RELAZIONE:
Text-Summarizer

Enrico Ferraiolo 0001191698

Laurea Magistrale in Informatica

Corso: Natural Language Processing
a.a. 2024-2025

Indice

1 Introduzione

Questo progetto vuole implementare un modello di text summarization (riassunto dei testi) utilizzando un'architettura Sequence-to-Sequence (Seq2Seq) basata su reti LSTM (Long Short-Term Memory).
L'obiettivo principale è generare riassunti concisi e significativi a partire da recensioni di prodotti più lunghe, mantenendo il significato del testo originale.

2 Dataset

Per questo progetto è stato utilizzato il dataset SNAP Amazon Fine Food Reviews, che contiene recensioni di prodotti alimentari di Amazon.
In particolare, il dataset contiene, per ogni riga, una recensione completa e il rispettivo riassunto.
Del dataset originale, composto da circa 500.000 righe, è stato selezionato un sottoinsieme di 10.000 righe per l'analisi e l'allenamento del modello.

3 Preprocessing dei Dati

Il preprocessing dei dati è una fase critica per garantire la qualità e l'efficacia del modello di summarization, infatti è fondamentale pulire e filtrare i dati in modo accurato.
Sul dataset, infatti, sono stati eseguiti diversi passaggi di pulizia e filtraggio dei dati per garantire qualità e coerenza al modello durante l'addestramento.
Vediamo di seguito gli step effettutati durante questa fase:

3.1 Pulizia del Testo

Sono stati applicati i seguenti step di preprocessing:

1. **Conversione del testo in minuscolo**
   - Questa conversione garantisce l'uniformità del testo, evitando che la stessa parola venga considerata diversa solo per la presenza di maiuscole. Ad esempio, "Home", "HOME" e "home" vengono trattate come la stessa parola, riducendo la dimensionalità del vocabolario e migliorando l'efficienza dell'addestramento.
2. **Rimozione dei tag HTML**
   - Le recensioni potrebbero contenere tag HTML residui dal formato web originale. Questi elementi non contribuiscono al significato semantico del testo e potrebbero interferire con l'apprendimento del modello, pertanto vengono rimossi.
3. **Espansione delle contrazioni**
   - Le contrazioni nella lingua inglese (come "don't", "I'm", "we're") vengono espanse nelle loro forme complete ("do not", "I am", "we are"). Questo processo vuole standardizzare e garantire coerenza in tutto il testo e aiuta il modello a catturare meglio le relazioni semantiche, eliminando variazioni non necessarie della stessa espressione.

4. **Rimozione degli apostrofi possessivi ('s)**

    - La forma possessiva in inglese non altera sostanzialmente il significato della frase ai fini del riassunto.
      La sua rimozione semplifica il testo e riduce ulteriormente la dimensione del vocabolario, permettendo al modello di concentrarsi sui concetti principali.

5. **Eliminazione del testo tra parentesi**

    - Il testo tra parentesi spesso contiene informazioni supplementari che non sono generalmente essenziali per il riassunto.
      La loro rimozione aiuta a mantenere il focus sulle informazioni principali della recensione.

6. **Rimozione della punteggiatura e caratteri speciali**

    - La punteggiatura e i caratteri speciali, pur essendo importanti per la leggibilità umana, possono introdurre rumore nell'addestramento del modello.
      La loro rimozione semplifica il testo mantenendo intatto il contenuto semantico essenziale per la generazione del riassunto.

7. **Eliminazione delle stopwords**

    - Le stopwords sono parole molto comuni (come "the", "is", "at", "which") che appaiono frequentemente ma portano poco significato semantico.
      La loro rimozione riduce significativamente la dimensionalità del problema senza perdere informazioni cruciali per il riassunto, permettendo al modello di concentrarsi sulle parole più significative.

8. **Rimozione delle parole troppo corte**

    - Le parole molto corte (solitamente di una o due lettere) spesso non contribuiscono al significato del testo.
      La loro rimozione aiuta a ridurre ulteriormente il rumore nei dati, mantenendo solo i termini più significativi per l'analisi.

3.2 Filtraggio dei Dati

Dopo l'analisi statistica del dataset, sono stati applicati i seguenti vincoli:

- Lunghezza massima delle recensioni: 30 parole
- Lunghezza massima dei riassunti: 8 parole

Questi limiti sono stati determinati attraverso un'analisi statistica della distribuzione delle lunghezze nel dataset, come possiamo vedere nella figura 1.

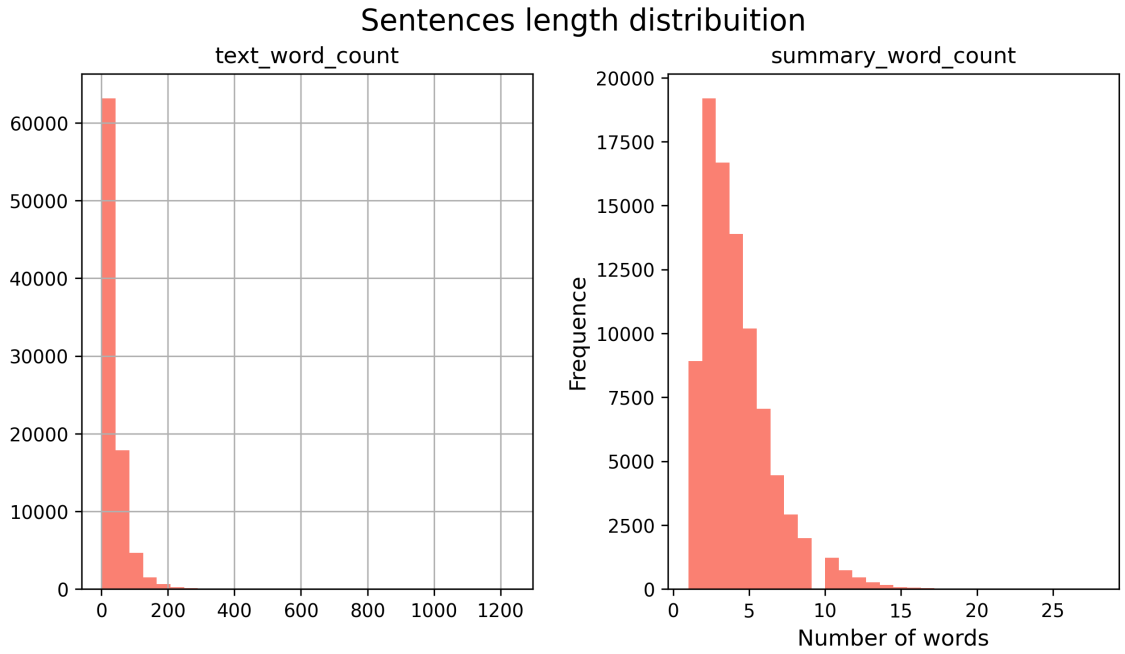


Figura 1: A sinistra la distribuzione delle lunghezze delle recensioni, a destra la distribuzione delle lunghezze dei riassunti

Infatti, come si può notare dai due grafici, la maggior parte delle recensioni e dei riassunti ha lunghezze inferiori ai limiti stabiliti, quindi questi vincoli permettono di mantenere la maggior parte dei dati del dataset.

3.3 Tokenizzazione e Token Speciali

Per preparare i dati per il modello ho aggiunto i token speciali `"sostok"` e `"eostok"` per indicare l'inizio e la fine di una sequenza, in modo da facilitare la tokenizzazione e l'addestramento del modello.
Inoltre, ho effettuato la tokenizzazione separata per le recensioni (testo di input) e i riassunti (testo di output) per garantire che il modello possa apprendere correttamente la relazione tra i due. I due tokenizer servono a creare il vocabolario per le recensioni e per i riassunti, in modo da poter convertire i testi in sequenze di token.

4 Architettura del Modello

L'implementazione del modello è stata effettuata attraverso una classe astratta `BaseModel` e una classe derivata `LSTM`.
Questo permette di definire un'interfaccia comune per tutti i modelli di summarization e di estendere facilmente l'architettura in futuro.

4.1 Classe Base Astratta

La classe `BaseModel` fornisce l'interfaccia base per tutti i modelli di summarization:

- Metodi astratti per costruire encoder e decoder.
- Funzionalità per il salvataggio, caricamento e inferenza del modello.
- Conversione tra sequenze e testo tramite i tokenizzatori.

4.2 Implementazione Seq2SeqLSTM

La classe `Seq2SeqLSTM` implementa l'architettura specifica per il modello di summarization Sequence to Sequence con layer LSTM.
Vediamo di seguito le caratteristiche principali dell'architettura, composta da encoder e decoder:

4.2.1 Encoder

L'encoder è composto da:

- **Layer di embedding**: mappa i token di input in vettori di lunghezza fissa
- **Tre layer LSTM** con:
  - Dimensione latente fissa
  - Dropout del 40%
  - Recurrent dropout del 20%

4.2.2 Decoder

Il decoder include:

- **Layer di embedding**: mappa i token di output in vettori di lunghezza fissa
- **Layer LSTM** con:
  - Stessa dimensione latente dell'encoder
  - Dropout del 40%
  - Recurrent dropout del 20%
- **Layer di attention**: calcola i pesi di attenzione tra l'encoder e il decoder
- **Layer denso di output**: questo layer restituisce la distribuzione di probabilità sul vocabolario per la generazione delle parole del riassunto.
  Utilizza la funzione di attivazione softmax per la normalizzazione delle probabilità.

Di seguito, nella figura 2, possiamo vedere un diagramma dell'architettura del modello:

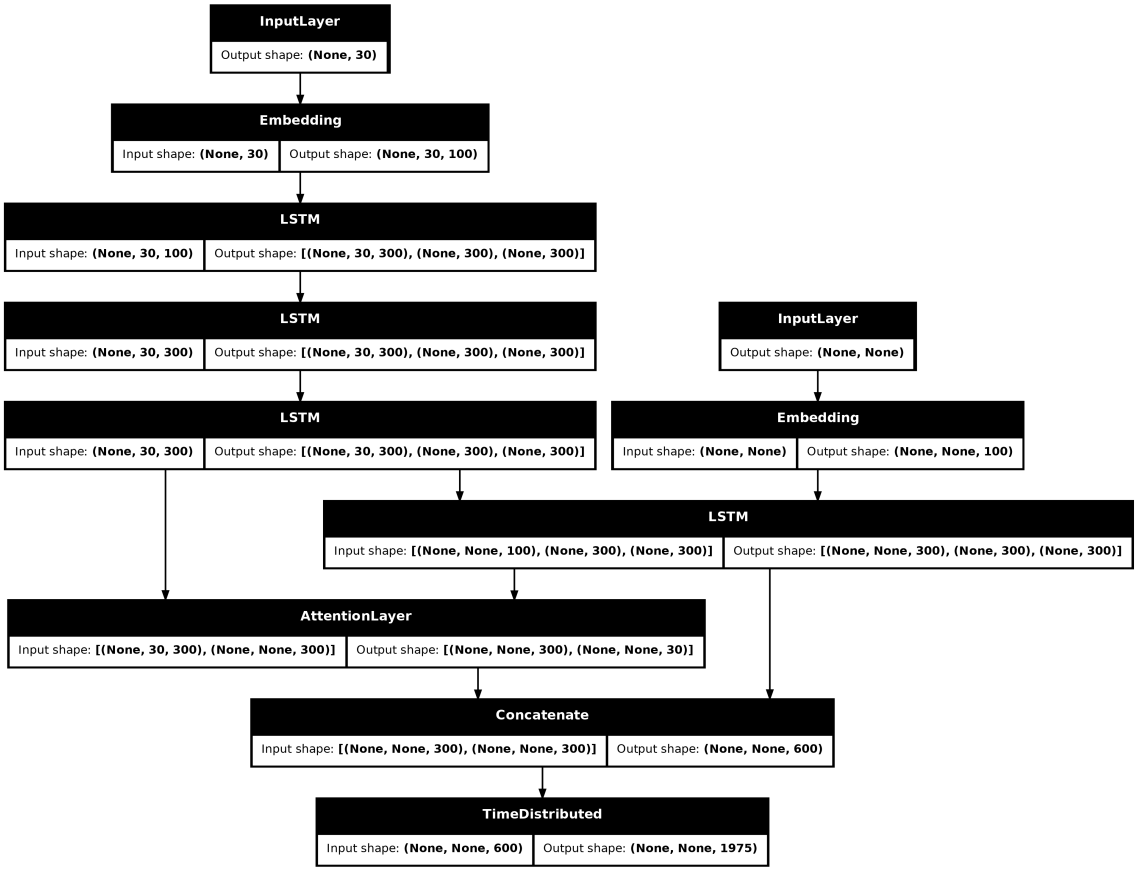


Figura 2: Diagramma dell'architettura del modello

5 Training del Modello

L'addestramento del modello è stato effettuato utilizzando il dataset preprocessato. Prima di iniziare l'addestramento, il dataset è stato suddiviso in training set e validation set, con una proporzione del 90% e 10% rispettivamente.
A questo punto ho definito una funzione di early stopping per monitorare la loss sul validation set e fermare l'addestramento quando la loss non diminuisce per un certo numero di epoche, ciò per evitare l'overfitting.
Dopodiché sono passato alla fase effettiva di training del modello, utilizzando l'ottimizzatore `rmsprop` e la loss function `categorical crossentropy`.

5.1 Loss

Alla fine dell'addestramento, la loss sul training set è scesa a circa , mentre la loss sul validation set è arrivata a .
Possiamo verifcare l'andamento delle loss durante l'addestramento nella figura 3.

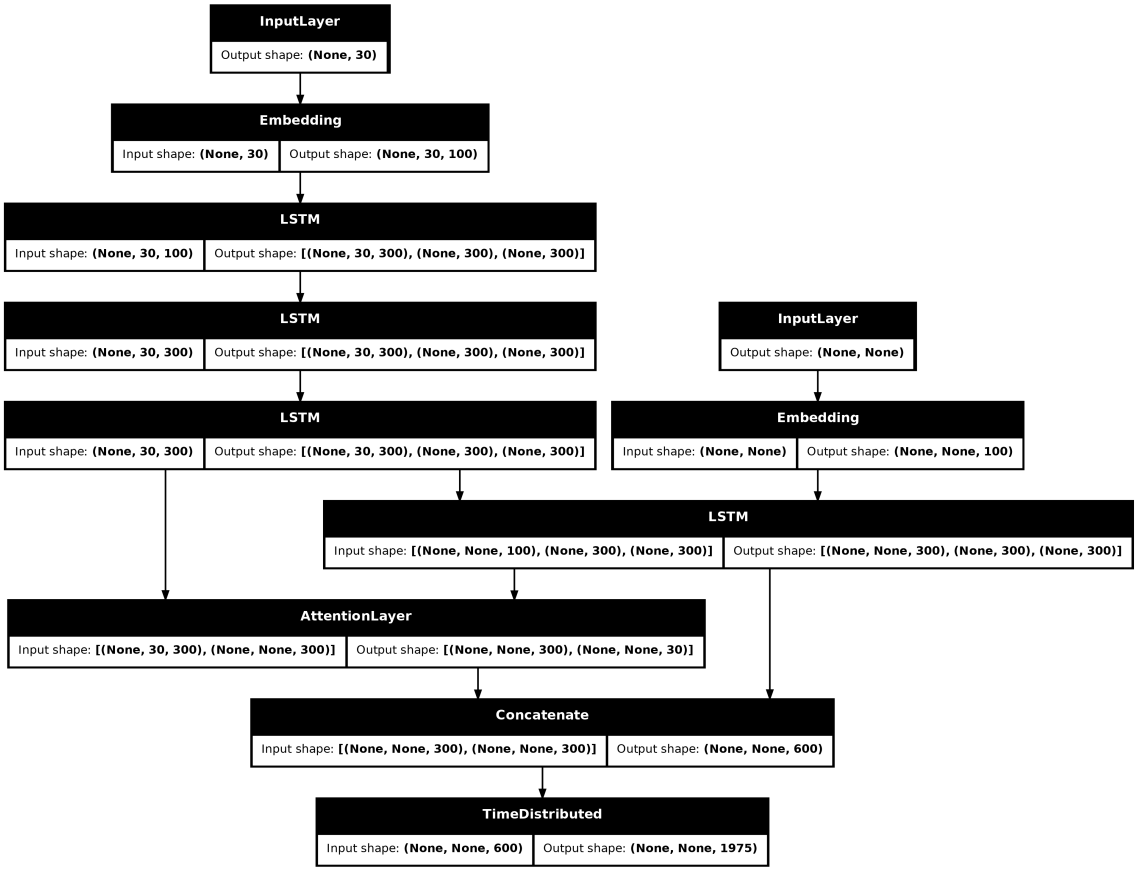


Figura 3: Andamento delle loss durante l'addestramento

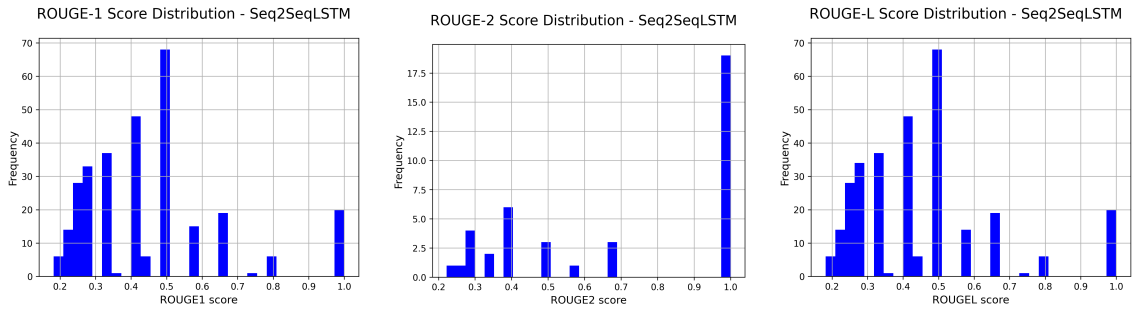
6 Metriche di Valutazione

Per valutare le performance del modello sono state utilizzate diverse metriche, utili per valutare la qualità dei riassunti generati rispetto a quelli di riferimento.
Per eseguire queste valutazioni ho utilizzato il dataset di test, che non è stato utilizzato durante l'addestramento del modello e ho fatto inferenza sui dati di test per generare i riassunti, per la precisione ho generato 1000 riassunti.

6.1 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

Sono state calcolate tre varianti di ROUGE:

- ROUGE-1: confronta unigrammi tra il riassunto generato e quello di riferimento, 4a
- ROUGE-2: considera bigrammi per valutare la similarità tra i due testi, 4b
- ROUGE-L: confronta la sottosequenza più lunga comune tra i due testi, 4c



(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-L

Figura 4: Valori ROUGE per il modello Seq2SeqLSTM: (a) ROUGE-1, (b) ROUGE-2, and (c) ROUGE-L.

6.2 Word Error Rate (WER)

Il WER è una metrica che calcola il tasso di errore tra due sequenze di parole.
In particolare, il WER calcola il numero di operazioni di inserimento, cancellazione e sostituzione necessarie per trasformare una sequenza di parole in un'altra.

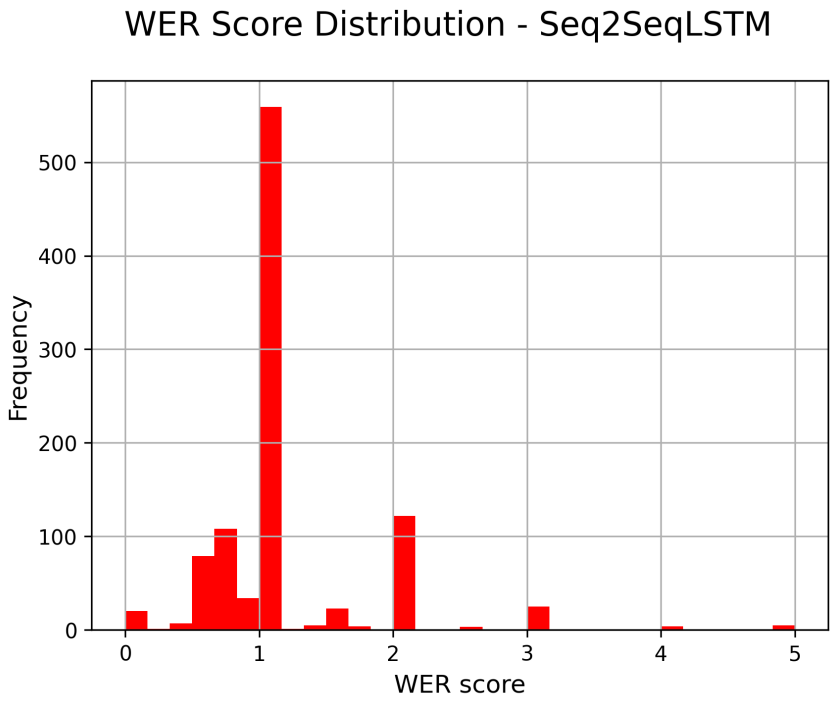


Figura 5: Valori WER per il modello Seq2SeqLSTM

6.3 Cosine Similarity

La similarità cosenica è una metrica che calcola la similarità tra due vettori in uno spazio multidimensionale.
Nel caso specifico della generazione di riassunti, la similarità cosenica è stata calcolata tra i vettori di embedding delle parole nei riassunti generati e quelli nei riassunti di riferimento, con il fine di valutare la qualità dei riassunti generati.

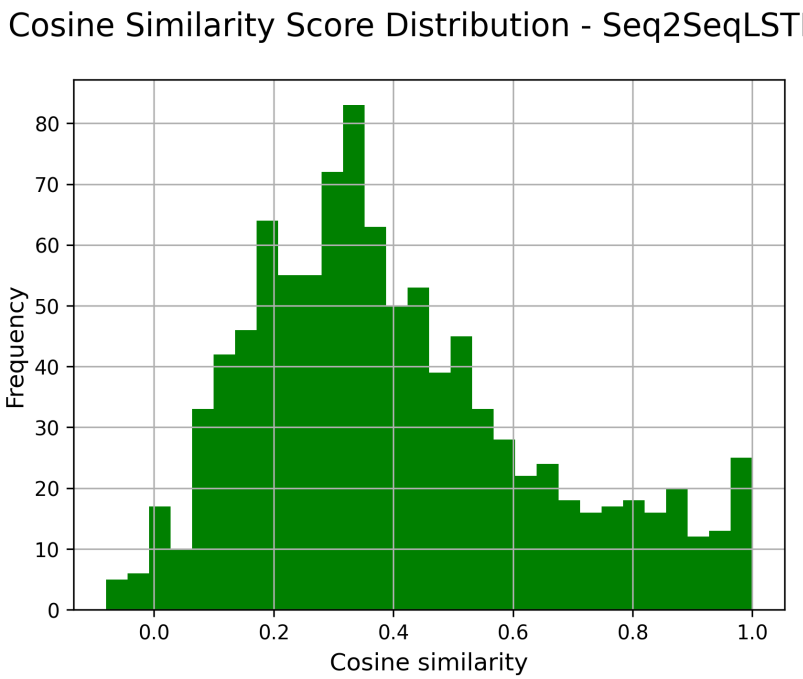


Figura 6: Valori di similarità cosenica per il modello Seq2SeqLSTM

7 Implementazione del Modello Seq2SeqBiLSTM

In questo caso, abbiamo implementato un modello di tipo Sequence-to-Sequence con una versione BiLSTM (Bidirectional LSTM).
L'architettura è simile al modello Seq2Seq LSTM, ma con un'encoder bidirezionale, il che consente di catturare informazioni contestuali da entrambe le direzioni della sequenza.

7.1 Architettura del Modello Seq2SeqBiLSTM

L'architettura del modello Seq2SeqBiLSTM include le seguenti modifiche rispetto al modello LSTM standard:

7.1.1 Encoder

L'encoder è composto da:

- **Layer di embedding**: mappa i token di input in vettori di lunghezza fissa.
- **Due layer LSTM bidirezionali** con:
  - Dimensione latente fissa.
  - Dropout del 40%.
  - Recurrent dropout del 20%.

7.1.2 Decoder

Il decoder include:

- **Layer di embedding**: mappa i token di output in vettori di lunghezza fissa.

- **Layer LSTM**: stessa dimensione latente dell'encoder.
- **Layer di attenzione**: calcola i pesi di attenzione tra l'encoder e il decoder.
- **Layer denso di output**: restituisce la distribuzione di probabilità per la generazione delle parole del riassunto.

Di seguito, possiamo vedere un diagramma dell'architettura del modello Seq2SeqBiLSTM nella figura 7.

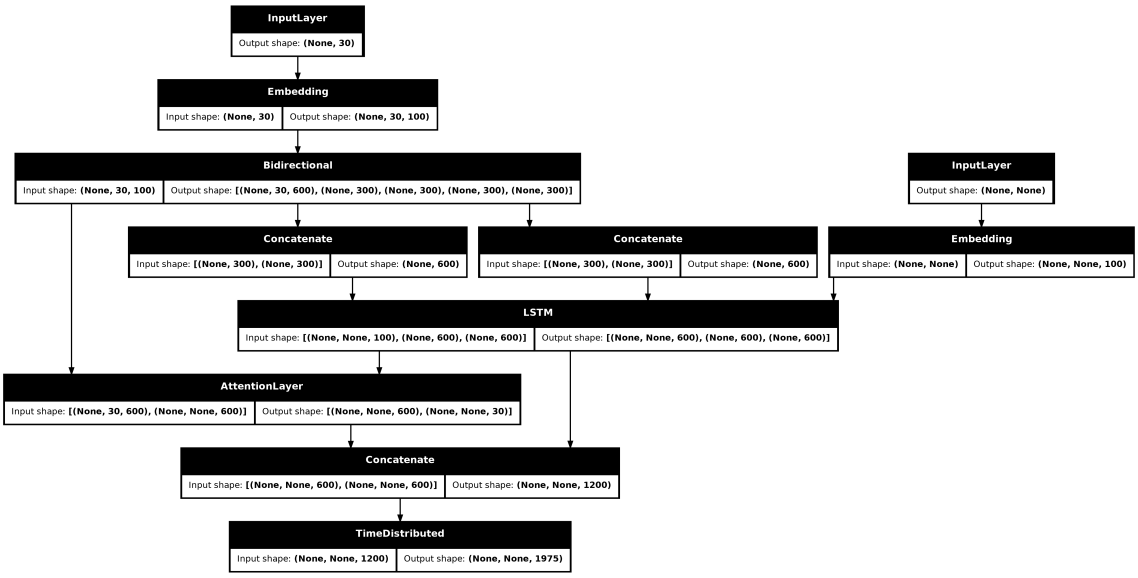


Figura 7: Diagramma dell'architettura del modello Seq2SeqBiLSTM

8 Training del Modello Seq2SeqBiLSTM

Il processo di addestramento per il modello Seq2SeqBiLSTM è stato simile a quello del modello Seq2Seq LSTM, ma con l'encoder bidirezionale, che permette al modello di considerare il contesto sia precedente che successivo per ogni parola.

8.1 Loss

Durante l'addestramento, la loss sul training set è scesa a circa [valore loss training set], mentre la loss sul validation set è arrivata a [valore loss validation set].
L'andamento della loss durante l'addestramento è mostrato nella figura 8.

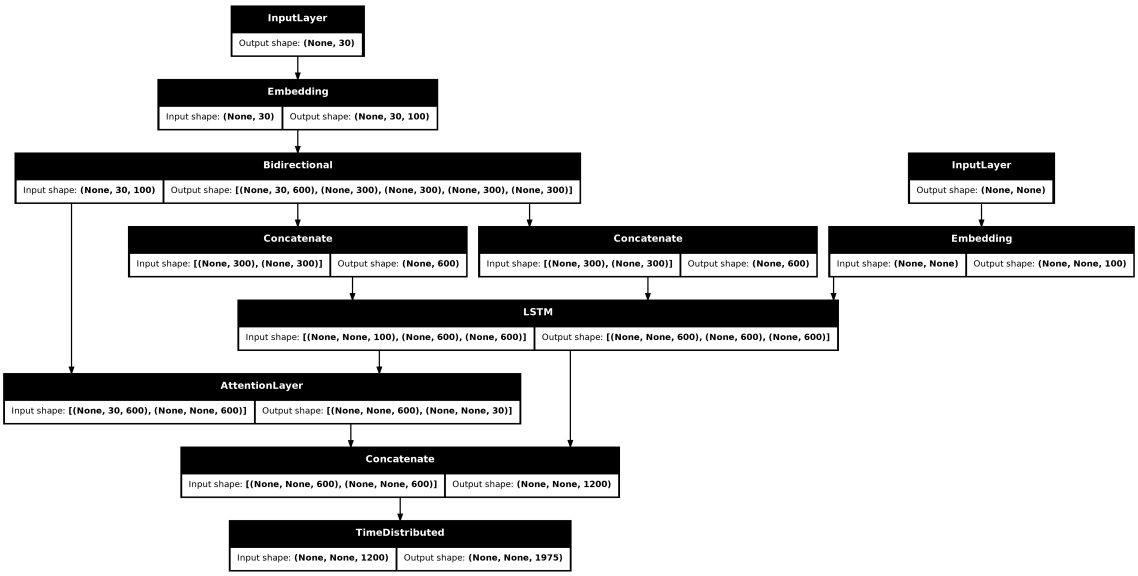


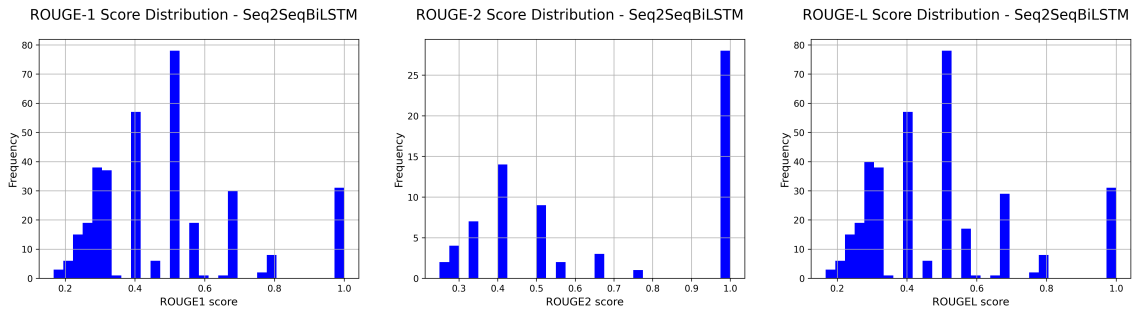
Figura 8: Andamento delle loss durante l'addestramento per il modello Seq2SeqBiLSTM

9 Metriche di Valutazione per Seq2SeqBiLSTM

Come per il modello Seq2Seq LSTM, sono state calcolate le metriche ROUGE, WER, e la Similarità Cosine per valutare le performance del modello Seq2SeqBiLSTM.

9.1 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

I valori delle metriche ROUGE per il modello Seq2SeqBiLSTM sono mostrati nelle seguenti figure:



(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-L

Figura 9: Valori ROUGE per il modello Seq2SeqBiLSTM: (a) ROUGE-1, (b) ROUGE-2, and (c) ROUGE-L.

9.2 Word Error Rate (WER)

Il WER per il modello Seq2SeqBiLSTM è stato calcolato sui riassunti generati e i riassunti di riferimento.
I valori di WER sono mostrati nella figura 10.

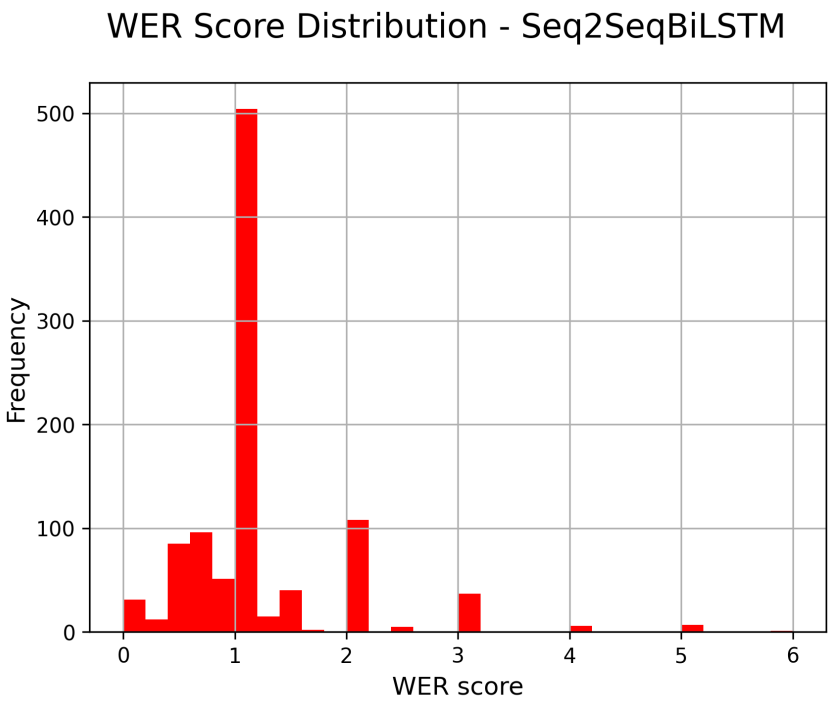


Figura 10: Valori WER per il modello Seq2SeqBiLSTM

9.3 Cosine Similarity

La similarità cosenica per il modello Seq2SeqBiLSTM è stata calcolata tra i vettori di embedding delle parole nei riassunti generati e quelli nei riassunti di riferimento. I valori di similarità cosenica sono mostrati nella figura 11.

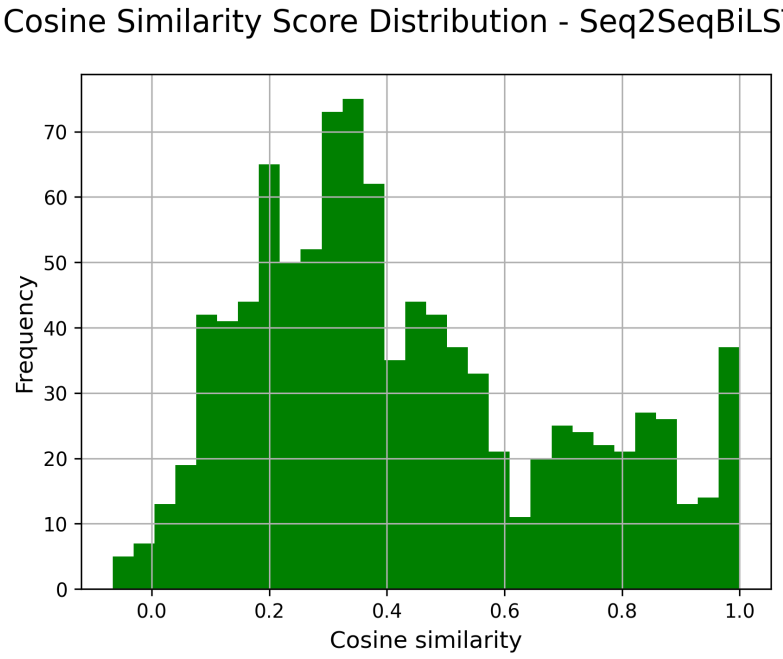


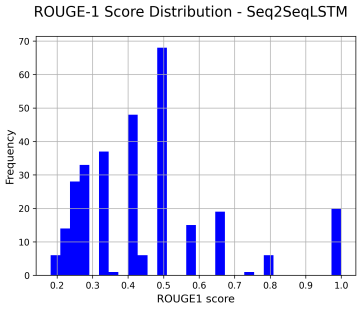
Figura 11: Valori di similarità cosenica per il modello Seq2SeqBiLSTM

10 Confronto tra Seq2SeqLSTM e Seq2SeqBiLSTM

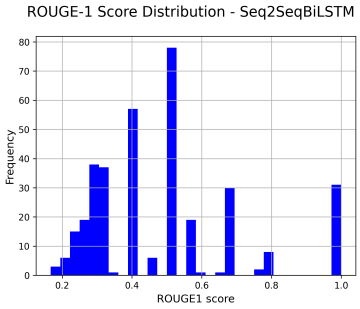
In questa sezione vengono confrontate le prestazioni dei due modelli utilizzando i grafici delle metriche di valutazione: ROUGE, WER e cosine similarity.

10.1 ROUGE Scores

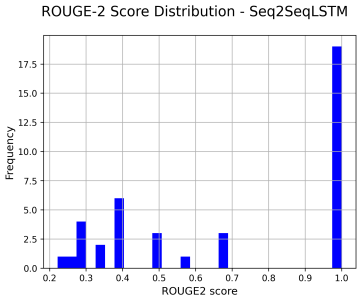
I grafici nella Figura 12 confrontano le performance in termini di ROUGE-1, ROUGE-2 e ROUGE-L per i due modelli. Il modello Seq2SeqBiLSTM mostra un miglioramento nei punteggi ROUGE rispetto al Seq2SeqLSTM, indicando una maggiore capacità di catturare similarità lessicali.



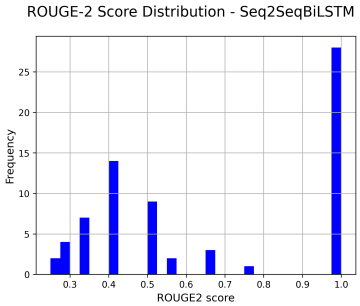
(a) ROUGE-1 Seq2SeqLSTM



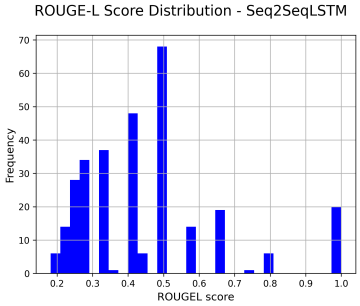
(b) ROUGE-1 Seq2SeqBiLSTM



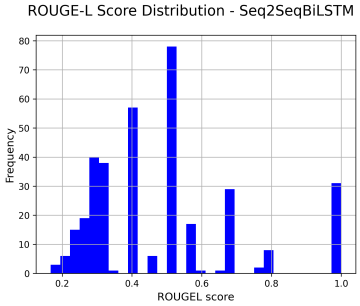
(c) ROUGE-2 Seq2SeqLSTM



(d) ROUGE-2 Seq2SeqBiLSTM



(e) ROUGE-L Seq2SeqLSTM



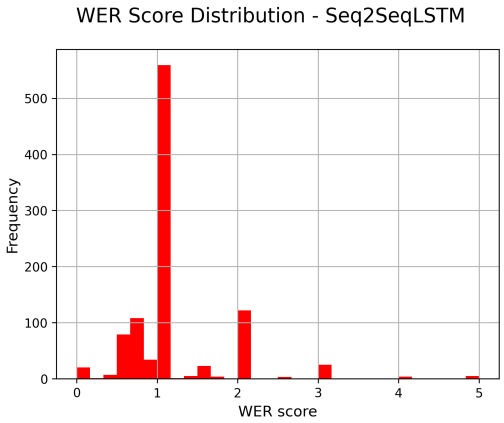
(f) ROUGE-L Seq2SeqBiLSTM

Figura 12: Confronto dei punteggi ROUGE tra i modelli Seq2SeqLSTM e Seq2SeqBiLSTM.

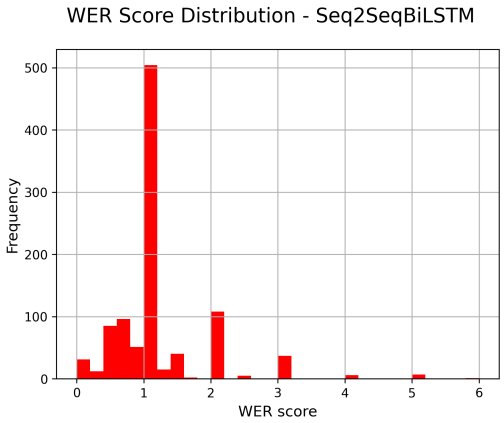
10.2 Word Error Rate (WER)

Il confronto del WER, mostrato nella Figura 13, evidenzia che il modello Seq2SeqBiLSTM ottiene risultati migliori, indicando una maggiore accuratezza nella generazione delle

parole.



(a) WER Seq2SeqLSTM

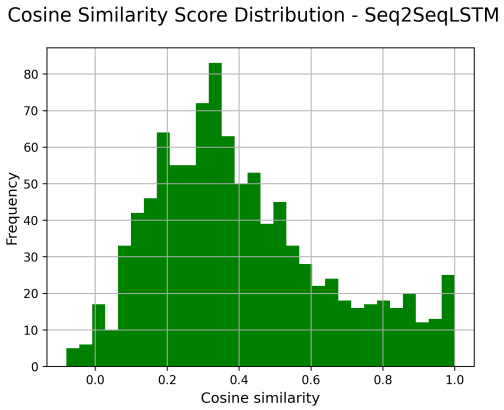


(b) WER Seq2SeqBiLSTM

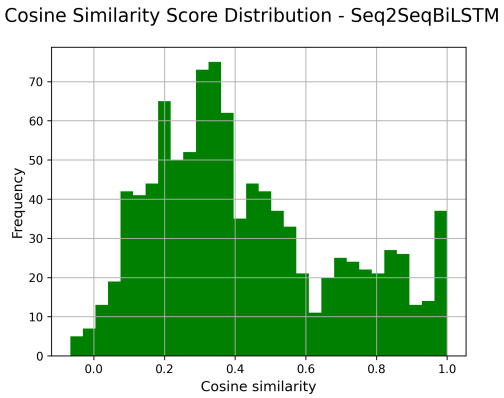
Figura 13: Confronto del Word Error Rate tra i modelli Seq2SeqLSTM e Seq2SeqBiLSTM.

10.3 Cosine Similarity

La Figura 14 confronta i valori di similarità cosenica. Anche in questo caso, il modello Seq2SeqBiLSTM ottiene valori più alti, suggerendo una maggiore correlazione semantica con i riassunti di riferimento.



(a) Cosine Similarity Seq2SeqLSTM



(b) Cosine Similarity Seq2SeqBiLSTM

Figura 14: Confronto della cosine similarity tra i modelli Seq2SeqLSTM e Seq2SeqBiLSTM.

11 Conclusioni

Il modello implementato dimostra la capacità di generare riassunti efficaci delle recensioni di prodotti.
I risultati complessivi indicano che il modello Seq2Seq LSTM ha difficoltà significative nel generare output precisi, sia dal punto di vista lessicale che sintattico.
Tuttavia, la similarità cosenica superiore a zero per la maggior parte delle righe suggerisce che il modello riesce a mantenere una correlazione semantica, seppure debole, con il testo di riferimento.

Per un fine di generazione di riassunti ci si può ritenere soddisfatti, grazie alla similarità cosenica.