Natural Language Processing: Bilateral LSTM vs. BERT Transformer nel PoS Tagging

Manuel Placella 1099701
Giuseppe Pio Salcuni 1100090

Laurea Magistrale in Informatica

2023-2024

1 Introduzione

Il Part-of-Speech Tagging, comunemente abbreviato come PoS Tagging, rappresenta un elemento fondamentale nel campo del Natural Language Processing (NLP), consentendo di assegnare a ciascuna parola di un testo una specifica categoria grammaticale. Questo processo, cruciale nell'analisi del linguaggio naturale, attribuisce etichette come nomi, verbi, aggettivi, avverbi e così via, ai fini dell'analisi sintattica e semantica.
L'importanza del PoS Tagging risiede nella sua capacità di fornire contesto e struttura al testo, permettendo alle macchine di comprendere meglio il significato e la relazione tra le parole all'interno di una frase o di un documento. Questo processo non solo facilita la comprensione delle regole grammaticali di una lingua, ma è fondamentale anche per applicazioni più avanzate nell'ambito del NLP, come l'analisi sentimentale, la traduzione automatica e la generazione di testi.

Le tecniche utilizzate per il POS Tagging sono varie e includono approcci basati su regole, modelli statistici e più recentemente modelli di deep learning come ad esempio le reti neurali ricorrenti (come la Long Short-Term Memory networks - LSTM) o i modelli basati su transformers (come BERT). Questi due ultimi modelli hanno senza dubbio rivoluzionato l'efficacia e la precisione di questa attività.
In questo lavoro vengono esaminate e confrontate due architetture: la Bilateral LSTM e il fine-tuning di un modello basato su BERT Transformer. Attraverso un'analisi comparativa dei risultati e degli errori commessi da queste due architetture, miriamo a valutare le prestazioni e le peculiarità di ciascuna nel contesto dell'assegnazione delle etichette grammaticali, contribuendo così alla comprensione e all'avanzamento di questa importante area nell'ambito del NLP.

2 Dataset e Pre-Processing

Abbiamo scelto di utilizzare i dati forniti da UD -Universal Dependencies [1]. I dati Universal Dependencies (UD) sono un insieme di alberi di dipendenza annotati manualmente, utilizzati principalmente per l'analisi linguistica, inclusa l'etichettatura grammaticale del discorso (POS tagging), l'analisi delle dipendenze e altro ancora. L'obiettivo principale di UD è quello di fornire un framework comune e condiviso per l'annotazione sintattica delle lingue naturali.
Per l'etichettatura grammaticale del discorso (POS tagging), i dati UD forniscono un insieme di etichette standardizzate che rappresentano le parti del discorso come nomi, verbi, aggettivi, avverbi, preposizioni, coniugazioni e altro ancora. Queste etichette consentono di identificare e categorizzare le parole all'interno di una frase in base alla loro funzione grammaticale.

Nel corso del nostro studio, abbiamo impiegato tre file distinti caratterizzati dall'estensione .txt: train.txt, valid.txt e test.txt. Ognuno di questi file è composto rispettivamente da 13121 esempi per il set di training, 564 per la fase di validazione e 482 per il set di test.

La nostra metodologia ha implicato un processo dettagliato di preparazione dei dati per renderli idonei alle specifiche richieste di entrambi i modelli. Queste operazioni di pre-elaborazione hanno coinvolto passaggi di tokenizzazione, conversione in rappresentazioni numeriche e altre trasformazioni mirate, atte a garantire che i dati fossero in grado di soddisfare le esigenze peculiari dei modelli di bilateral LSTM e BERT.

2.1 Pre-processing per Bilateral LSTM

Per creare un set di dati adatto all'addestramento della nostra rete neurale abbiamo dei campi, determinando in modo chiaro e preciso come gestire le parole e le loro categorie grammaticali all'interno dei nostri dati. All'interno del nostro documento, ogni coppia di informazioni rappresenta una parola accompagnata dalla sua categoria grammaticale associata. La separazione delle frasi mediante una riga vuota ha consentito una chiara distinzione tra i diversi esempi.
Abbiamo quindi generato tre distinti insiemi di dati destinati ai processi di addestramento, validazione e test della nostra rete. Ogni campione di questi set è costituito da due campi principali: il campo "text", che descrive il testo in forma di frase, e il campo "udtags", che fornisce i tag specifici associati a ciascuna parola all'interno del testo.
Abbiamo quindi proceduto creando vocabolari, molto importanti per il processo di elaborazione del testo.

2.2 Pre-processing per BERT

Per preparare i dati per il modello BERT, abbiamo utilizzato il tokenizzatore specifico del modello stesso, un componente cruciale che determina come il testo viene elaborato all'interno del modello. Questo tokenizzatore non solo stabilisce le regole per il trattamento del testo, ma include anche il vocabolario con cui il modello è stato addestrato. Utilizzando questo tokenizzatore, convertiamo i token speciali e otteniamo la lunghezza

massima degli esempi che il modello può elaborare.

Così come fatto in precedenza, definiamo i campi che indirizzeranno il preprocessamento del testo e creiamo i tre set di dati fondamentali che saranno successivamente impiegati per affinare e ottimizzare il modello attraverso il fine-tuning.

Da una semplice analisi del testo del set di addestramento, emerge che i 10 token più frequenti sono quelli elencati nella Tabella 1. Inoltre, le percentuali relative alle diverse etichette grammaticali presenti nei dati sono dettagliate nella Tabella 5.

Token	**Frequenza**
di	17628
,	11859
il	11240
.	9945
la	8968
a	6729
in	6059
e	5187
i	4527
l’	4347

Table 1: Token e relative frequenze

Tag	**Frequenza**	**Percentuale**
NOUN	54983	19.9%
DET	45034	16.3%
ADP	41890	15.2%
PUNCT	31294	11.3%
VERB	23364	8.5%
ADJ	18322	6.6%
PROPN	13671	5.0%
AUX	10878	3.9%
ADV	10536	3.8%
PRON	10428	3.8%
CCONJ	7550	2.7%
NUM	4791	1.7%
SCONJ	2826	1.0%
X	272	0.1%
SYM	93	0.0%
INTJ	62	0.0%
PART	24	0.0%

Table 2: Frequenza dei Tag e Relative Percentuali

3 Architetture

3.1 Bilateral LSTM

La Long Short-Term Memory (LSTM) bidirezionale, conosciuta come BiLSTM, rappresenta un'evoluzione della classica rete neurale ricorrente (RNN) progettata per l'elaborazione di sequenze di dati. È un tipo di architettura che integra due LSTM separate, consentendo al modello di acquisire informazioni contestuali sia dal passato che dal futuro di ciascun elemento nella sequenza.
Come le LSTM tradizionali, la BiLSTM è dotata di meccanismi per mantenere e aggiornare informazioni rilevanti nel tempo.
Nella Figura 3 è rappresentata una versione semplificata del modello con un unico strato LSTM. Il modello riceve una sequenza di token, li passa attraverso uno strato di em-

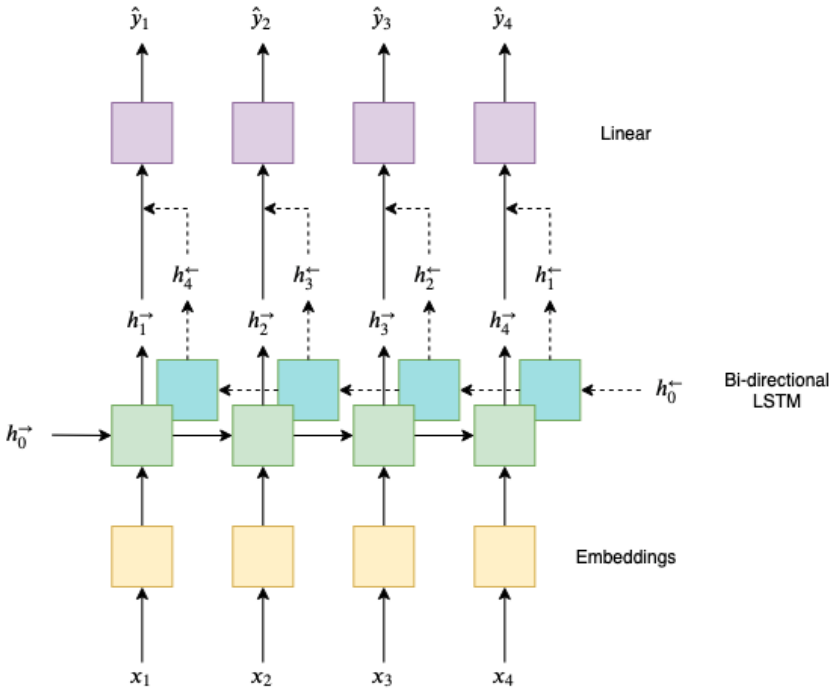

Figure 1: Struttura di una BiLSTM

bedding per ottenere i "token embeddings" (operazione che consiste nell'assegnare un vettore numerico ad ogni token consentendo così al modello di comprendere e manipolare il significato del testo). Questi embeddings vengono quindi elaborati - uno per ogni passaggio temporale - dai LSTM sia in avanti che all'indietro. L'LSTM in avanti elabora la sequenza da sinistra a destra, mentre l'LSTM all'indietro lo fa da destra a sinistra.
Ogni LSTM riceve non solo l'input corrente (cioè il token corrente nella sequenza), ma anche gli stati nascosti e di cella dal passaggio temporale precedente. Questi stati rappresentano le informazioni memorizzate e elaborate dal modello durante il passaggio precedente attraverso la sequenza. Gli LSTM elaborano l'input corrente utilizzando le informazioni ricevute dall'input corrente e dagli stati nascosti/cella precedenti. Dopo l'elaborazione del token corrente, gli LSTM generano nuovi stati nascosti e di cella, che vengono passati al passaggio successivo nella sequenza.

Alla fine, gli stati nascosti risultanti dalle direzioni in avanti e all'indietro vengono concatenati e passati attraverso uno strato lineare per prevedere il tag appropriato per ciascun token.
Durante l'addestramento del modello, le previsioni dei tag vengono confrontate con i tag effettivi associati a ciascun token nella sequenza. Questo confronto consente al modello di calcolare un'accuratezza o una perdita (loss) associata alle previsioni. Il modello viene quindi ottimizzato regolando i parametri per minimizzare questa perdita, migliorando progressivamente la precisione delle previsioni dei tag durante il processo di addestramento.

Sebbene la BiLSTM sia potente nel catturare relazioni contestuali nelle sequenze, richiede più risorse computazionali rispetto alle RNN unidirezionali a causa della sua struttura bidirezionale, portando a una maggiore complessità computazionale e a un maggiore numero di parametri da addestrare. Tuttavia, spesso offre prestazioni migliori nella comprensione del contesto nelle sequenze di dati.
Il modello descritto sopra è implementato nella classe `BiLSTMPoSTagger`.

3.2 BERT Transformer

BERT, acronimo di "Bidirectional Encoder Representations from Transformers", è un potente modello di elaborazione del linguaggio naturale sviluppato da Google. Si distingue per la sua capacità di comprensione contestuale dei testi, addestrato su vasti corpus di testi mediante l'utilizzo di transformer neurali, una particolare architettura di reti neurali. La sua caratteristica principale è la capacità di comprendere il contesto dei testi analizzando l'intera sequenza di parole o token, anziché trattarli in modo isolato. La sua architettura bidirezionale consente di considerare sia le parole precedenti che quelle successive in una frase, permettendo una rappresentazione più ricca e contestualizzata delle parole all'interno del testo.

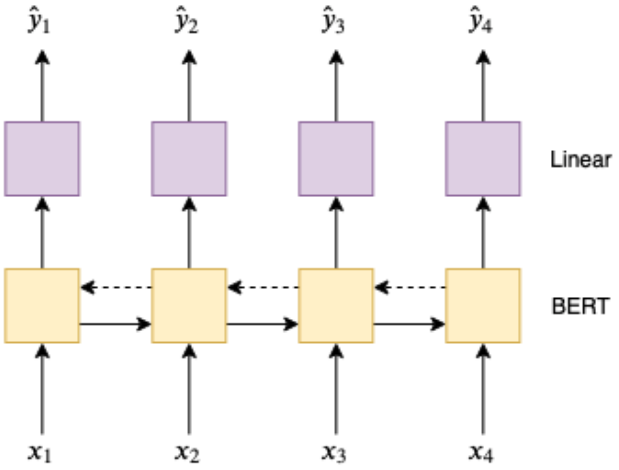

Figure 2: Struttura BERT

Grazie al suo addestramento su grandi quantità di dati testuali provenienti da internet,

BERT ha dimostrato un'elevata capacità di comprensione e generazione del linguaggio, superando molte sfide nei compiti di elaborazione del linguaggio naturale, come il riconoscimento dell'entità nominata, la classificazione del testo e il riassunto automatico. Dopo il preaddestramento iniziale, il modello BERT si presta a essere adattato per svolgere con successo una vasta gamma di compiti nell'ambito dell'elaborazione del linguaggio naturale (NLP).
Nel nostro caso specifico, abbiamo personalizzato BERT tramite un processo di fine-tuning per eseguire il Part-of-Speech (PoS) Tagging.
Nella configurazione del modello BERT da noi scelto, abbiamo aggiunto un livello lineare dedicato a prevedere il tag associato a ciascun token all'interno della sequenza di input.
Nello specifico, abbiamo optato per l'utilizzo del modello BERT
`dbmdz/bert-base-italian-xxl-uncased` [2], modello fornito dalla piattaforma HuggingFace e appositamente addestrato sul linguaggio italiano. Il corpus di addestramento per il modello italiano BERT è stato costituito da un recente dump di Wikipedia, insieme a una vasta selezione di testi provenienti dalla raccolta di corpora OPUS ampliata dalla sezione italiana del corpus OSCAR. Il corpus di addestramento è di circa 81 GB, contenente un totale di 13.138.379.147 token. Questa vasta mole di dati ha fornito al modello una solida base di conoscenze linguistiche, contribuendo a una comprensione approfondita e accurata del linguaggio italiano.

La decisione di adottare questo particolare modello è stata guidata dalla vasta quantità di dati impiegati nel suo processo di addestramento, che supera nettamente quella utilizzata per altri modelli disponibili.
Per sviluppare e implementare le due architetture e condurre l'addestramento e la valutazione dei modelli, ci siamo ispirati a un lavoro preesistente [3] che coinvolgeva la costruzione di una rete bi-LSTM e l'impiego di un modello basato su BERT. Tuttavia, è stato essenziale apportare modifiche e adattamenti significativi a queste architetture preesistenti per garantire la loro idoneità e adattamento al contesto specifico del nostro caso di studio.

4 Metodologia

Addestramento

Per condurre un confronto accurato tra le due architetture, entrambi i modelli sono stati addestrati utilizzando lo stesso set di dati di training, impiegando la stessa percentuale di dropout (il quale consiste nella disattivazione casuale di un insieme di unità durante l'addestramento) per prevenire l'overfitting. Ciascun modello è stato sottoposto a 10 epoche, dove un'epoca rappresenta un passaggio completo dell'intero set di dati attraverso il modello durante l'addestramento.
È importante sottolineare che in ciascuna epoca di addestramento, i modelli sono stati valutati utilizzando il validation set. Questa fase di valutazione sul set di validazione è cruciale poiché è quella con cui vengono regolati i pesi del modello. Durante questo processo, i pesi vengono aggiornati in base al confronto tra le previsioni del modello e le

etichette effettive del validation set, contribuendo a ottimizzare e affinare ulteriormente le prestazioni del modello.
Sia nel caso della Bilateral LSTM che nel modello BERT abbiamo utilizzato l'Adam optimizer. L'Adam optimizer è un algoritmo di ottimizzazione molto utilizzato per regolare i pesi del modello durante l'addestramento, aggiornandoli in base al gradiente calcolato rispetto alla funzione di perdita. Questo processo consente al modello di apprendere e migliorare durante l'addestramento, spostandosi verso una migliore capacità predittiva rispetto ai dati forniti.
Abbiamo adottato la funzione di perdita CrossEntropy come metrica per valutare le discrepanze tra le previsioni del nostro modello e i valori effettivi.
Abbiamo poi implementato la tecnica del dropout nel nostro modello, utilizzando un parametro specifico di 0.25. Il dropout è una strategia di regolarizzazione che mira a ridurre l'overfitting durante l'addestramento della rete neurale. L'uso di un valore del dropout pari a 0.25 implica che, durante ciascuna iterazione di addestramento, il 25% delle unità o dei nodi nella rete sono temporaneamente esclusi o "droppati", in modo casuale, per prevenire l'eccessiva dipendenza tra di essi.
Abbiamo selezionato gli iperparametri per entrambi i modelli basandoci su studi e lavori pregressi.

Metriche di valutazione

Per entrambi i modelli, abbiamo calcolato vari indicatori di performance come l'accuratezza, la precisione e la recall al fine di valutare dettagliatamente le loro prestazioni:

- **accuratezza**: rappresenta la percentuale di predizioni corrette rispetto al totale delle predizioni effettuate;
- **precisione**: misura la percentuale di predizioni corrette tra quelle classificate come positive dal modello;
- **recall**: indica la percentuale di istanze positive correttamente identificate dal modello rispetto al totale delle istanze positive presenti nei dati.
- **F1-measure**: rappresenta la media armonica tra precisione e richiamo. Viene utilizzato per fornire una valutazione complessiva delle prestazioni del modello, considerando sia la capacità di prevedere correttamente le istanze positive che la capacità di identificare correttamente tutte le istanze positive.

Questi valori sono fondamentali per condurre un'analisi approfondita degli errori commessi dai modelli, come sarà discusso nel capitolo successivo.
Durante la fase di valutazione sul set di test, abbiamo costruito anche la matrice di confusione per ciascun modello. Questa matrice fornisce una rappresentazione visiva degli errori commessi, mostrando in modo dettagliato come le previsioni del modello corrispondono o differiscono dalle etichette effettive nei dati di test. Questo strumento consente di individuare specificamente quali classi o categorie hanno generato più confusione per ciascun modello.
Al completamento dell'addestramento, abbiamo implementato un breve metodo che riceve come input sia il modello che una frase espressa come stringa di testo. Questo metodo

svolge la funzione di eseguire il PoS Tagging sulla frase fornita, applicando le regole di etichettatura delle parti del discorso tramite il modello addestrato.

5 Risultati e Analisi

Questo capitolo espone i risultati derivanti dalla valutazione dei due modelli precedentemente delineati utilizzando il set di dati di test. Vogliamo presentare quindi un'analisi dettagliata delle metriche rilevanti e delle osservazioni emerse durante questo processo di valutazione.

- **Accuratezza del modello** Come spiegato in precedenza questa metrica indica la percentuale di predizioni corrette effettuate dal modello rispetto al totale delle previsioni fatte. Da questa prima semplice metrica si può notare come il modello

BiLSTM	BERT
95.46%	97.52%

  BERT ha mostrato una perdita inferiore e un'accuratezza superiore rispetto al modello BiLSTM durante la fase di test (cioè su dati che il modello non ha mai visto). Questo suggerisce che il BERT ha una migliore capacità predittiva e una maggiore precisione nelle sue previsioni rispetto al BiLSTM su questo specifico set di dati.

- **Analisi delle percentuali di errore per classe**
  Iniziamo l'analisi degli errori commessi dai modelli esaminando la percentuale di errori per ciascuna classe.

  Modello BiLSTM

Classe	Percentuale di Errore
X	100.0%
ADJ	23.38%
PROPN	17.43%
SCONJ	13.00%
NUM	10.53%
NOUN	6.76%
PRON	6.33%
VERB	6.03%
ADV	5.49%
AUX	1.48%
DET	0.88%
ADP	0.79%
PUNCT	0.09%
CCONJ	0.0%
SYM	0.0%

Table 3: Percentuale di errore per ciascuna classe

Come riportato nella Tabella 3, la classe X (categoria grammaticale "altro") ha una percentuale di errore del 100.0%. Questa classe viene etichettata erroneamente in tutte le previsioni del modello. ADJ e PROPN presentano una percentuale di errore più elevata, pari rispettivamente al 23.38% e al 17.43%. Questo significa che circa il 23.38% delle previsioni etichettate come ADJ e il 17.43% delle previsioni etichettate come PROPN sono errate. SCONJ, NUM, NOUN, PRON, VERB, ADV: Mostrano percentuali di errore che vanno dal 13.00% al 5.49%. Queste classi presentano una percentuale moderata di errori nelle previsioni del modello. AUX, DET, ADP, PUNCT, CCONJ, SYM: Hanno una percentuale di errore più bassa, variando dal 1.48% al 0.0%. Queste classi sono soggette a un minor numero di errori di classificazione, indicando una migliore performance del modello nel riconoscimento di queste categorie.

Modello BERT

Classe	Percentuale di Errore
X	76.93%
PROPN	19.80%
ADV	7.23%
ADJ	6.03%
NUM	5.85%
VERB	4.17%
NOUN	3.72%
AUX	3.21%
SCONJ	3.00%
PRON	1.22%
ADP	0.79%
CCONJ	0.38%
DET	0.29%
PUNCT	0.09%
SYM	0.0%

Table 4: Percentuale di errore per ciascuna classe

In confronto al modello BiLSTM, i risultati ottenuti dal modello BERT sono notevolmente più soddisfacenti. Come si può vedere nella Tabella 4 anche in questo caso il modello commette errori sulla classe X, sebbene questa classe non abbia rilevanza pratica poiché rappresenta un'entità indefinita e non significativa in termini grammaticali. Tuttavia si può subito notare come la classe PROPN viene anche in questo caso classificata in modo errata nel 19.8% dei casi. Incrociando con il risultato del modello precedente si può dire che su questa classe in generale vengono commessi troppi errori.
Si osserva poi un significativo miglioramento nella precisione delle classificazioni successive: la classe ADV presenta un tasso di errore dell'7.2% (rispetto al 5.49% del modello precedente), mentre la classe ADJ mostra un tasso di errore del 6.03%, in netto ribasso rispetto alla sua precedente posizione come la classe con la per-

centuale di errore più alta nel modello BiLSTM.

Sin da questa prima analisi, è evidente che, ad eccezione della classe PROPN, i due modelli commettono errori su categorie di parole nettamente distinte. Questo aspetto conferma il diverso funzionamento dei due modelli.

- **Analisi della matrice di confusione**
  Di seguito vengono analizzate le matrici di confusione nel dettaglio.

Modello BiLSTM

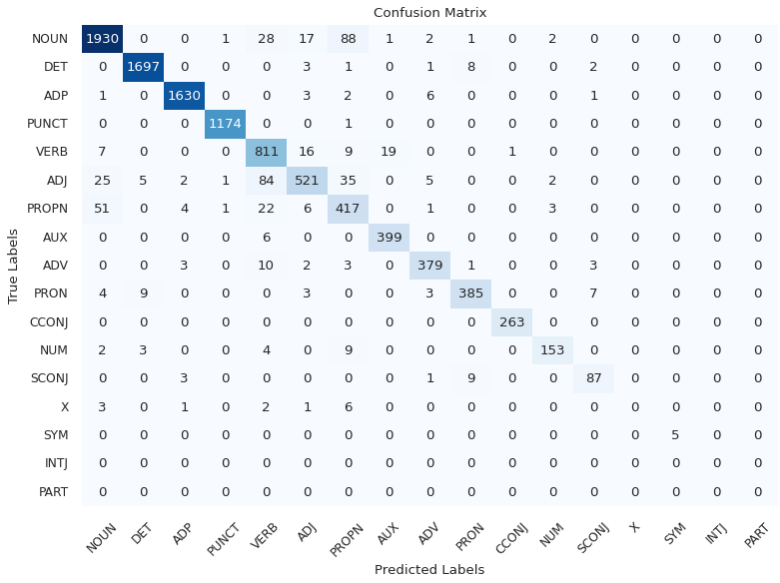

Figure 3: Matrice di confusione BiLSTM

Tag	Precision	Recall	F1-measure
NOUN	95.40%	93.24%	94.31%
DET	99.01%	99.12%	99.07%
ADP	99.21%	99.21%	99.21%
PUNCT	99.75%	99.91%	99.83%
VERB	83.87%	93.97%	88.63%
ADJ	91.08%	76.62%	83.23%
PROPN	73.03%	82.57%	77.51%
AUX	95.23%	98.52%	96.84%
ADV	95.23%	94.51%	94.87%
PRON	95.30%	93.67%	94.48%
CCONJ	99.62%	100.00%	99.81%
NUM	95.62%	89.47%	92.45%
SCONJ	87.00%	87.00%	87.00%
X	0.00%	0.00%	0.00%
SYM	100.00%	100.00%	100.00%
INTJ	0.00%	0.00%	0.00%
PART	0.00%	0.00%	0.00%

Table 5: Precisione, Recall e F1-Score per BiLSTM

Guardando ai risultati ottenuti (Tabella 5), nel dettaglio la metrica **precisione**, emerge chiaramente come il modello classifichi quasi alla perfezione le classi DET, ADP, PUNCT, CCONJ e SYM. Ciò significa che se il modello le classifica in queste classi è quasi certa la loro correttezza. Per queste classi quindi, il modello mostra una capacità quasi impeccabile nel fare predizioni corrette.

Si nota una leggera diminuzione della precisione e quindi un po' più di incertezza nella classificazione della classe NOUN. È interessante notare che molti elementi di questa classe vengono erroneamente classificati come PROPN, che rappresenta una categoria molto simile. Questi errori sulla classe NOUN sono in parte attribuibili alla loro numerosa presenza all'interno del dataset, rendendo la distinzione più complessa.
Per quanto riguarda la classe VERB, questa presenta una precisione ridotta, in quanto spesso vengono erroneamente assegnati elementi appartenenti principalmente alle classi ADJ e NOUN, segnalando un errore significativo nel modello.

Le restanti classi mantengono una precisione compresa tra il 90 e il 95%, un risultato comunque abbastanza importante.

Dalla metrica **recall** emerge invece come il modello sia in grado di catturare praticamente tutti gli elementi delle classi DET, ADP, PUNCT, AUX e CCONJ, oltre a SYM, sebbene ci siano solo 5 esempi di questa classe nel dataset. Tuttavia, si notano maggiori difficoltà nel modello nel catturare tutti gli elementi appartenenti alle classi ADJ, PROPN, NUM e SCONJ.

Nelle classi ADJ, PROPN, NUM e SCONJ, il modello sembra avere più difficoltà a identificare tutti gli esempi reali, suggerendo una certa inefficienza nel riconosci-

mento di queste categorie specifiche.

Modello BERT

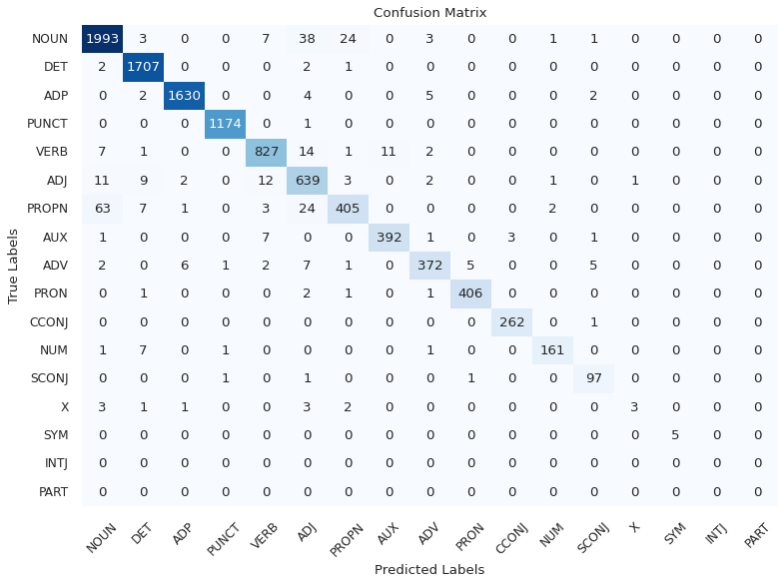

Figure 4: Matrice di confusione BERT

Tag	Precision	Recall	F1-measure
NOUN	95.68%	96.28%	95.98%
DET	98.22%	99.71%	98.96%
ADP	99.39%	99.21%	99.30%
PUNCT	99.75%	99.91%	99.83%
VERB	96.39%	95.83%	96.11%
ADJ	86.94%	93.97%	90.32%
PROPN	92.47%	80.20%	85.90%
AUX	97.27%	96.79%	97.03%
ADV	96.12%	92.77%	94.42%
PRON	98.54%	98.78%	98.66%
CCONJ	98.87%	99.62%	99.24%
NUM	97.58%	94.15%	95.83%
SCONJ	90.65%	97.00%	93.72%
X	75.00%	23.08%	35.29%
SYM	100.00%	100.00%	100.00%
INTJ	0.00%	0.00%	0.00%
PART	0.00%	0.00%	0.00%

Table 6: Precisione, Recall e F1-Score per BERT

Esaminando attentamente i risultati di precisione nella Tabella 6, emerge chiaramente che il modello dimostra una **precisione** praticamente impeccabile nelle classi

DET, ADP, PUNCT, PRON, CCONJ e SYM.
Si rileva una minore precisione nelle classi ADJ, dove spesso si confonde con le classi NOUN e PROPN. PROPN, come nel caso precedente, viene confusa con NOUN (ma come detto in precedenza queste rappresentano categorie molto simili).
Tra le categorie grammaticali di fondamentale importanza, spicca sicuramente la classe VERB. Nel modello BERT, si evidenziano significativi miglioramenti, specialmente in termini di precisione e recall di questa categoria. La precisione, attestata al 96.4%, conferma chiaramente che questo modello supera il suo predecessore in questa categoria di notevole importanza. In tutte le altre categorie, la precisione si attesta comunque oltre il 95%, un risultato estremamente significativo e soddisfacente.

Per quanto riguarda il **recall**, emerge una notevole difficoltà del modello nel catturare tutti gli elementi della classe PROPN, con un recall pari all'80%. Al contrario, il modello riesce a catturare praticamente tutti gli elementi delle classi DET, ADP, PUNCT, PRON, CCONJ, SCONJ e SYM. Per tutte le altre classi, riesce a identificare più del 90% degli elementi, un risultato tuttavia soddisfacente.

Dalla matrice di confusione, è evidente che gli errori commessi nella classificazione della classe PROPN sono in gran parte dovuti all'errata assegnazione di questi elementi alla classe NOUN.

- **Confronto finale**

Dall'analisi delle percentuali di errore possiamo concludere che entrambi i modelli incontrano difficoltà nella corretta classificazione della categoria PROPN (nome proprio). Sono presenti tuttavia differenze significative tra i due modelli in altri ambiti: il modello BiLSTM mostra una tendenza a commettere numerosi errori nelle classi ADJ (aggettivo) e SCONJ (congiunzione subordinata), mentre il modello BERT presenta delle difficoltà nella corretta identificazione degli elementi appartenenti alla classe ADV (avverbio), assegnandoli in modo più preciso nel modello BiLSTM.

È interessante notare che il modello BiLSTM sembra essere più efficace nella classificazione della classe AUX (ausiliare) rispetto al modello BERT, nonostante la sua performance generale inferiore. Ciò si evidenzia nel tasso di errore dell'1.48% del modello BiLSTM contro il 3.21% del modello BERT per questa specifica classe.

In entrambi i modelli c'è una netta difficoltà nella corretta categorizzazione degli elementi di tipo NUM.

Dall'esame della matrice di confusione e dei valori di F1-score, che indica l'equilibrio tra precisione e recall, emerge che entrambi i modelli presentano difficoltà relative (in base alle prestazioni complessive) nella corretta classificazione degli elementi appartenenti alle classi PROPN (nome proprio), ADJ (aggettivo) e SCONJ (congiunzione subordinata). Il modello BERT mostra una maggiore efficacia nella

classificazione delle classi VERB (verbo) e PRON (pronome), mentre il modello BiLSTM sembra andare meglio nella categorizzazione degli avverbi (ADV).

È evidente in entrambi i casi una tendenza a incontrare difficoltà nella distinzione dei nomi propri (PROPN), spesso confusi con sostantivi (NOUN) o aggettivi (ADJ). Analogamente, la classe NOUN mostra una tendenza simile alla PROPN nel essere confusa con altre categorie, sebbene in proporzione commetta meno errori rispetto alla PROPN, probabilmente a causa della sua presenza più consistente all'interno del set di dati.

- **Esempio di tagging**

  Abbiamo applicato una funzione dedicata per eseguire l'analisi del Part of Speech (POS) su una frase passata in input. Lo abbiamo fatto utilizzando entrambi i modelli sviluppati. Riportiamo di seguito i risultati ottenuti da entrambi i modelli, mettendo a confronto le categorie grammaticali assegnate alle parole all'interno della frase.

Token	Pred. Tag BiLSTM	Pred. Tag BERT
bologna	PROPN	PROPN
è	AUX	AUX
una	DET	DET
città	NOUN	NOUN
universitaria	ADJ	ADJ
molto	ADV	ADV
affascinante	ADJ	ADJ
.	PUNCT	PUNCT
è	AUX	AUX
famosa	ADJ	ADJ
per	ADP	ADP
i	DET	DET
suoi	DET	DET
portici	NOUN	NOUN
e	CCONJ	CCONJ
per	ADP	ADP
la	DET	DET
sua	DET	DET
cucina	NOUN	NOUN
deliziosa	VERB	ADJ
.	PUNCT	PUNCT
le	DET	DET
torri	NOUN	NOUN
rappresentano	VERB	VERB
un	DET	DET
punto	NOUN	NOUN
centrale	ADJ	ADJ
e	CCONJ	CCONJ
storico	ADJ	ADJ
della	VERB	ADP
città	NOUN	NOUN
.	PUNCT	PUNCT

Table 7: POS Tagging della frase fornita

La presenza di pochi errori nei modelli è evidente: il PoS-Tagging viene eseguito abbastanza accuratamente da entrambi i modelli. Tuttavia, bisogna sottolineare che il BiLSTM commette due errori significativi su "deliziosa" e "della", classificandoli erroneamente come VERB.

6 Conclusioni

In conclusione di questo studio, emerge chiaramente che il modello BERT, come previsto, mostra una maggiore performance rispetto al modello Bilateral LSTM. Tuttavia, la diversità nelle architetture dei due modelli si riflette negli errori commessi, che risultano notevolmente differenziati in alcune classi.
L'analisi delle parti che influenzano maggiormente le scelte effettuate dal modello potrebbe essere un ambito di ricerca molto interessante. Tecniche come SHAP (SHapley Additive exPlanations) o Lime (Local Interpretable Model-agnostic Explanations) potrebbero essere impiegate per esplorare in profondità gli attributi che maggiormente influenzano le predizioni del modello.

Notes

[1] `https://universaldependencies.org`
[2] `https://huggingface.co/dbmdz/bert-base-italian-cased`
[3] `https://github.com/bentrevett/pytorch-pos-tagging/blob/master/2_transformer.ipynb`