `train` e `validation`, come descritto nelle sezioni successive.

3.1 Costruzione dei subset e riproducibilità

Per rendere l'esperimento computazionalmente sostenibile e riproducibile, non utilizziamo l'intero dataset ma costruiamo subset controllati mediante:

1. *shuffle* deterministico con seed fissato (`SEED = 42`);
2. selezione dei primi elementi dopo lo shuffle (`select(range(...))`).

In particolare:

- **Training subset**: $N_{\text{train}} = 20\,000$ esempi estratti da `train`, utilizzati per il fine-tuning;
- **Validation subset (per training)**: $N_{\text{val-ft}} = 1\,000$ esempi estratti da `validation`, utilizzati per controllare l'andamento dell'addestramento;
- **Evaluation subset (finale)**: $N_{\text{eval}} = 500$ esempi estratti da `validation`, utilizzati per il confronto comparativo finale tra i metodi.

Questa scelta consente di lavorare su un campione sufficientemente ampio per ottenere risultati significativi, mantenendo però tempi di esecuzione compatibili con un progetto universitario e garantendo che gli esperimenti siano replicabili.

3.2 Tokenizzazione e vincoli di lunghezza

Il fine-tuning e l'inferenza di un modello encoder–decoder richiedono la conversione del testo in token. Utilizziamo il tokenizer associato al modello scelto, applicando vincoli di lunghezza per rispettare i limiti computazionali e stabilizzare il training:

- **Input (articolo)**: massimo 512 token (`MAX_SOURCE_LEN = 512`);
- **Target (riassunto)**: massimo 128 token (`MAX_TARGET_LEN = 128`).

Quando una sequenza eccede il limite imposto, applichiamo **truncation**. Questa scelta introduce un compromesso: si ottiene un processo più stabile e veloce, ma si può perdere parte dell'informazione nella porzione finale dell'articolo. Nel dominio news ciò è spesso accettabile perché molte informazioni rilevanti compaiono nelle prime parti del testo.

3.3 Preparazione dei dati per l'addestramento

Una volta definito come rappresentare articoli e riassunti, i dati vengono trasformati in un formato adatto al training del modello: l'articolo rappresenta l'*input* e gli *highlights* rappresentano l'*output atteso*. In altre parole, durante l'addestramento il modello impara a produrre un riassunto che assomigli il più possibile a quello di riferimento, osservando molte coppie articolo–riassunto.

Questa fase è importante perché rende coerente l'intero workflow: la stessa rappresentazione utilizzata per addestrare il modello viene poi riutilizzata nella fase di generazione e nella valutazione finale, evitando discrepanze tra training e test.

Le decisioni prese in questa sezione (uso di subset, seed fisso, vincoli di lunghezza) rispondono a due esigenze principali:

- **Riproducibilità**: a parità di seed e parametri, il campionamento e i risultati sono replicabili.
- **Sostenibilità computazionale**: l'uso di subset e di limiti di lunghezza rende il fine-tuning realizzabile in tempi contenuti senza compromettere l'obiettivo di confronto tra metodi.

In sintesi, il dataset viene preparato in modo da supportare sia l'addestramento del modello astrattivo sia il confronto comparativo con le baseline estrattive, mantenendo l'esperimento gestibile e ben documentato.

3.4 Gestione di articoli lunghi: chunking token-based e generazione gerarchica

Un aspetto pratico rilevante è la gestione della lunghezza degli articoli. I modelli transformer hanno un limite massimo di token gestibili in input; per evitare perdita eccessiva di informazione, in fase di generazione utilizziamo una strategia basata su:

- **chunking token-based**: l'articolo viene suddiviso in segmenti (chunk) di dimensione controllata;
- **generazione gerarchica**: si genera un riassunto per ciascun chunk e successivamente un riassunto finale a partire dalla concatenazione dei riassunti parziali.

Questa procedura permette di ottenere riassunti anche quando il testo supera i limiti di input, preservando informazione distribuita lungo l'articolo. Inoltre, per evitare output sproporzionati, adottiamo un controllo dinamico della lunghezza massima del riassunto, impostandola in modo proporzionale alla lunghezza dell'input considerato. In sintesi, il confronto è strutturato per mettere in evidenza trade-off complementari:

- **BART fine-tunato (abstractive)**: maggiore capacità di sintesi e riformulazione, potenzialmente più coerenza e compressione.
- **Lead-3 (extractive)**: massima semplicità e fedeltà, spesso competitivo nelle news.
- **TF–IDF (extractive)**: selezione guidata da salienza lessicale, interpretabile ma talvolta frammentaria.

Queste differenze verranno quantificate tramite ROUGE e discusse anche tramite esempi qualitativi nella sezione dei risultati.

4 Setup sperimentale

Questa sezione descrive l'ambiente di esecuzione e le principali scelte sperimentali adottate per l'addestramento e la valutazione, con l'obiettivo di garantire un confronto **replicabile** e computazionalmente sostenibile.

4.1 Ambiente computazionale

Gli esperimenti sono stati eseguiti in ambiente **Google Colab** con accelerazione **GPU NVIDIA A100** (40GB). L'uso della GPU è fondamentale per rendere il fine-tuning di un modello Transformer praticabile in tempi compatibili con un progetto universitario, soprattutto nelle fasi di training e generazione dei riassunti.

4.2 Impostazioni di training

Il fine-tuning è condotto sul subset di training ($N_{\text{train}} = 20\,000$) con monitoraggio su un subset di validation dedicato ($N_{\text{val-ft}} = 1\,000$). La durata dell'addestramento è fissata a **1 epoca**, scelta conservativa che consente di:

- verificare la capacità del modello di adattarsi al task con tempi contenuti;
- evitare overfitting su un subset non completo del dataset;
- mantenere l'esperimento ripetibile e facilmente estendibile (ad es. aumentando epoche o dimensione del training set).

A livello di iperparametri, viene utilizzato un **learning rate** pari a $2 \cdot 10^{-5}$, valore tipicamente stabile per il fine-tuning di modelli transformer. Per bilanciare efficienza e memoria, viene impiegato un batch per device pari a 4 con **gradient accumulation** (4 step), ottenendo un batch effettivo equivalente più ampio senza eccedere i limiti di VRAM. Infine, ove disponibile, viene abilitata la **mixed precision** (`fp16`), che riduce il costo computazionale e accelera il training.

5 Risultati sperimentali

In questa sezione presentiamo i risultati ottenuti dal confronto tra il modello fine-tunato e le due baseline estrattive. La valutazione quantitativa è condotta sul subset di *validation* ($N_{\text{eval}} = 500$) utilizzando le metriche ROUGE, mentre l'interpretazione è supportata da un'analisi qualitativa su esempi selezionati.

5.1 Analisi quantitativa (ROUGE)

La Tabella 1 riporta i punteggi ROUGE ottenuti dai tre sistemi. Nel complesso:

- il modello fine-tunato ottiene valori competitivi e risulta particolarmente solido su ROUGE-2 e ROUGE-L/Lsum;
- Lead-3 risulta molto competitivo, soprattutto su ROUGE-1, confermando che nei testi giornalistici le prime frasi contengono spesso gran parte dell'informazione rilevante;
- TF–IDF mostra prestazioni inferiori in tutte le metriche, coerentemente con i limiti tipici di metodi estrattivi basati su salienza lessicale (possibile frammentarietà e ridondanza).

Tabella 1: Punteggi ROUGE sul subset di evaluation ($N_{\text{eval}} = 500$).

Sistema	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-Lsum**
BART fine-tunato (`facebook/bart-base`)	0.4164	0.1968	0.2883	0.3850
Baseline Lead-3	0.4181	0.1935	0.2649	0.3471
Baseline TF–IDF (extractive)	0.3671	0.1538	0.2419	0.3138

L'andamento dei punteggi suggerisce un comportamento coerente con la natura dei metodi:

- **Lead-3** raggiunge il valore più alto su ROUGE-1, indicando un'elevata copertura lessicale del riferimento. Ciò è plausibile nel dominio news, dove gli highlights condividono spesso termini e concetti presenti nell'incipit dell'articolo.
- Il **modello fine-tunato** ottiene il miglior risultato su ROUGE-2 e migliora sensibilmente ROUGE-L e ROUGE-Lsum rispetto alle baseline. Questo pattern è compatibile con una maggiore capacità di produrre un riassunto più *strutturato* e con una migliore coerenza locale (bigrammi) rispetto a metodi puramente estrattivi.
- La baseline **TF–IDF** risulta sistematicamente inferiore: la selezione guidata dalla centralità lessicale può individuare frasi informative, ma non garantisce coesione narrativa e può includere dettagli ridondanti o non prioritari rispetto allo stile del riassunto di riferimento.

In sintesi, i risultati indicano che il modello fine-tunato riesce a bilanciare meglio *sintesi* e *organizzazione del contenuto*, mentre Lead-3 rimane un baseline forte in questo dominio per la sua capacità di catturare rapidamente le informazioni principali.

L'ispezione manuale di esempi conferma i trade-off osservati quantitativamente. In diversi casi il modello fine-tunato produce riassunti più compatti e scorrevoli, riaggregando le informazioni in poche frasi. Lead-3 mantiene alta fedeltà al testo sorgente e risulta spesso molto informativo, ma può essere più lungo e includere dettagli secondari. TF–IDF, pur selezionando frasi "centrali" dal punto di vista lessicale, può risultare più ridondante o meno coerente quando le frasi selezionate non sono contigue.

Queste osservazioni supportano l'interpretazione dei punteggi ROUGE e motivano l'utilità di affiancare una valutazione qualitativa a quella automatica.

6 Analisi qualitativa

Accanto alla valutazione automatica, è stata condotta un'analisi qualitativa su un sottoinsieme di esempi, confrontando direttamente: (i) il riassunto di riferimento (*highlights*) e (ii) gli output dei tre sistemi (modello fine-tunato, Lead-3 e TF–IDF). L'obiettivo è evidenziare differenze non completamente catturate da ROUGE, come coerenza discorsiva, ridondanza, copertura informativa e stile.

6.1 Esempio 1: trasferimento di Jarryd Hayne ai San Francisco 49ers

Nel primo esempio, il riferimento enfatizza tre elementi: la scelta di Hayne di lasciare la NRL, la firma del contratto con i 49ers e la reazione positiva della US Association of Rugby League. Il modello fine-tunato produce un riassunto molto compatto che mantiene i due fatti principali (abbandono della NRL e contratto triennale) ma omette il dettaglio sul *welcoming* dell'associazione. Lead-3 include più contesto e citazioni, risultando informativo ma meno sintetico; TF–IDF tende a ripetere l'informazione sul contratto e aggiunge dettagli sportivi (ruoli/posizioni) non presenti nel riferimento. Questo caso evidenzia il trade-off tra **sintesi** (modello neurale) e **copertura dettagliata** (estrattivi), nonché il rischio di **ridondanza** nei metodi basati su similarità lessicale.

6.2 Esempio 2: caso clinico e recupero tramite attività imprenditoriale

Nel secondo esempio, i riferimenti includono dettagli specifici (peso, dieta basata su solo caffè, collasso, trattamento ospedaliero) e l'avvio di un'attività (pasticceria) come parte del percorso di recupero. Il modello fine-tunato riesce a condensare in poche frasi la sequenza causale e l'esito, mantenendo le informazioni più importanti e una struttura scorrevole. Lead-3 è molto completo (include anche il contesto familiare), ma risulta più lungo e meno "riassuntivo". TF–IDF seleziona frasi fortemente descrittive e ricche di dettagli, ma tende a produrre un output più esteso e meno focalizzato, includendo elementi accessori (citazioni e dettagli aggiuntivi) che riducono la concisione. Questo esempio mostra come il modello neurale possa fornire maggiore **compressione** preservando comunque l'informazione essenziale.

6.3 Esempio 3: serie TV "Community"

Nel terzo esempio, il riferimento è molto generale e valuta la qualità della serie ("weirdly hilarious", "critics and fans loved the premiere"). L'output del modello fine-tunato introduce dettagli specifici (orario di uscita, piattaforma, cambiamenti nel cast) che possono essere coerenti con l'articolo ma non necessariamente riflessi nel testo del riferimento, riducendo l'overlap lessicale con gli highlights. Lead-3 e TF–IDF, essendo estrattivi, tendono invece a riprendere porzioni del testo più vicine allo stile e alla formulazione del riferimento. Questo caso è utile per comprendere un limite tipico delle metriche basate su overlap: una generazione informativa e plausibile può essere penalizzata se non condivide le stesse scelte lessicali del riferimento.

6.4 Esempio 4: "Dancing Man" e campagna social

Nel quarto esempio, il riferimento include molti dettagli narrativi (due foto, umiliazione, origine su 4chan, identificazione, invito a evento). Il modello fine-tunato produce un riassunto relativamente compatto che cattura la dinamica principale e l'esito (campagna e invito), mentre Lead-3 include un contesto più ampio ma si dilunga nella descrizione iniziale. TF–IDF, concentrandosi su frasi con parole salienti, seleziona

soprattutto la parte finale dell'articolo (inviti, tweet e dettagli della campagna), risultando meno bilanciato rispetto alla struttura del riferimento. L'esempio conferma che i metodi estrattivi possono essere sensibili alla distribuzione dei termini e non sempre selezionano una sequenza narrativa coerente.

6.5 Considerazioni generali

Nel complesso, l'analisi qualitativa conferma i risultati quantitativi:

- il modello fine-tunato tende a produrre output più **compatti** e **coerenti**, spesso con una struttura più simile a un vero riassunto;
- Lead-3 risulta competitivo nel dominio news grazie alla forte informatività dell'incipit, ma può risultare meno conciso;
- TF–IDF è interpretabile ma può generare riassunti più **lunghi**, **ridondanti** o sbilanciati verso porzioni del testo con termini ad alto peso.

Queste evidenze motivano l'uso congiunto di valutazione automatica e qualitativa per caratterizzare in modo più completo i trade-off tra approcci estrattivi e astrattivi.

7 Conclusioni

In questo progetto abbiamo sviluppato e valutato una pipeline completa per la *text summarization* sul dataset **CNN/DailyMail** (v3.0.0), confrontando un approccio *abstractive* basato su un modello encoder–decoder **fine-tunato** con due baseline *extractive* semplici e interpretabili (**Lead-3** e **TF–IDF + similarità coseno**). L'intero workflow è stato progettato in modo riproducibile (seed fissato) e computazionalmente sostenibile tramite l'uso di subset controllati per training, validazione e valutazione.

I risultati quantitativi mostrano che il modello fine-tunato raggiunge prestazioni competitive e supera le baseline su metriche sensibili alla coerenza locale e alla struttura del riassunto (in particolare ROUGE-2 e ROUGE-L/Lsum), evidenziando una maggiore capacità di sintesi e riorganizzazione del contenuto. Al contempo, la baseline Lead-3 risulta molto competitiva, specialmente su ROUGE-1, confermando che nel dominio news le informazioni principali sono spesso concentrate nelle frasi iniziali dell'articolo. L'approccio TF–IDF, pur essendo interpretabile e totalmente estrattivo, mostra prestazioni inferiori, coerentemente con la tendenza a produrre riassunti più lunghi, ridondanti o meno coesi.

L'analisi qualitativa supporta e chiarisce tali evidenze: il modello fine-tunato tende a generare output più compatti e scorrevoli, mentre Lead-3 privilegia fedeltà e completezza informativa a scapito della concisione. TF–IDF seleziona frasi lessicalmente salienti ma non sempre garantisce continuità narrativa, con possibili ripetizioni o sbilanciamenti verso porzioni specifiche del testo.

Nel complesso, il lavoro evidenzia come approcci estrattivi e astrattivi presentino vantaggi complementari: da un lato semplicità e interpretabilità (Lead-3, TF–IDF), dall'altro maggiore capacità di compressione e costruzione di un riassunto più strutturato (modello fine-tunato). L'uso congiunto di valutazione automatica (ROUGE) e ispezione qualitativa si è rivelato essenziale per fornire una lettura più completa delle prestazioni e dei comportamenti dei diversi metodi.