Università degli studi di Padova
Dipartimento di Matematica "Tullio Levi-Civita"

Corso di laurea magistrale in Informatica

Integrazione di regole linguistiche in modelli NLP

Candidato
Marco Cuccarini

Relatore
Prof. Nicolò Navarin

Anno accademico 2022-2023

A Matilde,
grazie a lei sono quello che sono.
E anche per lei sarò quello che sarò.

Indice

1 Introduzione **5**

2 Basi teoriche **7**
2.1 Machine learning 7
2.1.1 Il task T 8
2.1.2 Le performance P 8
2.1.3 L'esperienza E 8
2.1.4 Selezione del modello 8
2.1.5 Reti neurali e artificiali 10
2.1.6 Metriche di valutazione 23
2.2 Scienza dialogica 25
2.2.1 MADIT 27

3 Lavoro pregresso e primi test **28**
3.0.1 Divisione testo 28
3.0.2 Classificazione dei repertori 29
3.0.3 Classificazione Evalita 30
3.1 Lavoro pregresso 33
3.1.1 Primo lavoro esplorativo 33
3.1.2 Repertori per EVALITA2016 34
3.2 Proseguimento lavoro 36
3.3 Support Vector Machine per la classificazione del testo codificato prodotto dal modello BERT 37

4 Integrazione informazione repertori discorsivi nel modello BERT **39**
4.1 Stato dell'arte: Alberto [15] 39
4.1.1 Fine tuning 39
4.2 Alberto e transfer learning della classificazione dei repertori 40
4.2.1 Risultati 42
4.2.2 Prime conclusioni 51
4.3 Esplorazione di altri metodi di integrazione 51
4.3.1 Risultati 53

5 Classificazione dei repertori per hate speech e stereotipi **60**
5.1 Distribuzione repertori e albero di decisione 60
5.2 Modello Alberto e repertori 64
5.2.1 Risultati 65

5.2.2 Conclusioni 68

6 Analisi delle predizioni e metodi ensemble **69**
6.1 Ironia 69
6.2 Soggettività 70
6.3 Hate speech 71
6.4 Stereotipi 72
6.5 Modello ensemble 72
6.6 Conclusioni 74
6.7 Generalizzazione 76
6.7.1 Risultati 76

7 Conclusioni **78**

Capitolo 1

Introduzione

Se si cerca la definizione di "linguaggio" nel vocabolario Treccani [1], si può notare che viene definito come "strumento di comunicazione usato dai membri di una stessa comunità". Il linguaggio è infatti un elemento fondamentale per il funzionamento di una società: evolve con essa e ne subisce le influenze.

Tra i linguaggi utilizzati per comunicare, un ruolo importante è rivestito dal testo scritto, utilizzato come forma per fissare concetti espressi a voce. Per grande parte della storia umana la possibilità di scrivere è sempre stata riservata ad una piccola porzione della popolazione, solitamente la più abbiente. Con l'evoluzione dei mezzi di produzione si è arrivati ad industrializzare il processo di stampa di testi e si è resa molto più abbordabile la possibilità di produrre testi scritti.

L'ultima svolta è stata la digitalizzazione, che ha reso ancora più accessibile la fruizione e produzione di testo scritto e di conseguenza la quantità di testo prodotto. Prima della digitalizzazione la produzione di testo era in mano ad organizzazioni, aziende o enti che potevano vantare un certo grado di autorevolezza, e quindi avere anche un ruolo di controllo nei testi prodotti. La possibilità per tutti di produrre testi porta con se la necessità di controllare la natura dei testi stessi; sono innegabili i possibili danni che possono fare notizie false, discorsi d'odio e linguaggio offensivo non adeguatamente identificati ed eventualmente, o bloccati, o segnalati.

L'NLP, o in inglese Natural Language Processing, è la branca dell'intelligenza artificiale che si occupa dello studio, comprensione, manipolazione e riproduzione del linguaggio umano da parte di una macchina. Negli ultimi anni sono stati fatti grandi investimenti per la produzione di modelli che fossero in grado di comprendere un determinato contesto -ovvero, dato un testo, potessero classificare sentimenti o altre caratteristiche- o che potessero generare testo una volta date in input alcune sue caratteristiche (tema, stile di scrittura, etc.).

L'obiettivo di questo lavoro è studiare in che maniera le informazioni fornite da regole linguistiche (tra le quali prenderemo in considerazione le regole definite come "scienza dialogica") possono migliorare le prestazioni dei modelli che rappresentano lo stato dell'arte per alcuni task di sentiment analysis e classificazione testo in lingua italiana. Per fare ciò sono stati presi ad esempio 6 differenti task di classificazione e tramite diverse tecniche sono state incorporate informazioni fornite dalla scienza dialogica all'interno dei modelli che rappresentano lo stato dell'arte per questi problemi. Tali task riguardano l'analisi dei sentimenti (positivi o negativi) o l'analisi

delle caratteristiche di un testo, per capire se ci si trovi davanti un testo con elementi di ironia, di soggettività, di hate speech e stereotipali.

Sono stati scelti task differenti in termini di difficoltà, per poter capire in che misura l'utilizzo dei repertori aiuti nella loro risoluzione. Si tratta di dataset composti da tweet in italiano creati per delle competizioni indette da EvalIta in anni differenti:

- Dataset EvalIta2016 (ironia, soggettività, positività e negatività): in questo dataset sono presenti 7410 tweets per il set di allenamento e 2000 per quello di test, estrapolati durante il periodo della riforma "Buona Scuola" del governo Renzi.

- Dataset EvalIta2020 (hate speech e stereotipi): questo dataset è composto da 4000 esempi di train, e 496 di test postati tra Ottobre 2016 e Aprile 2017 con aggiunta del subset ottenuto grazie al progetto di monitoraggio dell'hate speech "Contro l'Odio".

- Dataset iroIta2018 (sarcasmo e ironia): verrà usata solo la parte di ironia per testare le prestazioni del modello allenato nel dataset precedente con una nuova tipologia di dati. Questo riguarda temi di hate speech e posizionamento politico. Il dataset di test che andremo poi ad utilizzare è composto da circa 800 esempi, con le due classi ben bilanciate.

I risultati mostrano come le informazioni estrapolate tramite la scienza dialogica permettono un considerevole aumento della performance predittiva dei modelli impiegati nella classificazione dei tweet ironici, soggettivi, di hate speech e stereotipali, mentre invece per la polarità non si sono visti miglioramenti significativi. Sarà possibile notare che tutti i task che in qualche modo hanno visto un miglioramento sono in qualche modo correlati al concetto di ironia.

Capitolo 2

Basi teoriche

Le basi teoriche del presente lavoro di tesi riguardano principalmente concetti legati all'intelligenza artificiale -nello specifico machine learning e deep learning- applicati al campo del processamento del linguaggio naturale. Quando si parla di processamento del linguaggio naturale o NLP, si può fare riferimento a diversi task: ci si può riferire ad una traduzione da una lingua ad un altra, alla generazione di testo dati determinati elementi o caratteristiche (argomento, modalità di scrittura, etc.) oppure alla sua comprensione, tipico dei problemi di classificazione come detenzione di fake news o di hate speech.

Questa tesi sfrutta inoltre concetti di linguistica, applicando il lavoro teorico svolto negli ultimi anni da parte del prof Turchi, docente del dipartimento FISSPA dell'Università di Padova con cui ho collaborato.

La scienza dialogica studia le regole del linguaggio per definire una correlazione tra realtà percepita dal soggetto che produce il testo e il testo stesso. In altre parole, tramite delle regole di alto livello è possibile classificare porzioni di testo, chiamati stralci, in 24 possibili classi, chiamate repertori. In base a come questi repertori compaiono nel testo, risulta ragionevole fare assunzioni riguardo la percezione del mondo da parte del soggetto.

2.1 Machine learning

Con machine learning si intende un sotto gruppo dell'AI che consente ad una macchina di apprendere e migliorare automaticamente l'esperienza attraverso dei dati; maggiore è la qualità e quantità di questi dati migliori sono le prestazioni di questi modelli.

Data questa definizione, sorge spontaneo chiedersi che cosa si intenda con il termine "apprendere" quando si parla di una macchina. Mitchell (2007): "Un programma computerizzato si dice che apprende dall'esperienza E rispetto qualche classe del Task T e performance P, se l' aumento delle sue performance nel task T, misurate da P, migliora la sua esperienza E".

A partire da questa definizione andremo a definire gli elementi teorici che compongono il presente lavoro, andando a descrivere il task T che troveremo ad affrontare, in che maniera misureremo le performance P e in che modo sarà rappresentata l'esperienza E [9].

2.1.1 Il task T

In questa tesi verrà affrontato il problema della classificazione binaria (T). La funzione di classificazione si definisce formalmente nella seguente maniera:

$$f : \mathbb{R}^n \rightarrow \{0, ..., k\} \tag{2.1}$$

dove $x \in \mathbb{R}^n$ rappresenta uno scalare o un vettore -nel nostro caso il testo che siamo intenzionati a classificare-. Invece y un valore compreso tra 0 e k, con k il numero di classi del rispettivo task. Nel nostro caso, visto che si tratta di una classificazione binaria, k assume il valore 1.

2.1.2 Le performance P

La performance P misura quanto preforma bene il modello nel task T; la scelta del modello è fortemente influenzata dalla tipologia di task. Quando parliamo di classificazione la misura più classica delle performance è l'accuratezza, una misura che rapporta il numero di predizioni corrette al numero totale di predizioni. Questo approccio però mostra dei limiti in situazioni dove c'è uno sbilanciamento nelle classi, ovvero quelle situazioni in cui sono presenti un numero estremamente diverso di esempi per ogni possibile classe. Per questo motivo sono state utilizzate metriche che tengono conto anche di questi fattori.

Le performance sono un elemento fondamentale nel sistema: la misura di performance scelta influenzerà profondamente il processo di apprendimento, andando a penalizzare certi errori piuttosto che altri.

2.1.3 L'esperienza E

L'esperienza E corrisponde al dataset, ovvero una collezione di esempi differenti. Questi esempi possono essere etichettati oppure no. Nel primo caso si definisce apprendimento non supervisionato, nell'altro apprendimento supervisionato. Nel primo caso il modello deve imparare a riprodurre una distribuzione di probabilità p(x), mentre nel secondo la distribuzione di probabilità sarà definita da p(y|x).

In alcuni casi viene utilizzata una versione ibrida delle due tecniche, chiamato apprendimento semi-supervisionato. In questo lavoro si parla di approccio supervisionato, visto che andremo ad utilizzare testi etichettati.

2.1.4 Selezione del modello

Nel caso in cui il dataset sia fissato, esso viene solitamente diviso in 3 parti: il train set, il validation set e il test set. Il train set è utilizzato per allenare il modello, il validation set serve a scegliere gli iperparametri, il dataset di test per valutare il modello con gli iperparametri migliori definiti nel validation set.

I modelli non vengono testati direttamente nel test set per evitare problemi di overfitting e perché il risultato sarebbe un modello con una stima unbias dell'errore ideale.

L'overfitting si verifica quando il modello si specializza eccessivamente sui dati che ha a disposizione, perdendo generalità. E' un problema estremamente comune

ed esistono diversi metodi per evitarlo; uno tra tutti, come già detto, l'utilizzo di un validation set. Questo spesso però non è sufficiente, quindi si devono anche adottare altri sistemi di regolarizzazione.

L'underfitting invece è il problema opposto: si verifica quando il modello è eccessivamente generale e non coglie le relazioni presenti tra i dati; questo solitamente avviene quando un modello non è stato allenato sufficientemente.

In maniera più formale possiamo dire che si ha overfitting quando l'errore del dataset di train e l'errore di generalizzazione (corrispondente all'errore del dataset di validation) divergono, con l'errore di train che tende a diminuire e l'errore di generalizzazione che dopo un certo punto inizia a crescere, come si può vedere in Figura 2.1. Questo certo punto è definito come capacità ottimale.

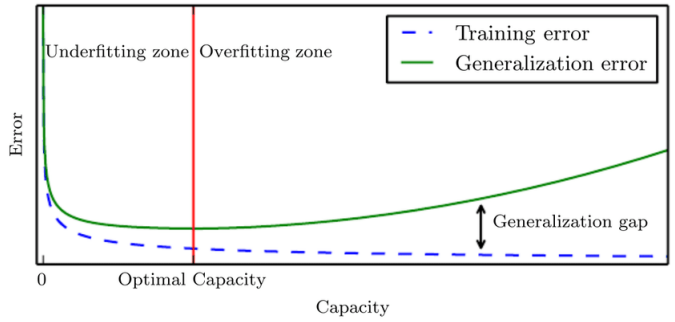

Figura 2.1: Esempio di errore di generalizzazione e di test.

I concetti di overfitting e underfitting sono strettamente legati alla capacità di un modello. La capacità di un modello è definita come il livello di semplicità nel poter dividere i dati da parte di una retta; questo valore è misurato con la dimensione Vapnik-Chervonenkis nel caso di un classificatore binario.

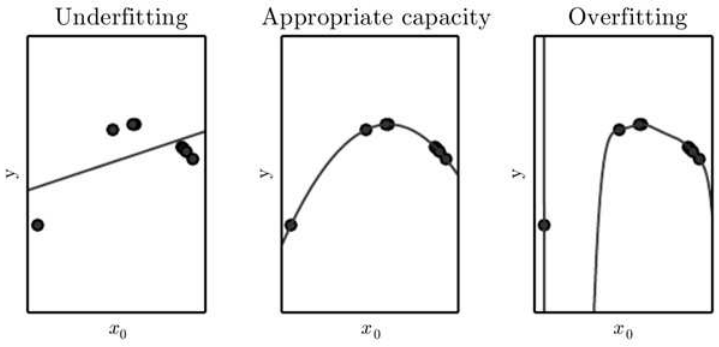

Figura 2.2: Esempio overfitting per una regressione lineare.

La dimensione VC è definita dal maggior valore di m per il quale esiste un training set di x punti diversi che il classificatore riesce a dividere arbitrariamente. Nell'esempio in Figura 2.3 si può vedere che la VC dimension è 3 ed è un'indicazione di quanto sarà facile per il modello dividere i dati.

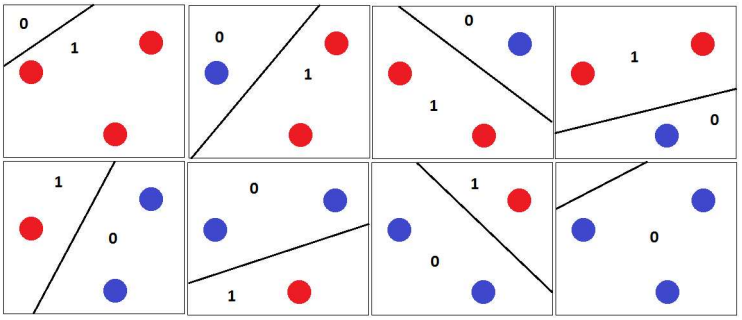

Figura 2.3: VC dimension di tre punti.

2.1.5 Reti neurali e artificiali

Neurone artificiale

La struttura basilare di una rete neurale è il neurone. Il neurone è composto da una funzione lineare e una funzione di attivazione:

- La funzione lineare moltiplica l'input x per il vettore dei pesi W. Entrambi i vettori hanno lo stesso numero di elementi.

$$a = Wx + b = \sum_{i=1}^{N} w_i \cdot x_i + b \tag{2.2}$$

- La seconda parte di un neurone è la funzione di attivazione, della quale esistono diverse tipi; la scelta della funzione di attivazione è fortemente influenzata dalla posizione che il neurone assume in una rete neurale e dalla tipologia di classificazione, binaria o non.

$$y = f(a) \tag{2.3}$$

Rete neurale

Una rete neurale è un insieme di neuroni organizzati in livelli collegati da dei pesi. I livelli possono essere divisi in input, nascosti e output, come è possibile notare in Figura 2.4. Solitamente ogni livello della rete neurale è un livello denso, che indica che ogni neurone di quel livello è collegato a tutti i neuroni dei livelli inferiori e superiori.

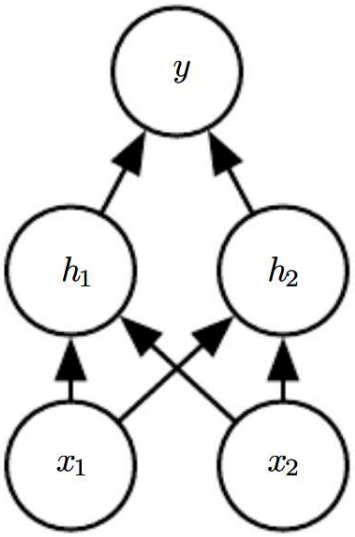

Figura 2.4: Esempio di una rete neurale.

Allenamento: fase feedforward

L'allenamento è ciò che permette alla rete di prendere in input i dati ed imparare come classificarli in maniera corretta. La prima fase dell'allenamento consiste nel calcolare la loss della rete, ovvero l'errore che la rete commette nel predire il giusto valore dati in input un insieme di vettori $X \in \mathbb{R}^n$. Quando il risultato verrà calcolato per la prima volta in fase di forward, i pesi saranno valori casuali che poi verranno corretti nella fase di backward. La scelta di inizializzazione dei pesi non è sempre casuale, e in alcuni casi si possono adottare approcci più complessi. A seconda dei livelli la fase di feedforward è formalmente definita:

- Livello nascosto $\forall i \in$ [0, ..., len(h)-1]:

$$a_i = w_{i1} \cdot x_1 + w_{i2} \cdot x_2 + b \tag{2.4}$$

$$h_i = f(a_i) \tag{2.5}$$

- Livello di output:

$$z = Vh \tag{2.6}$$

$$\hat{y} = f(z) \tag{2.7}$$

La funzione di loss che si può scegliere dipende dalla tipologia del problema. Solitamente per problemi di multi-classificazione è utilizzata la cross entropy, mentre se il problema di classificazione è binario la binary cross entropy, definita come segue:

$$BCE = -(y \log(p) + (1-y)\log(1-p)) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) \tag{2.8}$$

$$CE = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{2.9}$$

Nel contesto dei problemi di classificazione viene utilizzata la cross entropy come funzione di costo, perché minimizzarla è uguale a massimizzare la funzione di maximum likelihood estimation. La MLE è un metodo per definire quali siano i parametri che, con più probabilità, hanno generato una certa distribuzione (solitamente questa distribuzione sono i dati osservati).

Allenamento di una rete: fase backward

La fase di backward è la correzione dell'errore commesso nella fase di forward: tramite la discesa di gradiente della funzione di costo si regolano i vettori w e v per provare a raggiungere un punto di minimo locale. La discesa del gradiente è un algoritmo iterativo di primo ordine che permette di trovare un punto minimo di una funzione differenziabile. Il gradiente è un vettore composto dalle derivate parziali di una funzione scalare.

Per fare ciò è necessario applicare la derivata rispetto tutti i parametri alla funzione di costo, ovvero la cross entropy. Una volta calcolato il gradiente, aggiorno i pesi della funzione con un adeguato learning rate, che definisce il peso delle modifiche applicate ad ogni step. L'algoritmo di backpropagation calcola più volte la stessa derivata, quindi per ridurre il costo computazionale alcuni risultati possono essere salvati e calcolati un'unica volta.

$$\frac{\partial f}{\partial w_i} = \sum_i \frac{\partial f}{\partial h} \frac{\partial h}{\partial w_i} \tag{2.10}$$

$$\nabla_w f(x) = [\frac{\partial}{\partial w_1} f(x), ..., \frac{\partial}{\partial w_m} f(x)] \tag{2.11}$$

$$w' = w + \alpha \nabla_w f(x) \tag{2.12}$$

Viene calcolata la derivata della funzione costo per permettere la sua minimizzazione, percorrendo la direzione opposta del gradiente. Più la curva della funzione costo sarà ripida, maggiore sarà il valore del gradiente; meno sarà ripida, minore sarà tale valore. I valori risultanti aggiorneranno i pesi w con un parametro α detto learning rate, che controllerà il peso dell'aggiornamento del gradiente.

Struttura di una rete

La struttura di una rete influisce pesantemente sulle sue prestazioni. Solitamente si preferisce aumentare i livelli verticalmente piuttosto che orizzontalmente; questo processo dà una maggiore capacità espressiva alla rete. Per quanto riguarda i problemi lineari, per la loro risoluzione è sufficiente l'utilizzo di un singolo neurone. La scelta di organizzare i neuroni in una rete è resa necessaria perché nella maggior parte dei casi ci si trova di fronte a problemi di classificazione che non sono risolvibili in maniera lineare; in tal caso è anche necessario inserire tra due livelli consecutivi una funzione di attivazione non lineare, perché la semplice combinazione di più funzioni lineari produce sempre una funzione lineare.

Il teorema di approssimazione universale dimostra che una rete neurale con un singolo livello nascosto e con un numero sufficiente ma finito di neuroni può approssimare una qualsiasi funzione continua.

Funzioni di attivazione

Come già accennato, esistono diverse funzione di attivazione. Quando si parla di livelli nascosti si usa principalmente la ReLU. Se il valore ricevuto dalla pre-attivazione è minore di 0, il gradiente non viene aggiornato. Nel caso in cui la funzione assuma un valore maggiore di 0, invece, si comporta come una normale funzione lineare.

La funzione è formalmente definita nella seguente maniera:

$$ReLU(z) = max(0, z) \tag{2.13}$$

Si tratta di una funzione non lineare; spesso viene scelta per i suoi vantaggi in termini di tempo di computazione e per la scarsa influenza che ha la funzione di attivazione nel risultato se si tratta di un livello nascosto. Per questo solitamente si preferisce utilizzare la funzione di attivazione più complessa dal punto di vista del calcolo solo sul livello di output.

La scelta per le funzioni di attivazione di output dipende dalla tipologia di classificazione che si va ad affrontare: se binaria si utilizza la Sigmoide, mentre nel caso di multi-classificazione si utilizza la Softmax.

La Sigmoide prende in input un valore $x \in \mathbb{R}$ presente nel neurone di output e restituisce un valore $x \in [0, 1]$. Se x è maggiore di un limite, solitamente 0.5, allora l'esempio appartiene alla classe 1, altrimenti alla classe 0. É formalmente definita con la formula:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.14}$$

Nel caso della Softmax, verrà prodotto un vettore di probabilità. La classe classificata dalla rete neurale corrisponde al risultato $argmax(softmax(z_i))$, ovvero l'indice del valore più grande presente nel vettore, viene definita nel seguente modo:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{2.15}$$

Funzione di ottimizzazione

Spesso nel contesto dell'ottimizzazione si usa la tecnica della discesa del gradiente stocastica, ma, considerati i grandi costi computazionali, non è sempre possibile calcolare il gradiente di tutti gli esempi contemporaneamente; tuttavia, anche se approssimato, il gradiente riesce comunque a dare dei buoni risultati, rallentando però il processo di discesa, come possiamo vedere in Figura 2.5.

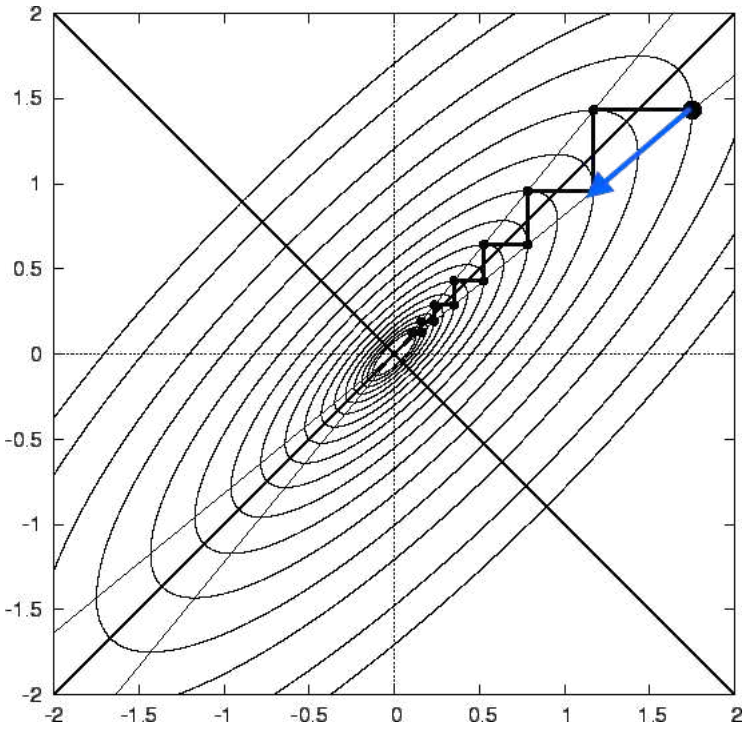

Figura 2.5: Discesa di gradiente stocastica

Esistono vari algoritmi di ottimizzazione che provano a velocizzare la discesa; un esempio è l'Adam, l'algoritmo maggiormente utilizzato nello stato dell'arte e figlio dell'approccio misto ispirato sia a Momentum che RMSprop.

L'algoritmo di ottimizzazione Adam tiene traccia dello spostamento esponenziale medio del gradiente, avente come primo momento m (media del gradiente) e la radice quadrata del gradiente come secondo momento v (varianza decentrata). L'indice t corrisponde al time step dell'algoritmo e, tramite le seguenti formule, è possibile calcolare la stima con bias del primo e del secondo momento al punto t:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{2.16}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{2.17}$$

I parametri β_1 e β_2 controllano per quanto tempo conservare le informazioni passate: maggiore sarà il valore di β, tanto più dovrà essere tenuta presente la storia del gradiente. Al crescere del timestep aumenta anche il bias, quindi si usa la seguente formula per rimuoverlo:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.18}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.19}$$

infine, i parametri del gradiente vengono aggiornati con la seguente formula:

$$\theta_t = \frac{\theta_{t-1} - \alpha \cdot \hat{m}_t}{\sqrt{(\hat{v}_t)} + \varepsilon} \tag{2.20}$$

RNN: Reti neurali ricorrenti

Le reti neurali ricorrenti sono dei sistemi che sono stati adattati per la gestione di sequenze di dati. Una sequenza è un insieme di valori $x_1, x_2, x_3, ...x_\tau$ dove il valore di ogni singolo elemento dipende anche dal valore degli elementi vicini. Solitamente più due elementi tendono ad essere vicini più sono dipendenti. Questa dipendenza può essere monodirezionale, da sinistra verso destra o da destra verso sinistra, oppure bidirezionale.

Quando un elemento della sequenza è influenzato dagli elementi precedenti, si chiama causalità. Se si prende un qualsiasi testo, come ad esempio "Il cane scodinzola felice", è possibile dire con certezza che la parola "scodinzola" o la parola "felice" sono fortemente influenzate dalle parole presenti prima. In un qualsiasi testo, la parola n sarà certamente influenzata dalle parole $n-1, n-2, n-3, etc.$ Questa informazione non è presa in considerazione nelle normali reti neurali, quindi è nata la necessità di implementare una struttura più complessa che potesse cogliere pienamente la proprietà della causalità.

Un altro limite delle reti neurali è la necessità di standardizzazione l'input; in una normale rete neurale, se c'è la necessità di classificare un insieme di frasi occorre che queste abbiano lo stesso numero di parole, altrimenti si avranno degli esempi con frasi troncate e altri con neuroni di input vuoti.

Una soluzione ad entrambi questi problemi sono le reti neurali ricorrenti, delle reti fondamentali nel mondo del NLP. Un esempio di questo tipo di rete può essere visto in Figura 2.6.

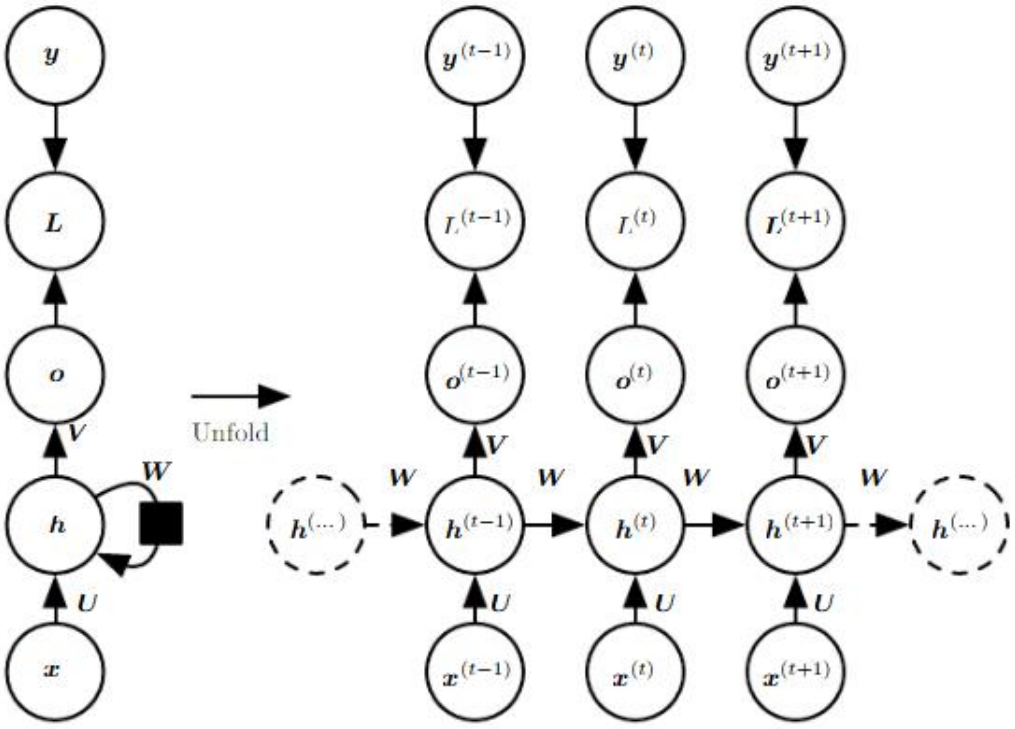

Figura 2.6: Esempio di una rete neurale ricorrente.

La rete neurale ricorrente prende in input una sequenza di x con indice t; ogni livello nascosto h_t è definito dal livello nascosto h_{t-1} e dal livello di input t, eccezion fatta per il primo elemento in input, che non può essere influenzato dal livello nascosto precedente. In maniera formale è possibile definire in questo modo una rete ricorrente:

$$h^t = f(Wh^{t-1} + Ux^t + b) \tag{2.21}$$

$$h^0 = Ux^0 + b \tag{2.22}$$

$$o^t = g(Vh^t + c) \tag{2.23}$$

Si possono adottare diverse strutture di reti ricorrenti per risolvere diversi problemi: classificazione, traduzione, generazione di una sequenza, etc.

Uno dei limiti più grandi della rete neurale ricorrente è l'esplosione o la scomparsa del gradiente: andando avanti con una sequenza si andrà a moltiplicare il vettore dei pesi W per sè stesso molte volte.

$$\frac{\partial h}{\partial h_{t-3}} = W^T \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{bmatrix} W^T \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{bmatrix} W^T \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_r \end{bmatrix} \tag{2.24}$$

Se lo spectral radius $\rho(W) > 1$

$$\lim_{k \to \infty} W^k = 0 \tag{2.25}$$

Se lo spectral radius $\rho(W) < 1$

$$\lim_{k \to \infty} W^k = \infty \tag{2.26}$$

La formula per calcolare il gradiente è la seguente: a seconda se i valori $\frac{\partial h^t}{\partial h^\tau}$ tendono a divergere o convergere, il gradiente esploderà o sparirà.

$$\frac{\partial L}{\partial W} = \sum_i \frac{\partial L}{\partial h^t} \frac{\partial h^t}{\partial h^\tau} \frac{\partial h^\tau}{\partial W} \tag{2.27}$$

Entrambi i casi portano all'incapacità della rete di poter apprendere; le reti neurali hanno dei grossi limiti nell'apprendimento quando le sequenze iniziano ad avere una lunghezza importante, considerato anche il fatto che le stesse informazioni del passato vengono perse perché "sovrascritte" da informazioni nuove. Da questo è nata la necessità di introdurre reti ricorrenti più efficienti nel conservare informazioni passate.

LSTM e GRU

Una soluzione alla perdita di informazione nel RNN è la LSTM; il suo funzionamento è simile a una RNN ma con l'utilizzo di 3 gate:

- Input gate: all'input viene permesso di essere salvato in memoria.

- Output gate: il valore presente in memoria può essere considerato nel calcolo dell'output.
- Forget gate: la cella di memoria viene resettata e la rete dimentica quello che ha imparato.

La GRU funziona in maniera simile, ma è composta solamente da due gate:

- Update gate: definisce quando il livello nascosto può essere allenato.
- Forget gate: definisce in che situazioni il livello nascosto può essere ignorato.

La GRU ha in generale prestazioni peggiori della LSTM, però offre migliori prestazioni computazionali. Filtrando le informazioni inutili e lasciando passare solo quelle più importanti, si mitiga l'esplosione o la scomparsa del gradiente. Entrambe sono basate sul concetto di far memorizzare solo le informazioni più importanti, andando a scartare quelle considerate inutili.

Transformer encoder

Il transformer [16] è una rete neurale che sfrutta il meccanismo di attenzione per la trasformazione sequence to sequence, ed è largamente utilizzato nel contesto del natural language processing. Presenta notevoli vantaggi rispetto alle classiche reti ricorrenti GRU e LSTM, uno tra tutti il fatto di poter parallelizzare l'input immesso ed essere più efficiente dal punto di vista computazionale.

Un altro vantaggio è sicuramente la possibilità di considerare sia il contesto sinistro, $n-1, n-2, etc.$ che il contesto destro, $n+1, n+2, n+3$, etc. Esistono architetture RNN che possono fare lo stesso, ma non contemporaneamente, visto che la RNN bidirezionale prima controlla il contesto sinistro, poi il destro concatenandoli tra di loro successivamente.

Si prenderà in considerazione solo la parte encoder del transformer perché è quella utilizzata da BERT, modello che verrà introdotto più avanti.

I concetti principali su cui si basa il transformer encoder sono il meccanismo di auto attenzione e il positional embedding; il meccanismo di attenzione permette al programma di focalizzarsi solo sulle parti definite importanti mentre il positional embedding considera anche la posizione nella codifica della parola.

Gli step che compie un testo all'interno di un transformer per essere codificato sono i seguenti:

- Il primo step consiste nel codificare tramite un algoritmo di embedding la frase, che poi viene sommata al vettore posizionale. Questo permette di assegnare a parole uguali ma con posizione diversa una differente rappresentazione.
- Successivamente si applica il multihead self attention sul vettore. L'attenzione è un meccanismo che permette di focalizzarsi solo sulle parti più importanti del testo; il meccanismo di auto attenzione invece permette di capire in che maniera le parole di una frase sono correlate alle parole della stessa frase. Viene definita multi attention perché anziché utilizzare un'unica matrice di attenzione se ne usano molteplici.

- Livello di normalizzazione (simile al batch) a cui sommo il vettore prima dell'applicazione del meccanismo di attenzione.
- Si applica una feedforward network, con una ReLU come funzione di attivazione. Formalmente definita:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b \tag{2.28}$$

- Un altro livello di normalizzazione

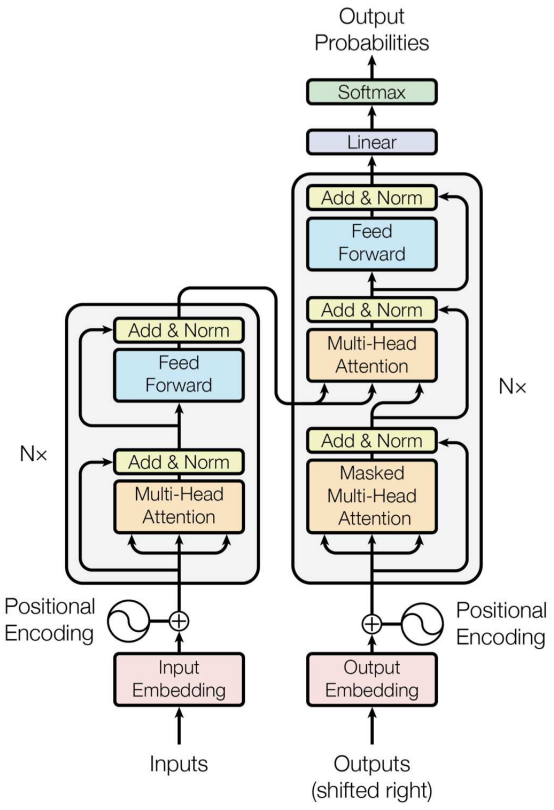

Figura 2.7: Esempio di una rete neurale ricorrente.

BERT Bi-direction encoder representation for transformer

Il modello BERT o Bidirectional encoder representation for transformer [7] è un modello NLP basato sul concetto di transformer encoder. Ha un notevole livello di comprensione del linguaggio, grazie anche alla tecnica del masked ml che sfrutta il mascheramento delle parole per forzare il modello a predirle secondo il contesto. Nella fase di allenamento del modello BERT, una sequenza di testo è allenata sia da sinistra verso destra che da destra verso sinistra; applicando la bidirezionalità si ottengono ottimi risultati.

Il processo di allenamento può essere diviso in due parti:

- Pretrain: questa fase è utilizzata dal modello per provare a comprendere la lingua a cui siamo interessati. Vengono fornite in input grandi quantità di dati testuali e sono utilizzati per comprendere contesto e significato di ogni parola. Per fare questo sono fornite in input due frasi diverse, che possono essere correlate tra di loro. Il modello si allenerà a capire la correlazione tra due frasi diverse e contemporaneamente il significato delle parole secondo il contesto.

- Fine tuning: il modello si specializza in un singolo task modificando solo leggermente i propri parametri; questa fase è molto più rapida rispetto al Pretrain ed è particolarmente efficace anche se non si dispone di grandi quantità di dati

Questa divisione dei compiti risulta essere un approccio vincete perché alleggerisce il carico di lavoro per chiunque implementa questi modelli: buona parte dell'apprendimento e della comprensione della lingua (fase di pretrain) viene effettuata da centri di ricerca o aziende con le adeguate risorse; mentre con il processo di fine tuning l'utente ha la possibilità di specializzare il modello su un determinato task. Una simile portabilità ha favorito la diffusione di questi modelli pre allenati.

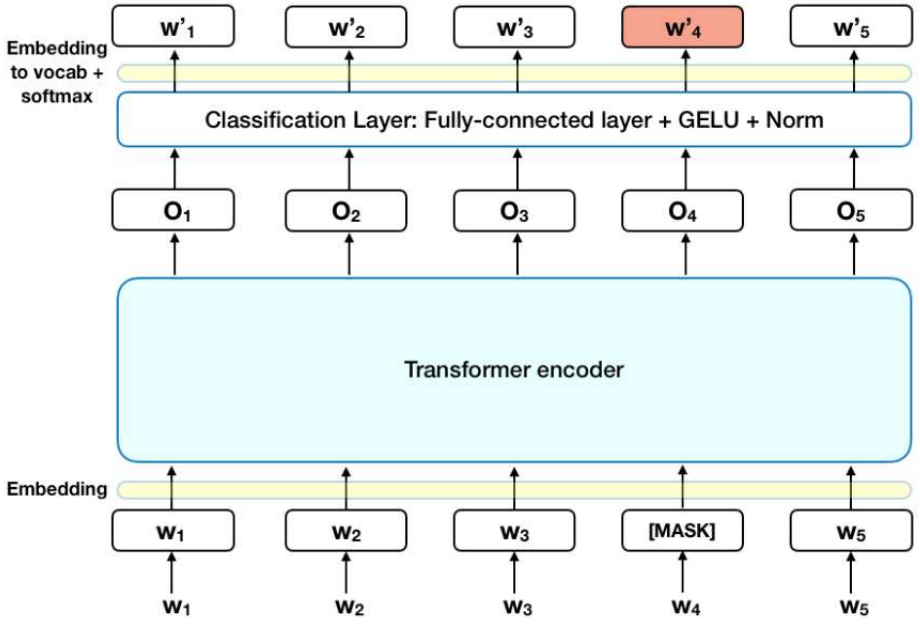

Figura 2.8: BERT structure.

Pretrain BERT

La prima parte del processo di pretrain è l'embedding dell'input testuale; per fare ciò si considerano 3 tipi di embedding, come possiamo vedere in Figura 2.9.

Prima di questo, si passa in input un vettore di parole, definendo quali parole saranno mascherate e inserendo i due token: [CLS] e [SEP]. [CLA] indica l'inizio di una frase mentre [SEP] indica quando si passa dalla frase 1 alla frase 2.

- Token embedding: codifica classica di una parola.

- Sentence embedding: definisce quale delle due frasi sia.
- Positional embedding: considera la posizione della parola, esattamente come il transformer.

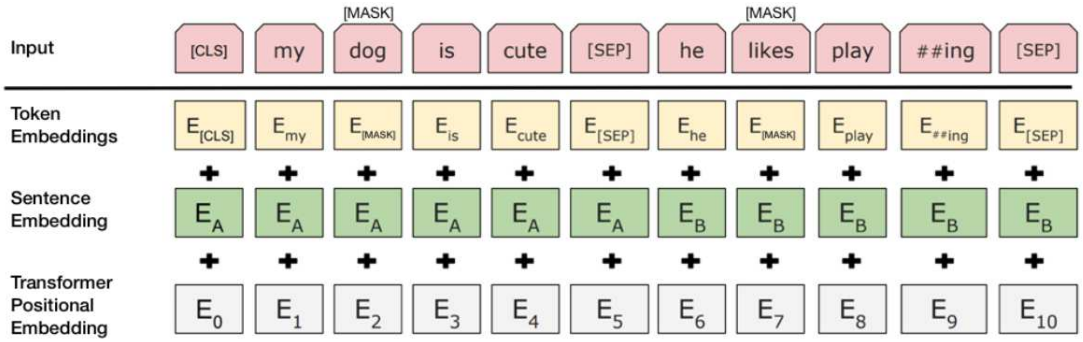

Figura 2.9: Input embedding.

Nella fase successiva del pretrain, si allena il modello a compiere contemporaneamente due task: Masked Language Model (MLM) e Next Sentence Prediction (NSP). Nel primo caso si fornisce in input un testo con delle parole, a cui viene poi applicata una maschera. Nel secondo invece si allena il modello a predire se le frasi fornite in input sono correlate oppure no.

L'allenamento viene svolto attraverso una serie di transformer encoder posizionati in sequenza. Esistono due tipologie di modelli BERT a seconda del numero di transformer messi in sequenza:

- $BERT_{BASE}$: 12 encoder, con un numero di parametri totali uguale a 110 milioni.
- $BERT_{LARGE}$: 24 encoder, con 340 milioni di parametri.

In output vengono usate solo due tipologie di informazioni:

- il token C che contiene l'informazione riguardante la correlazione tra le due frasi
- vettori delle parole nascoste, che vengono estratti dal modello utilizzando unicamente le informazioni fornite dal contesto. Come si può vedere in Figura 2.10

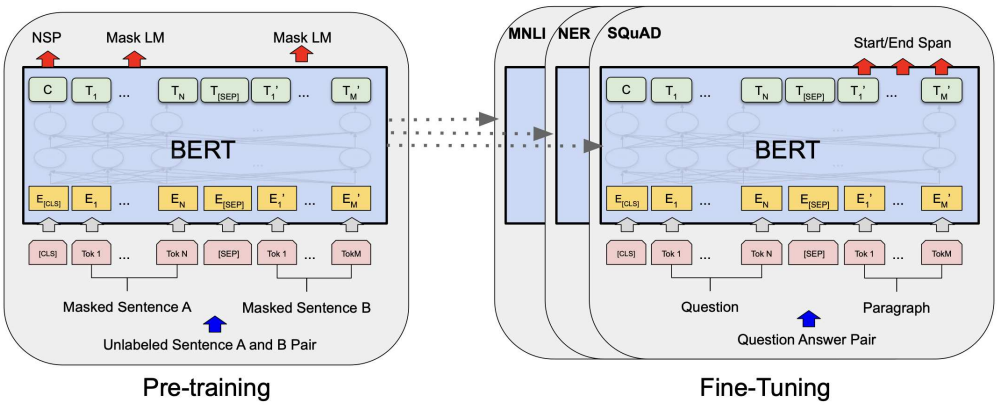

Figura 2.10: Pretrain e fine-tuning.

Fine tuning BERT

Il processo di fine tuning consiste nello specializzare il modello su un task; è un'operazione molto meno dispendiosa dal punto di vista computazionale.

Solitamente i parametri degli encoder all'interno del BERT vengono solo leggermente modificati. Viene invece allenato da zero il livello denso che connette il token [CLS] al neurone di output.

Solitamente nell'approccio classico si permette all'algoritmo di modificare tutti i parametri possibili. Questo in alcune situazioni può portare ad un overfitting. Esistono differenti soluzioni per mitigare questo fenomeno.

Transfer learning

Effettuando il fine tuning, ovvero prendendo un modello già allenato nella comprensione del significato di una determinata lingua, si può, con delle piccole modifiche, applicare anche il transfer learning.

Il transfer learning si applica quando si utilizza un modello allenato su uno specifico task A per predire anche un determinato task B, cambiando solo gli ultimi livelli della rete utilizzata. Ovviamente A e B devono essere abbastanza simili. Questo ha diverse ragioni: in primo luogo aumenta la generalizzazione del problema, rendendolo meno specificamente rivolto al task A, e in secondo luogo, in alcuni casi, può aiutare le performance del modello, perché informazioni ottenute tramite l'allenamento del task A posso essere utili anche nella risoluzione del task B.

Il modello BERT, impiegato come classificatore, usa un livello denso connesso al token [CLS] ed ha un numero di neuroni di output uguale alla quantità di classi. Nel caso si voglia applicare il transfert learning cambiando il problema di classificazione, è sufficiente sostituire il livello denso.

Ensemble methods

L'idea dietro i metodi di ensemble è quella di creare differenti modelli per poi farli votare, prendendo come output il voto medio. Questo viene fatto per differenti ragioni: per prima cosa ha un effetto di regolarizzazione, essendo il risultato della

scelta di n modelli -per cui il peso che ha un evento eccezionale è inferiore-, e in secondo luogo perché può portare anche ad un aumento delle prestazioni.

Questo avviene principalmente perché differenti modelli non commettono gli stessi errori, e quindi un modello può predire correttamente differenti esempi di uno stesso dataset; con il modello di ensemble si prova a cogliere tutti gli esempi che da un singolo modello non sarebbero predetti.

Tendenzialmente un modello di ensemble funziona meglio quando la varianza dei singoli modelli è maggiore cosicchè tutti gli esempi, o quasi, vengono predetti correttamente da almeno uno dei modelli presenti.

Nell'approccio classico si associa peso 1/n a tutti i modelli, ma si può anche scegliere di dare pesi diversi in base alle prestazioni.

Si può dimostrare che i modelli performano meglio quando variano molto tra di loro: si prendono k regressioni lineari e si suppone che il modello ha errore ε su ogni esempio, e che l'errore sia definito come una distribuzione normale multi variata con media 0, con varianza $\mathbb{E}[\varepsilon_i^2] = v$ e covarianza $\mathbb{E}[\varepsilon_i \varepsilon_j] = c$

$$\mathbb{E}\left[\left(\frac{1}{k}\sum_{i=1}^{n}\varepsilon_i\right)\right] = \frac{1}{k^2}\mathbb{E}\left[\sum_i\left(\varepsilon_i^2 + \sum_{i\neq j}^{n}\varepsilon_i\varepsilon_j\right)\right] = \frac{1}{k}v + \frac{k-1}{k}c \tag{2.29}$$

Nel caso in cui l'errore sia perfettamente correlato con c=v, la scelta media non aiuta; se l'errore invece non è correlato e c=0, ci si aspetta che l'errore quadratico sia solo $\frac{1}{k}v$

L'albero di decisione

Gli alberi di decisione sono algoritmi di apprendimento supervisionato; sono largamente utilizzati in campo medico e finanziario per la loro semplicità e interpretabilità. Possono essere visti come un insieme di regole del tipo: "se il parametro A>0 e il parametro B<0 e il parametro C=0, allora la classe dell'esempio è la classe 1".

La struttura di un albero di decisione (Figura 2.10) è composta principalmente da 3 elementi:

- Nodo: definisce il parametro scelto e il suo valore per la classificazione dell'esempio.
- Ramo: connette due nodi o un nodo e una foglia. Il ramo percorso dall'esempio dipende dalla scelta del nodo padre.
- Foglia: definisce in che modo è classificato l'esempio che raggiunge quella foglia.

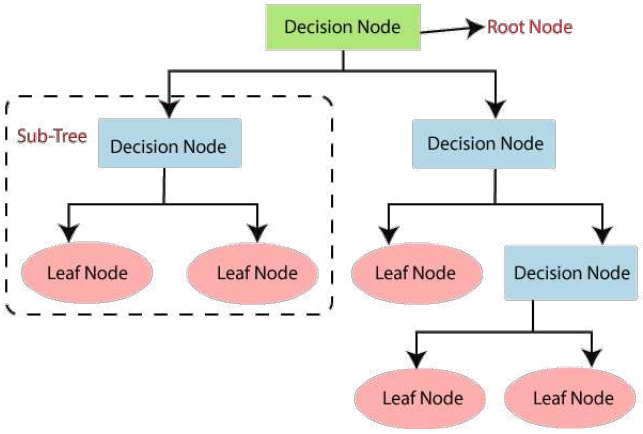

Figura 2.11: Struttura albero di decisione.

Ogni esempio è classificato partendo dal nodo padre e discendendo l'albero lasciando ogni nodo decidere se percorrere il ramo sinistro o il destro. Una volta raggiunta la foglia, la classificazione corrispondente sarà quella definita dalla foglia.

Un punto fondamentale è la scelta del parametro nel albero di decisione: questo influenzerà profondamente l'efficienza della classificazione. Esistono diversi modi per definire l'attributo ottimale per dividere un insieme di dati; uno di questi è l'information gain, un valore che dipende fortemente dall'entropia.

L'entropia può essere definita come il grado di impurità di un esempio S, con S_c il subset di classe C di S:

$$E(S) = \sum_{c=1}^{m} \frac{|S_c|}{|S|} \log(\frac{|S_c|}{|S|}) (2.30)$$

che combinata con l'information gain:

$$G(S, a) = E(S) - \sum_{v \in V}^{m} E(S_{a=v}) \tag{2.31}$$

l'information gain definisce il nodo da inserire, ovvero quello che lo fa crescere maggiormente.

2.1.6 Metriche di valutazione

Matrice di confusione

La matrice di confusione è un elemento fondamentale nell'analisi dell'errore; permette di visualizzare specificatamente quale tipologie di errori commette il modello, a differenza dell'accuratezza che ci dà un risultato generale di quanto corretto è il modello.

Come possiamo vedere in Figura 2.12 la matrice di confusione mostra quante volte l'esempio della classe 1 è predetto correttamente, e quante volte erroneamente. Stessa cosa per la classe 2.

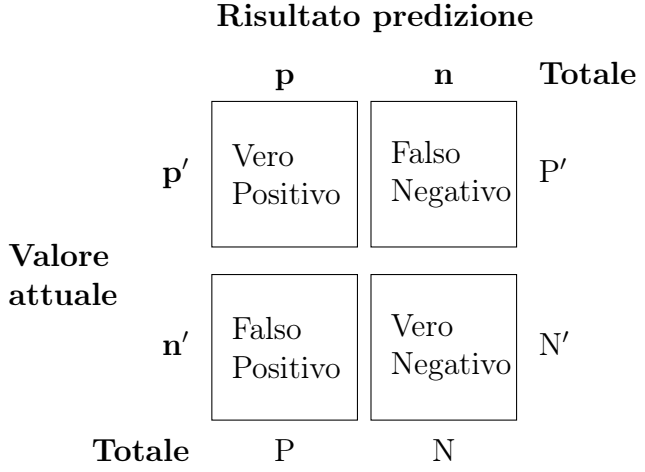

Figura 2.12: Esempio matrice di confusione.

In questo modo le predizioni giuste vengono raggruppate nella diagonale (vero positivo, vero negativo), mentre al di fuori della diagonale ci sono i falsi positivi e negativi. Più i valori vengono raggruppati nella diagonale, migliori sono le performance del modello; la formula dell'accuratezza infatti è il rapporto tra gli elementi della diagonale e tutti gli elementi presenti nella matrice:

$$\text{Accuratezza} = \frac{\text{VP+VN}}{\text{VP+VN+FN+FP}} \tag{2.32}$$

F1-score

Come già accennato, l'accuratezza ha dei limiti, come, ad esempio, nel caso si lavori con classi fortemente sbilanciate. Supponendo ad esempio di lavorare con un task che ha 99 esempi per la classe 1 e 1 per la classe 2, un modello che predice sempre la classe 1 ha un'accuratezza di 0.99. Per questo motivo è nata la necessità di utilizzare una metrica che considerasse anche questo fattore. L' F1-score o gli F-score in generale considerano per ogni modello due elementi, la recall e la precisione.

$$Precision = \frac{VP}{VP + FP} \tag{2.33}$$

La precisione misura il numero di elementi etichettati positivamente su tutti gli elementi che sono stati etichettati (correttamente o erroneamente) con quella classe.

$$Recall = \frac{VP}{VP + FN} \tag{2.34}$$

Recall o richiamo è il numero di elementi etichettati positivamente rispetto al numero totale di elementi che appartengono effettivamente a quella classe.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * VP}{2 * VP + FP + FN} \tag{2.35}$$

F1-score è la media armonica tra precision e recall, e solitamente è la metrica maggiormente utilizzata in caso di competizioni o nella pubblicazione di articoli.

2.2 Scienza dialogica

La Scienza Dialogica [10] è la disciplina che si pone l'obiettivo di analizzare e studiare le configurazioni discorsive generate nell'interazione tra esseri umani e tra esseri umani e macchine tramite l'uso del linguaggio naturale. L'oggetto di conoscenza della Dialogica è il dato testuale (o discorsivo), ovvero ciò che si manifesta nell'impiego del linguaggio naturale, in ogni sua forma e codificazione (grafica, fonetica). Ogni produzione discorsiva rappresenta così del "testo" che crea degli "eventi discorsivi" (le configurazioni discorsive) dotati di statuto di realtà e passibili di rilevazione e di misurazione. La Dialogica si occupa, dunque, di formalizzare, nell'uso del rigore scientifico, le regole del linguaggio naturale che consentono di configurare discorsivamente ciò che per senso comune viene definito come "realtà". A partire dalla formalizzazione delle regole del linguaggio (definite Repertori Discorsivi), la Dialogica ha la facoltà di misurare il processo discorsivo, tramite specifici indici, e di rilevare l'impatto che le configurazioni di realtà hanno per la comunità umana ed i suoi membri. Il processo di formalizzazione ha dato origine a 24 repertori discorsivi descritti ciascuno rispetto alle relative proprietà processuali (di generare senso) e attraverso queste regole si procede nel processo di denominazione di stralci di testi.

Per repertorio discorsivo si intende:

"una modalità finita di configurazione della realtà, linguisticamente intesa, con valenza pragmatica, che raggruppa anche più enunciati (denominati "arcipelaghi di significato"), articolata in frasi concatenate e diffusa con valenza di asserzione di verità, volta a generare (configurare)/mantenere una coerenza narrativa".

L'insieme dei Repertori Discorsivi costituisce "l'alfabeto" del linguaggio formalizzato impiegato dalla Dialogica per descrivere gli usi che i parlanti fanno del Linguaggio Naturale. A tal proposito si sottolinea come i repertori siano di numero finito, ma le modalità con cui si possono combinare sono infinite, rendendo in questo modo possibile descrivere qualsiasi configurazione discorsiva generata attraverso il Linguaggio Naturale. Al momento si dispone di 24 repertori discorsivi che, a loro volta, sono organizzati in una tavola periodica semi-radiale. L'organizzazione semi-radiale della tavola permette di descrivere filogeneticamente la sintesi degli elementi in essa contenuti.

I repertori sono rappresentabili su una Tavola periodica semi-radiale, come possiamo vedere in Figura 2.13

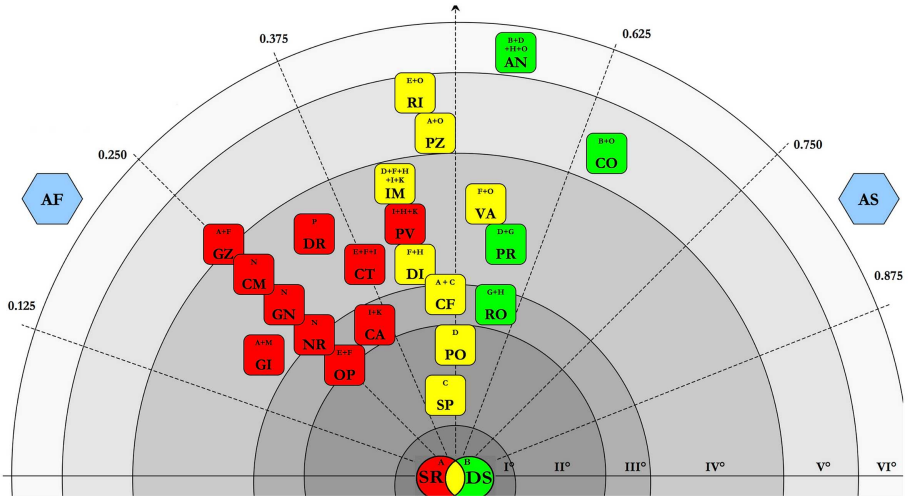

Figura 2.13: Tavola radiale repeortori.

A partire dall'interazione fra le proprietà processuali "primordiali" (nella tavola indicate dalle lettere "A" e "B"), si coniano via via tutte le ulteriori modalità d'uso del linguaggio. Le proprietà processuali di ogni repertorio discorsivo sono definite tramite proposizioni che descrivono i criteri conoscitivi che permettono di isolare le proprietà stesse nel momento della denominazione. Ciò implica che nel corso dell'analisi della produzione discorsiva uno stralcio di testo, considerato come modalità finita di configurazione della realtà, può essere denominato tale in base alla saturazione delle proprietà processuali descritte nella tavola periodica.

A seconda delle loro proprietà processuali, i repertori discorsivi sono suddivisi in tre gruppi:

- repertori discorsivi generativi, definiti come regole dell'uso del linguaggio naturale che si caratterizzano per conservare e promuovere una spinta verso la generazione di configurazioni discorsive inedite e la riconfigurazione di configurazioni già disponibili;

- repertori discorsivi di mantenimento, definiti come regole dell'uso del linguaggio naturale che concorrono a mantenere le configurazioni discorsive "identiche a loro stesse" rispetto alle proprietà processuali;

- repertori discorsivi ibridi, definiti come regole dell'uso del linguaggio naturale che possono assumere un orientamento sia di mantenimento sia generativo, non apportando singolarmente né la possibilità della generazione di configurazioni di senso diverse da quelle in corso (variabilità del processo discorsivo), né la possibilità del mantenimento di quanto si sta configurando (stabilità del processo discorsivo). Essi assumono e aggiungono un valore di generazione o di mantenimento a seconda della classe di appartenenza dei repertori con cui si trovano a interagire nella configurazione.

Il gruppo teorico dei repertori discorsivi artificiali denota regole d'uso del linguaggio naturale che generano/mantengono/configurano realtà per affermazione (AF) o per asserzione (AS).

Ogni repertorio discorsivo è caratterizzato da parametri che consentono di ripartire lo spazio discorsivo: la generatività e la dialogicità. A ciascuno di essi è stato attribuito un valore numerico in virtù dell'interazione fra le proprietà processuali disponibili.

- La generatività, misurata tramite unità di peso dialogico, rappresenta il contributo generativo di "realtà discorsive" potenziali che ogni repertorio apporta nelle genesi di una configurazione discorsiva rispetto al gruppo cui appartiene (repertori generativi, di mantenimento, ibridi). Tale grandezza indica un valore che proviene direttamente dalle proprietà processuali che danno assetto al repertorio stesso.

- La dialogicità, misurata tramite unità di momento dialogico, rappresenta la forza di legame tra le proprietà processuali che intelaiano i repertori, mettendo a disposizione la possibilità di rilevare quanto una configurazione discorsiva può essere "flessibile", dunque modificabile. Questa grandezza rende conto della capacità propria dei repertori di legarsi fra loro nella tessitura delle produzioni discorsive, mantenendo narrativamente coesa e coerente la configurazione che si genera.

In ambito di ricerca e di intervento, una volta raccolta la configurazione discorsiva questa viene analizzata utilizzando la formalizzazione rappresentata dalla tavola semi-radiale, al fine di misurare il peso dialogico e il momento dialogico della stessa. Grazie a tali operazioni, partendo dal dato osservativo raccolto, il ricercatore è in grado di disporre di una descrizione e di una misura dell'impatto di una configurazione sulla realtà del senso, di anticipare le possibili configurazioni discorsive che si andranno a generare, nonché di progettare e implementare anche un eventuale piano operativo dell'intervento.

2.2.1 MADIT

Il processo di denominazione dei repertori è definito dalla metodologia MADIT [14], ed è formato dai seguenti passaggi:

- Porsi la domanda che ha generato la risposta

- Anticipare le configurazione discorsive e provare a redigere le possibili risposte alla domanda

- Enunciare i passaggi argomentativi e individuare gli snodi argomentativi che permettono di dividere le porzioni di testo

- Leggere la risposta data dal gruppo di rispondenti

- Denominare gli stralci di testo attraverso la definizione dei singoli repertori

- Individuare gli elementi del contenuto che possono definire un repertorio

Capitolo 3

Lavoro pregresso e primi test

Questa tesi è la prosecuzione di un lavoro già precedentemente svolto, in un primo momento durante una borsa di studio di ricerca in collaborazione col dipartimento di Filosofia, Sociologia, Pedagogia e Psicologia Applicata e poi con la tesi di un mio collega, Michele Bortone.

Come già accennato, l'obbiettivo di questa tesi è verificare l'utilità della classificazione dei repertori nei principali task di classificazione testuale. Questo comporterà l'implementazione di diversi modelli BERT.

Per fare ciò si prenderà in considerazione un determinato task di classificazione del testo, come ad esempio l'ironia. Preso un testo che può essere ironico o meno, verrà implementato un modello BERT per dividerlo in stralci di testo, e successivamente verrà adottata sempre la stessa tipologia di modello per la classificazione degli stralci di testo nei rispettivi repertori.

Fatto questo si avrà una rappresentazione del testo sotto forma di repertorio che servirà per la classificazione dello stesso.

3.0.1 Divisione testo

La classificazione dei repertori ha una fase fondamentale, che nell'approccio teorico è svolta parallelamente alla classificazione: la divisione del testo. Secondo la teoria, un testo completo si compone di diversi stralci di testo; ogni stralcio corrisponde ad un repertorio.

Nell'approccio MADIT l'operazione di classificazione e divisione del testo avviene contemporaneamente. In questo contesto invece viene fatto in sequenza: come prima cosa un modello BERT divide il testo utilizzando la "Next Sentence Prediction" , e successivamente con un altro modello classifica tutti gli stralci di testo.

Visto che dagli esperimenti svolti dal mio collega si può notare come una divisione perfetta del testo comparato ad una mediocre porta un miglioramento minimo nella classificazione dei repertori, questa parte non è stata esplorata ulteriormente e mi sono limitato ad implementare lo stesso modello del mio collega [5].

3.0.2 Classificazione dei repertori

La classificazione dei repertori è il concetto fondamentale su cui si basa questa tesi. Concretamente si prova ad aiutare un modello nella classificazione di alcuni task NLP utilizzando i repertori come step intermedio, ovvero si verifica se in qualche modo queste regole linguistiche possano aiutare ad esplicitare delle informazioni che per il modello possono essere interessanti. Dopo aver definito un modello che classifichi i repertori, si cercherà di integrare le informazioni estratte dal modello e si verificherà se possono essere utili in contesti di classificazione del testo. Per fare ciò, verranno esplorate diverse tecniche finalizzate ad integrare le informazioni sfruttando il transfer learning o la concatenazione al testo della sua versione codificata secondo i repertori discorsivi. Per poter applicare il transfert learning verrà utilizzato un modello BERT per la classificazione allenandolo a predire i repertori discorsivi, successivamente si utilizzerà questo modello come pre-train del modello BERT che classificherà il testi ironici, soggettivi, polarizzati, di hate speech e stereotipali.

La classificazione di repertori è un problema di multi classificazione: dato un testo ed una domanda che lo ha generato si deve classificare in una delle 23 classi possibili.

Dataset hyperium

Per allenare il modello è stato utilizzato il dataset hyperium, composto da circa 15 mila testi; ad ogni testo è associato un repertorio discorsivo e la domanda che ha generato la risposta. Il testo è composto da articoli di giornali e tweet relativi l'argomento COVID19. Queste classi hanno diversa composizione in termini di elementi e sono estremamente sbilanciate, come si può vedere in Figura3.1. Ogni repertorio ha una difficoltà intrinseca differente, dipendente dal grado di astrazione della regola. Questo avrà una forte influenza sulla facilità di classificare un repertorio da parte del modello.

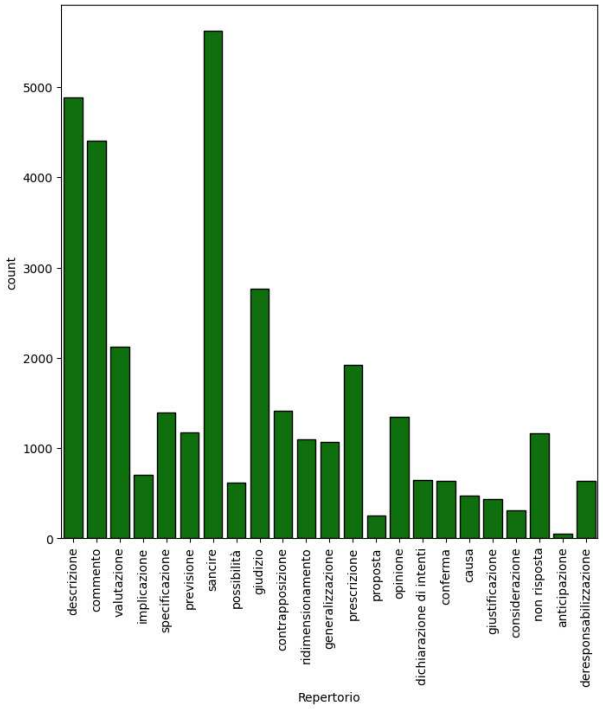

Figura 3.1: Distribuzione repertori dataset hyperium.

Dataset 2.1 e 3.1

I dataset 2.1 e 3.1 sono due dataset distinti che vengono utilizzati insieme e che contengono 14567 stralci. Sono il frutto dei risultati di alcune interviste riguardanti la scuola di psicologia di Padova ed esperienze personali. Essendo stati raccolti e organizzati tramite questionari, i testi hanno una grammatica e una sintassi abbastanza corrette.

3.0.3 Classificazione Evalita

Per provare l'utilità dei repertori utilizzeremo 6 task differenti, provenienti da due tipologie di competizione promosse dal gruppo Evalita. Per verificare il grado di generalizzazione del modello che predice l'ironia, verrà poi implementato un terzo dataset di test proveniente da una terza competizione .

EVALITA2016 (Ironia, soggettività, positività e negatività [8])

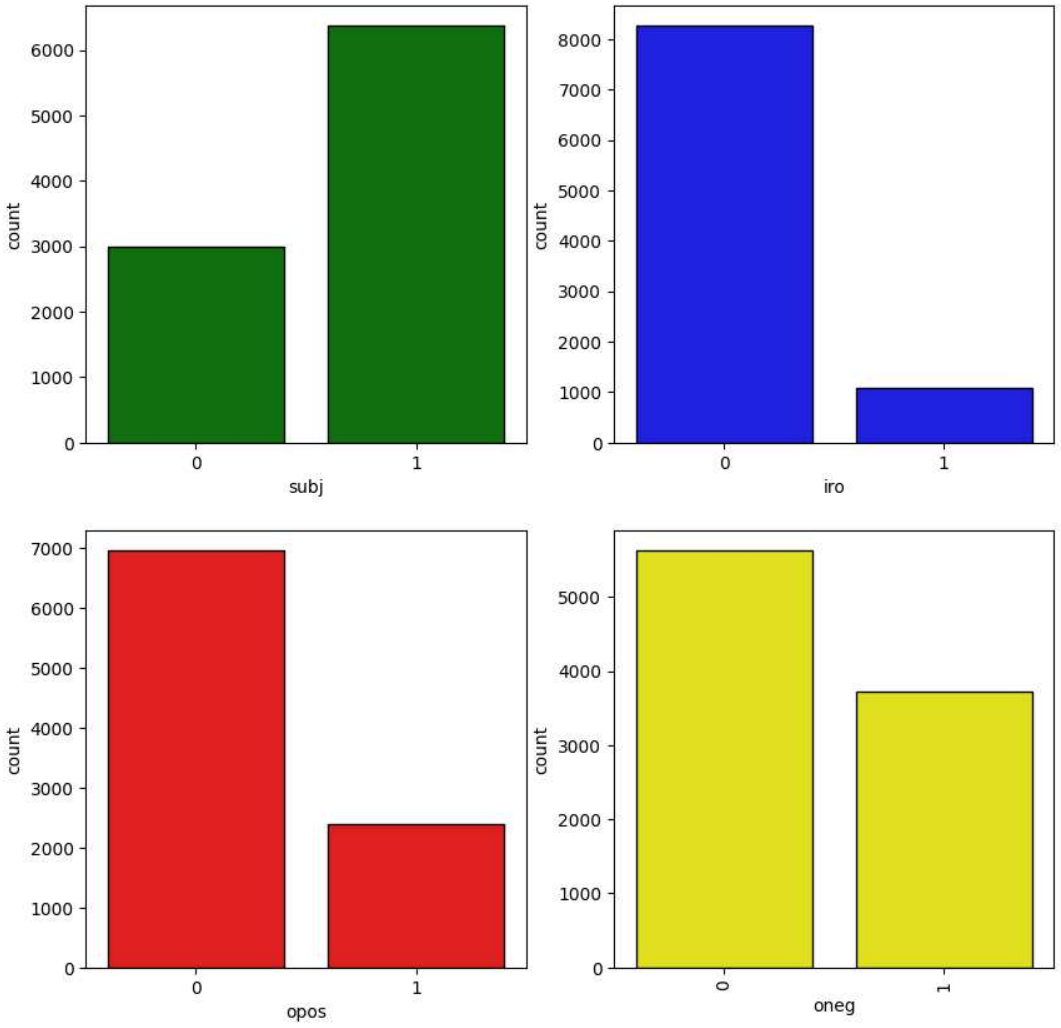

Figura 3.2: Distribuzione classi EVALITA2016 (opos: positività, oneg: negatività).

Il dataset di allenamento è composto da 7410 tweet, mentre in quello di test ve ne sono 2000.

- Ironia (iro): è il dataset maggiormente sbilanciato, dove l'ironia rappresenta circa il 10% di tutti i possibili i dati. Se un tweet è etichettato come ironico allora sarà sicuramente etichettato come soggettivo. Se nel dataset è presente il valore 1 nel campo iro allora si considera ironico e deve avere polarità o negativa o positiva, quindi se $iro = 1$ allora $opos \neq oneg.$ e $subj = 1$
- Soggettività (subj): Se un tweet è etichettato come soggettivo allora può anche non avere una particolare polarità, quindi se $subj = 1$ allora è possibile che $opos = 0$ e $oneg = 0$. Se un tweet è etichettato come oggettivo allora non è ne polarizzato ne ironico, quindi se $subj = 0$ allora $opos = 0$ $iro = 0$ $oneg = 0$.

- Positività e Negatività (opos e oneg): Definito anche come polarità, con $opos = 1$ un tweet è positivo; altrimenti se $opos = 0$ è neutro (dal punto di vista della positività). Stesso discorso per $oneg$. In un tweet può coesistere $opos = 1$ e $oneg = 1$, quindi un tweet può essere considerato sia positivo che negativo.

EVALITA2020 (Hate speech e stereotipi [3])

Una caratteristica di questo task è quella di avere due tipologie di test: uno che contiene articoli di giornali, un altro che contiene tweet. Il dataset di allenamento è composto unicamente da tweet, che sono 4000, mentre per il test sono in tutto 496.

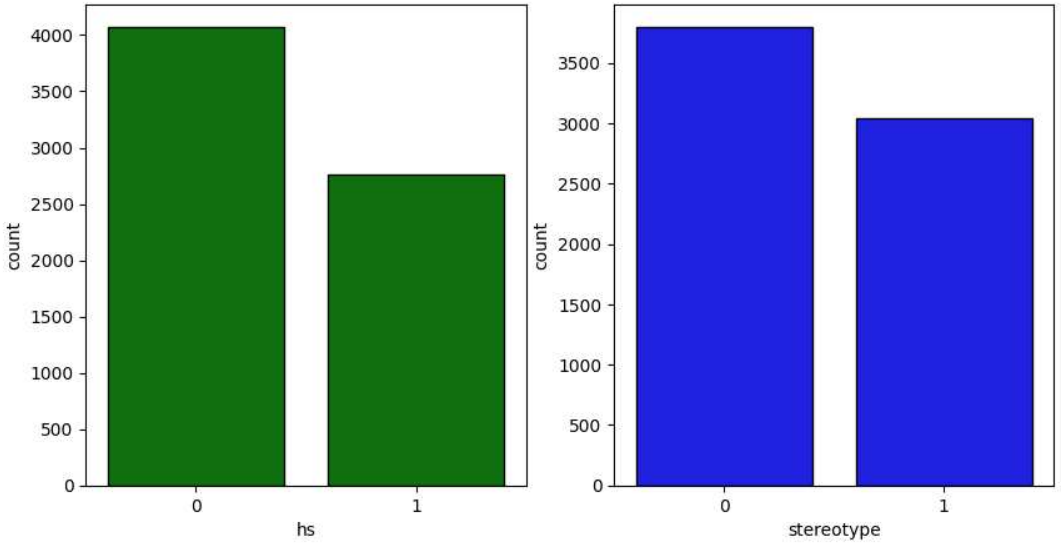

Figura 3.3: Distribuzione classi EVALITA2020

- Hate speech (hs): rileva discorso d'odio nei confronti di musulmani, immigrati e persone che vivono a Roma. Non ci sono particolari correlazioni con la presenza di stereotipi, se $hs = 0$ o $hs = 1$ allora $st = 0$ oppure $st = 1$.
- Stereotipi (st): rileva la presenza di stereotipi nel testo. Non ci sono particolari correlazione con hate speech.

IronITA2018 (Ironia e sarcasmo [6])

Questo dataset è utilizzato unicamente per testare il modello, utilizzando dei testi differenti rispetto a quelli forniti nel dataset di test dell'altra competizione. In questo caso è presente ironia o sarcasmo, ma sarà necessaria unicamente l'ironia. Una caratteristica di questo dataset di test è il bilanciamento delle due classi composte da 400 esempi a testa.

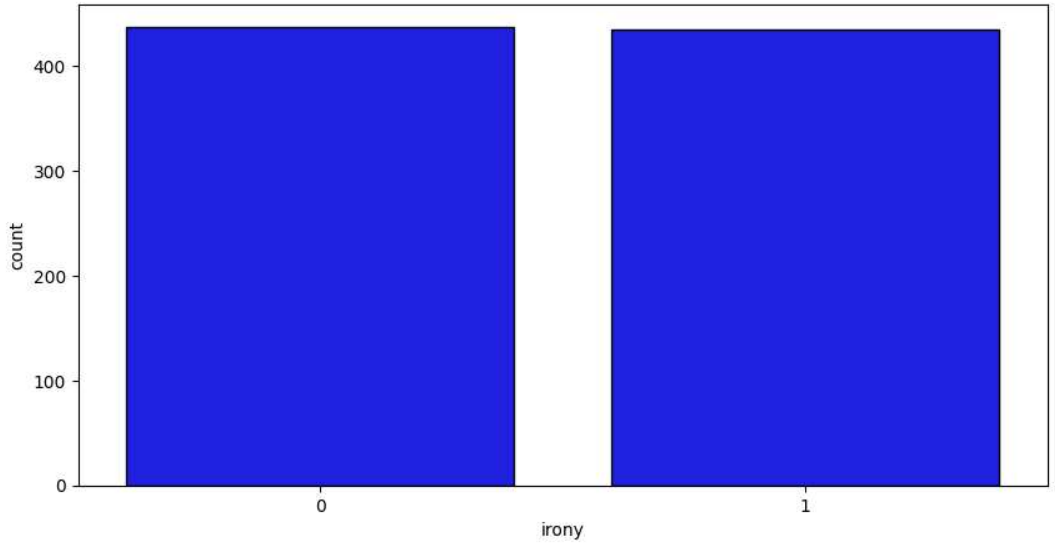

Figura 3.4: Distribuzione classi EVALITA2018

3.1 Lavoro pregresso

3.1.1 Primo lavoro esplorativo

Durante la mia borsa di studio di ricerca [12] è stato sviluppato un modello BERT per la classificazione dei 23 repertori discorsivi, utilizzando il dataset 2.1 e 3.0. Nel processo di denominazione degli stralci si possono definire due ruoli, senior e junior. Con senior si intende una persona con un'importante esperienza di studio della teoria e classificazione dei repertori, con junior invece una persona con ancora poca esperienza, che conosce la teoria dietro la scienza dialogica ma ancora non riesce a gestire correttamente la denominazione; l'errore umano è però anche correlato alla natura stessa di un repertorio, ovvero alla facilità del suo riconoscimento.

Il dataset era composto per ogni esempio da:

- Domanda che genera il testo
- Testo
- Repertorio discorsivo domanda
- Repertorio discorsivo risposta
- Natura repertorio domanda (mantenimento, ibrido o generativo)
- Natura repertorio testo

Nel dataset sono presenti intorno ai 14 mila esempi e tutte le denominazioni sono state effettuati da junior. Gli argomenti trattati sono vari: da domande riguardanti il corso di studio di Psicologia a esperienza traumatiche di soggetti fragili o vittime di dipendenze. Grazie a questo dataset si è potuto creare un modello BERT che classificasse i repertori; come dataset di pretrain si è utilizzato DBMDZ. Sono

state create due varianti: la prima che considera la domanda, la seconda invece che non la considera. Nel primo caso la domanda viene concatenata alla risposta separata dal token [SEP], andando a considerare nella classificazione il livello di correlazione tra risposta e domanda. Questo in alcuni casi può essere estremamente informativo, come nel caso della non risposta, dove il repertorio è classificato in base a come la risposta è correlata alla domanda. Nel secondo caso invece i repertori sono classificati utilizzando solo l'informazione portata dalla risposta.

I risultati sono visibili in Tabella 3.1

Classificazione repertori	Precison	Recall	F1
Con domanda	**0,35**	**0,45**	**0,38**
Senza domanda	0,30	0,39	0,32

Tabella 3.1: Risultati modello classificazione repertori.

Successivamente si è creato un dataset di real e fake news, composto da articoli online scaricati; per la parte delle real news si è utilizzato i principali quotidiani nazionali, invece per le fake news sono stati utilizzati i principali siti complottisti e di diffusione di notizie false.

Una volta creato il dataset si è utilizzata la funzione senttokenise() della libreria NTKL[4] per dividere le notizie in stralci di testo classificabili dal modello BERT, e successivamente è stato applicato il modello allenato senza la domanda per definire per ogni notizia, real o fake, la sua distribuzione in termini di repertori.

Successivamente è stata usata una random forest per classificare ogni notizia in base all'informazione fornita dai repertori. I risultati sono in Tabella 3.2

Classificazione fake news	Precison	Recall	F1
	0,75	**0,73**	**0,74**

Tabella 3.2: Risultati classificazione fake news.

Questo lavoro presentava diverse criticità, dalla modalità di estrazione dei dati al dataset stesso, però è stato indice delle possibilità offerte dall'utilizzo dei repertori combinato con le più recenti tecniche di NLP.

3.1.2 Repertori per EVALITA2016

Successivamente il lavoro è stato preso in carico da un mio collega [5], che lo ha applicato all'analisi della polarità e rilevamento di ironia e soggettività. Siccome in questo caso si tratta di tweet, è stato utilizzato il dataset hyperion, anch'esso composto da tweet.

Il lavoro svolto è visibile in Figura 3.5: prima di tutto si è allenato un modello BERT sia sulla divisione del testo che sulla classificazione dei repertori.

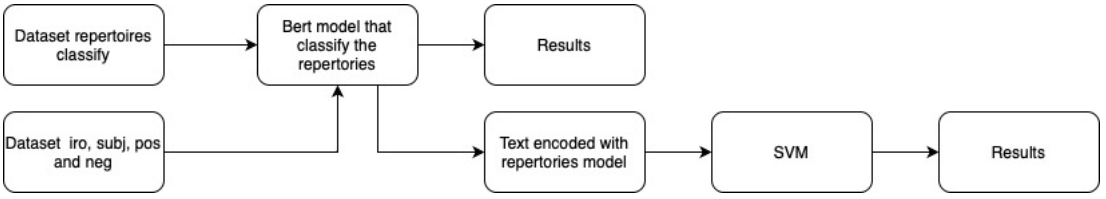

Figura 3.5: Pipeline modello.

Successivamente è stato preso il dataset contenente i testi per il task di classificazione dell'ironia, soggettività, negatività e positività e si è utilizzato il modello BERT per produrre una rappresentazione dell'esempio utilizzando i repertori. Per ogni testo si avevano 5 possibili rappresentazioni:

- Frequenza repertori relativa.
- Frequenza repertori assoluta.
- Ultimo livello del modello BERT per la classificazione dei repertori.
- Ultimi 4 livelli concatenati del modello BERT.
- Token [CLS] del modello BERT.

Una volta fatto questo, ad ogni tipologia di rappresentazione è stata applicata una support vector machine. I risultati sono stati positivi per il task dell'ironia, che supera lo stato dell'arte, come si può vedere in Tabella 3.3.

Soggettività	F1 Oggettività	F1 Soggettività	Macro F1
Alberto	**0.74**	**0.84**	**0.79**
SVM Repertori discorsivi	0.72	0.80	0.76
Ironia	F1 Non ironia	F1 Ironia	Macro F1
Alberto	**0,9408**	0,2772	0,6090
SVM Repertori discorsivi	0,93	**0,42**	**0,67**
Polarità	F1 Positività	F1 Negatività	Macro F1
Alberto	**0.74**	0.76	**0.75**
SVM Repertori discorsivi	0.71	**0.77**	0.74

Tabella 3.3: Prestazioni Alberto nella competizione EVALITA2016.

3.2 Proseguimento lavoro

Arrivati a questo punto, tenendo in considerazione alcuni buoni risultati riscontrati precedentemente, l'obiettivo è mostrare il reale contributo che la classificazione dei repertori può dare nel mondo del processamento dei linguaggi naturali.

Per ora abbiamo mostrato che per l'ironia offrono innegabilmente un importante contributo rispetto allo stato dell'arte, però questo deve essere una tendenza condivisa anche con gli altri task per dimostrare prima di tutto che i miglioramenti che offrono i repertori non sono solo frutto del caso, e in secondo luogo poter confrontare i modelli con la stessa tipologia di modello pretrained.

Nel lavoro precedente è stato confrontato un modello BERT pretrain Alberto, che rappresenta lo stato dell'arte, con un modello pretrain DBMDZ Uncased XXL utilizzato per la rappresentazione del testo poi classificata da una SVM.

Entrambi i modelli sono BERT pre allenati sulla lingua italiana. Alberto è allenato sul dataset TWITA, un corpus di tweet in italiano, mentre DBMDZ nel corpus OPUS, composto da testi più generici, non recuperati da social network e quindi con una grammatica e una sintassi più corretta.

Sono due modelli con architettura diversa: Alberto è un MODEL$_{BASE}$, ovvero composto da 12 transformer encoder, mentre DBMDZ è MODEL$_{LARGE}$, composto da 24 transformer encoder messi in sequenza.

Detto questo ci sono differenti elementi che si devono considerare: prima di tutto i testi che andremo a classificare sono tweet, quindi per questo fattore Alberto potrebbe essere avvantaggiato; però DBMDZ ha una struttura più complessa. Solitamente i MODEL$_{LARGE}$ tendono a performare meglio, anche se richiedono una potenza di calcolo maggiore.

Se si osservano i test fatti precedentemente, possiamo chiaramente vedere che il modello DBMDZ tende a performare leggermente meglio per quanto riguarda la predizione dei repertori, come si può vedere in Tabella 3.4.

Classificazione repertori	DBMDZ	Alberto
F1 pesata	**0,36**	**0,35**
F1 macro	**0,27**	**0,26**
Accuratezza	**0,37**	**0,34**

Tabella 3.4: Risultati classificazione repertori secondo modello pre-allenato.

Il primo obiettivo è quello di vedere in che maniera le prestazioni legate alla predizione dei repertori siano correlate alla classificazione dei task di EVALITA2016. Nel lavoro di tesi precedente, una volta selezionato DBMDZ come miglior modello, si è proseguito implementando unicamente tale modello per le generazioni del testo codificato.

In altre parole, l'obiettivo è trovare la correlazione tra quanto un testo viene codificato bene utilizzando la teoria della scienza dialogica e quanto bene questo testo verrà classificato nei task di ironia, soggettività, positività e negatività.

3.3 Support Vector Machine per la classificazione del testo codificato prodotto dal modello BERT

Come prima sono state prese in considerazione le prestazioni nella classificazione dei task nel caso della codifica del testo utilizzando il modello pre-allenato "Alberto". Questo è stato fatto per due motivi principali:

- Per un confronto efficace con lo stato dell'arte, che sfrutta il modello "Alberto".
- Per vedere se le prestazioni nella classificazione dei quattro task fossero coerenti con le prestazioni per la classificazioni dei repertori.

Essendo complesso portare avanti gli esperimenti per tutti i 4 i task contemporaneamente, si è deciso di portare avanti solamente gli esperimenti che riguardano la soggettività e l'ironia.

La scelta è ricaduta su questi due task per diverse ragioni: innanzitutto perché la classificazione dell'ironia era quella che forniva risultati migliori, quindi era atteso che con l'implementazione di modelli più complessi il divario di prestazioni si facesse più importante; il secondo motivo è che la soggettività e l'ironia sono fortemente legate. Quando una frase risulta ironica è anche sicuramente soggettiva. L'aspettativa è quindi che se l'ironia performa bene allora anche la soggettività dovrebbe performare bene; però questo non accade.

In Tabella 3.5 possiamo vedere i risultati: nel caso dell'ironia le prestazioni del modello diminuiscono fortemente, perdendo fino al 5% nella f1 macro, mentre invece per la soggettività si può notare una crescita importante, che però non supera lo stato dell'arte.

La codifica utilizzata per rappresentare il testo è quella che al mio collega dava i risultati migliori, cioè quella prodotta dal token [CLS].

Ironia	DBMDZ	Alberto	Stato dell'arte
F1 ironia	**0,39**	0,31	0,28
F1 no ironia	0,92	0,93	**0,94**
F1 macro	**0,65**	0,62	0,61
Soggettività	DBMDZ	Alberto	Stato dell'arte
F1 soggettività	0,77	**0,84**	**0,84**
F1 no soggettività	0,70	0,67	**0,75**
F1 macro	0,73	0,75	**0,79**

Tabella 3.5: Risultati classificazione SVM di testi codificati con token [CLS].

Capitolo 4

Integrazione informazione repertori discorsivi nel modello BERT

Come si è potuto notare dai primi esperimenti esplorativi, applicare un classificatore alla versione codificata del testo non è il migliore degli approcci, dal momento che, producendo prima la versione codificata e poi applicando una SVM, il processo di backpropagation dell'errore è "spezzato", e quindi non va a modificare la codifica del testo. Se viene utilizzato direttamente il modello BERT con un livello denso, la backpropagation dell'errore modifica anche i parametri dei transformer.

4.1 Stato dell'arte: Alberto [15]

Il modello Alberto ha segnato un punto importante nel NLP italiano: l'obiettivo era quello di creare un qualcosa che potesse comprendere il linguaggio social, ben diverso da ciò che è possibile trovare su dizionari o enciclopedie. É stato allenato con il dataset TWITA [2], composto da 200 millioni di tweet. Nel caso di Alberto è stato implementato solamente il "masked learning" e non il "next sentence prediction". Per l'allenamento sono stati utilizzati principalmente testi presi da twitter, applicando due tipologie di preprocessing:

- Normalizzazione di caratteri speciali, come URL, numeri, etc. sostituendoli con un tag che ne descrivesse la natura (es. al posto di "www.google.com" viene messo il tag URL).
- Ogni tag è stato sostituito da un indicatore, che indica quando inizia e quando finisce la frase rinchiusa all'interno di quel hashtag.

La struttura del modello, come già accennato, era composta da 12 encoder in sequenza.

4.1.1 Fine tuning

Per il fine tuning è stato utilizzato un batch di 512 esempi, con un learning rate pari a 2e-5 e con 1000 step per ogni ciclo. Sono stati utilizzati gli stessi parametri

per tutti e 4 i task. Per la soggettività, positività e negatività sono stati utilizzate 3 epoche, mentre invece per l'ironia le epoche sono state 10.

Questo modello ha portato un miglioramento considerevole per tutti e 4 i task, come possiamo vedere in Tabella 4.10, sopratutto nel riconoscimento dell'ironia, uno dei task più ostici. Non è stato utilizzato alcun set di validazione perché non c'è stato bisogno di alcuna scelta degli iper-parametri.

Per quanto riguarda la polarità, anche se negatività e positività sono inserite nella stessa tabella vengono classificati con due modelli differenti, ovvero un modello che verifica se un commento è positivo o meno, e un modello che verifica se è negativo o meno. L'utilizzo dei due diversi modelli è legato al fatto che alcuni elementi del dataset sono definiti neutri, ovvero ne positivi ne negativi. Se si fosse scelto un unico classificatore sarebbe stato necessario scartare tutti quelli neutri e avere uno sbilanciamento in termini di esempi rispetto agli altri due task.

4.2 Alberto e transfer learning della classificazione dei repertori

La prima tecnica implementata per poter unire l'informazione portata dalla classificazione dei repertori e il modello Alberto è il transfert learning, ovvero utilizzare come modello pre allenato il modello Alberto, definito per la classificazione dei repertori. In questo modo si spera che le informazioni che il modello assimila dalla classificazione dei repertori possano essere utili nei 4 task considerati.

Questa procedura risulta facilmente implementabile, anche se nel primo caso si parla di una classificazione a 23 classi (repertori) e nel secondo di una classificazione a 2 classi (per ognuno dei singoli task EVALITA2016) cambiando l'ultimo livello denso, come è rappresentato in Figura 4.1. Nella fase di fine tuning, il modello avrà dei cambiamenti importanti dal punto di vista dei pesi solo nell'ultimo livello denso. Per quanto riguarda i parametri presenti nei 12 encoder, le modifiche saranno minime, visto che buona parte della comprensione della lingua è già interiorizzata dal modello.

Soggettività	F1 Oggettività	F1 Soggettività	Macro F1
AlBERTo	**0.7398**	**0.8415**	**0.7906**
Unitor.1.u	0.6784	0.8105	0.7444
Unitor.2.u	0.6723	0.7979	0.7351
samskara.1.c	0.6555	0.7814	0.7184
ItaliaNLP.2.c	0.6733	0.7535	0.7134
Ironia	F1 Non ironia	F1 Ironia	Macro F1
AlBERTo	**0,9408**	**0,2772**	**0,6090**
tweet2check16.c	0,9115	0,1710	0,5412
CoMoDI.c	0.8993	0,1509	0,5251
tweet2check14.c	0.9166	0,1159	0,5162
IRADABE.2.c	0.9241	0.1026	0,5133
BERT Multilang	0.9376	0.000	0,4688
Polarità	F1 Positività	F1 Negatività	Macro F1
AlBERTo	**0.7155**	**0.7291**	**0.7223**
Unitor.1.u	0.6850	0.6426	0.6638
Unitor.2.u	0.6354	0.6885	0.6620
samskara.1.c	0.6312	0.6838	0.6575
ItaliaNLP.2.c	0.6265	0.6743	0.6504

Tabella 4.1: Prestazioni Alberto nella competizione EVALITA2016.

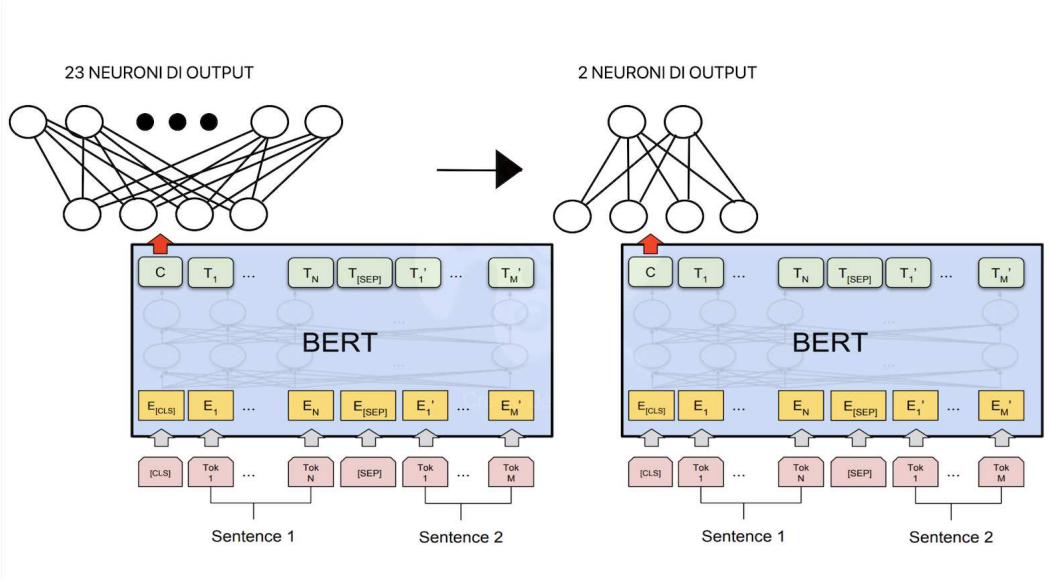

Figura 4.1: Transfer learning con BERT.

4.2.1 Risultati

Una caratteristica dei risultati da parte dei modelli BERT è la loro varianza [11]. Stesso modello, con gli stessi parametri, ma con pesi iniziali differenti possono produrre risultati estremamente diversi tra loro; alcuni modelli performeranno estremamente bene, mentre altri invece produrranno risultati più discreti. Questo rende difficile comparare due tecniche diverse. Nell'articolo precedentemente illustrato non ne viene fatta menzione, non specificando se i risultati prodotti rappresentano il miglior modello, un modello medio o altro.

Per questo motivo si è deciso prima di tutto di riprodurre ogni test 15 volte, ovvero allenando 15 volte diverse il modello e verificando ogni volta le sue prestazioni, andando a considerare per ogni approccio il valore medio, la deviazione standard e il valore del miglior modello.

In questo modo è sicuramente possibile confrontare correttamente i due approcci senza il rischio di far riferimento ad un caso particolarmente fortuito da una parte o dall'altra. Applicando questa metodologia di confronto, si devono però anche riprodurre i risultati riportati nell'articolo in cui viene presentato il modello Alberto per poter ottenere il risultato massimo, la deviazione standard e la media.

La scelta degli iper-parametri è stata fatta ovviamente utilizzando un set di validazione, andando a scegliere il modello con la loss inferiore. É stata scelta la loss come metro di giudizio per la valutazione dell'errore perché un modello con una loss bassa nel set di validazione produceva ottimi risultati nel set di test.

Gli iper-parametri sono risultati gli stessi per tutti i 4 task, eccezion fatta per la positività che non aveva nessun peso per la loss.

Ironia

Il primo task di cui si andrà a verificare i risultati è l'ironia, come si può vedere in Tabella 4.3. Per quanto riguarda la classe che rappresenta la sua assenza, i risultati

Iperparametri modello	Learning rate	Batch size	Numero di epoche	Peso loss
Neg Iro Subj	1e-5	100	2	[0:100,1:1000]
Pos	1e-5	100	2	None

Tabella 4.2: Iperparametri modello BERT.

rimangono praticamente inalterati; mentre invece le prestazioni riguardanti la sua presenza crescono in maniera importante, influendo quindi sul valore medio. Nella stessa tabella si può vedere che il valore massimo migliore risulta essere il modello che utilizza il pre-allenamento con i repertori, come dal punto di vista della deviazione standard, anche se non in maniera determinante come le precedenti.

Ironia	Alberto senza repertori	Alberto con repertori
F1 non ironia media	**0.9278**	0.9249
F1 ironia media	0.3314	**0.3570**
F1 macro media	0.6296	**0.6410**
F1 macro massimo	0.6644	**0.6704**
F1 deviazione standard	0.0271	**0.0186**

Tabella 4.3: Risultati predizione modello con e senza repertori ironia.

I risultati sono confermati anche dal boxplot presente in Figura 4.2, dove si può vedere come varia il valore mediano delle due tipologie di modelli. Un'altra informazione fondamentale è la presenza di due valori anomali per il modello che usa i repertori; questo indica che il valore della deviazione standard, come quello della media, ne sia fortemente influenzato, e quindi 13 dei 15 modelli sono compresi in un intervallo estremamente inferiore rispetto a quello mostratoci dalla deviazione standard.

Il punto indicato in legenda come "stato dell'arte" è invece il valore riportato nell'articolo in cui viene presentato Alberto; in questo caso è perfettamente plausibile visto che rientra tra l'intervallo dei risultati prodotti, e, secondo quanto affermato nell'articolo, il risultato è frutto di un'unica predizione.

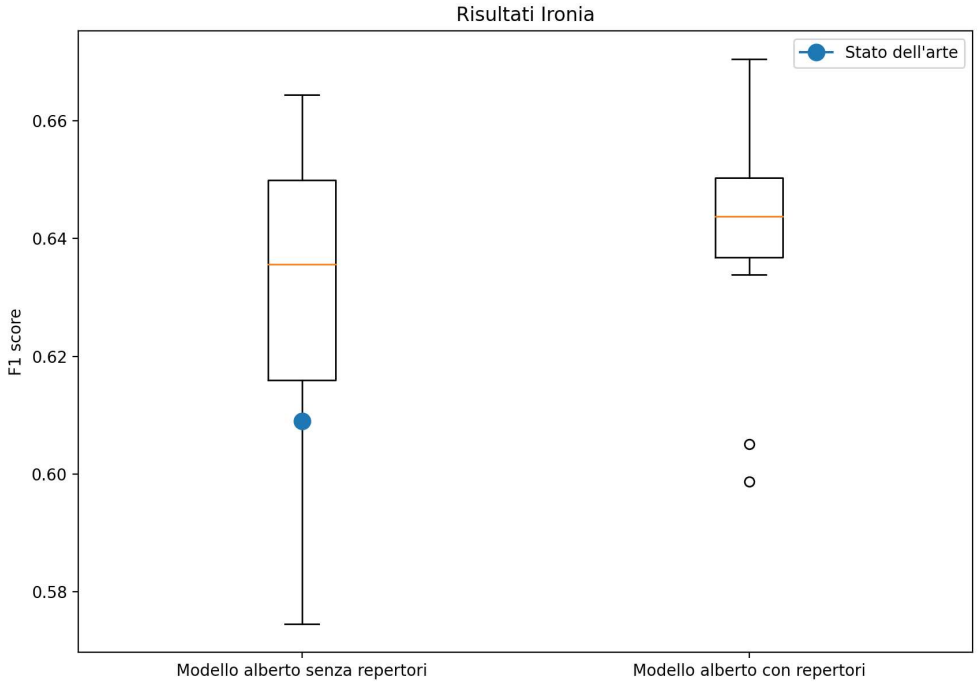

Figura 4.2: Boxplot predizione modello con e senza repertori ironia.

Si può vedere dalla Figura 4.3 che, dopo la seconda epoca, la loss del dataset di validazione tende a crescere e divergere dalla loss del dataset di allenamento. Un altro elemento estremamente interessante è il fatto che il modello pre-allenato nella classificazione dei repertori parte da una loss leggermente inferiore, e questo gap viene colmato con il crescere delle epoche. Questo potrebbe indicare che utilizzare la tecnica del transfert learning indirizzi nella direzione giusta l'apprendimento della classificazione dell'ironia.

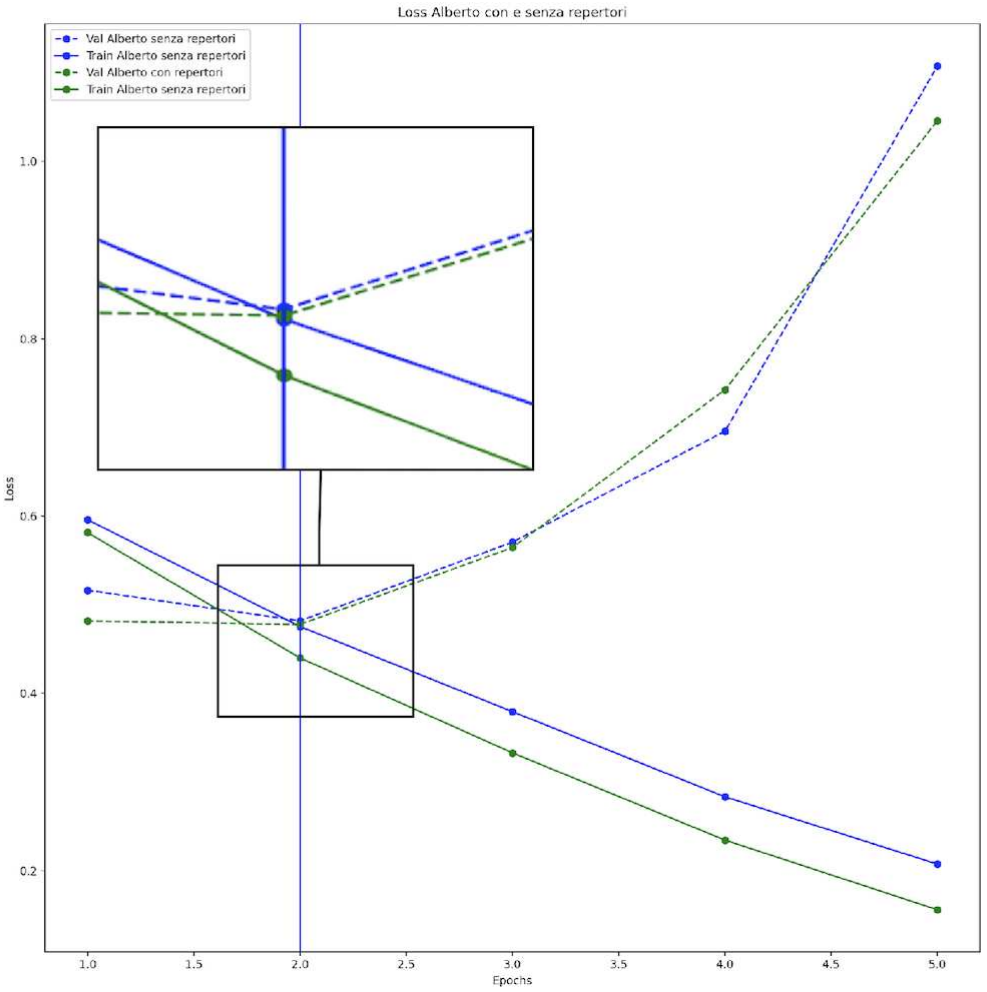

Figura 4.3: Loss set di test e di validazione ironia

Soggettività

Come nel caso dell'ironia, anche nel task della soggettività la classe di minoranza è quella che risulta più avvantaggiata dall'utilizzo del metodo del pre-allenamento con la classificazione dei repertori, mentre la classe di maggioranza rimane pressoché invariata. Dal punto di vista sia dei risultati medi, che della deviazione standard, il modello con i repertori performa meglio. Dal punto di vista invece del modello che produce il valore massimo, l'approccio classico dà risultati migliori. Tutto questo in Tabella 4.4

Soggettività	Alberto senza repertori	Alberto con repertori
F1 oggettività media	0.6185	**0.6649**
F1 soggettività media	**0.8518**	0.8493
F1 macro media	0.7352	**0.7571**
F1 macro massimo	0.7879	**0.7741**
F1 deviazione standard	0.0327	**0.0137**

Tabella 4.4: Risultati Alberto con e senza repertori soggettività.

Dal boxplot in Figura 4.15 è possibile notare come il valore massimo prodotto dal modello è leggermente inferiore rispetto allo stato dell'arte. Questo è comunque considerato fattibile se ipotizziamo che il modello dell'articolo risulti un caso particolarmente buono, cosa non così improbabile visto che con la riproduzione di 15 istanze il modello ha ottenuto risultati simili. In aggiunta ci mostra come nel caso del modello che utilizza i repertori i risultati tendono ad essere maggiormente concentrati -ovvero con una mediana in una posizione più alta- , mentre invece, per il modello che non li usa, la deviazione standard è sicuramente maggiore, permettendo infatti di ottenere il modello con le prestazioni migliori ma parallelamente molti modelli con prestazioni inferiori.

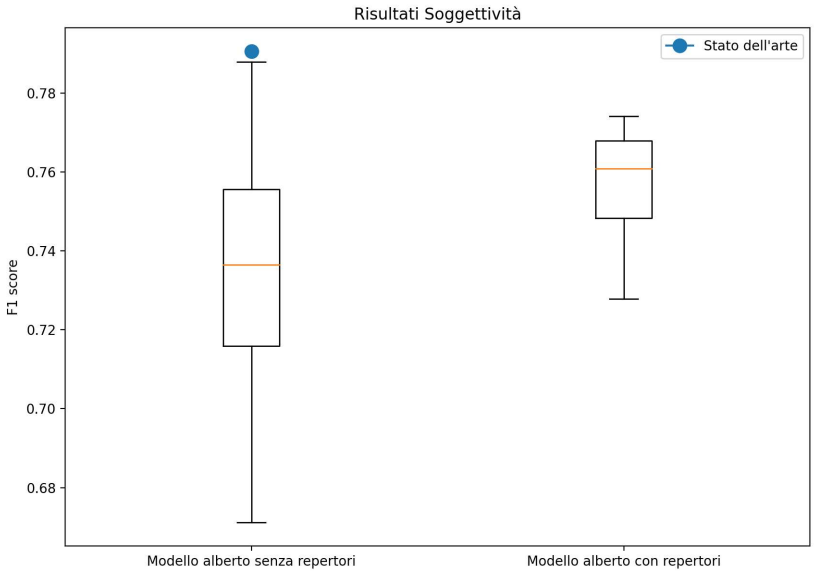

Figura 4.4: Boxplot predizione modello con e senza repertori soggettività.

Dal plot della loss in Figura 4.5 si può notare che, a differenza dell'ironia, la

funzione di apprendimento della soggettività nel set di validazione tende a divergere maggiormente dalla funzione di allenamento del set di validazione.

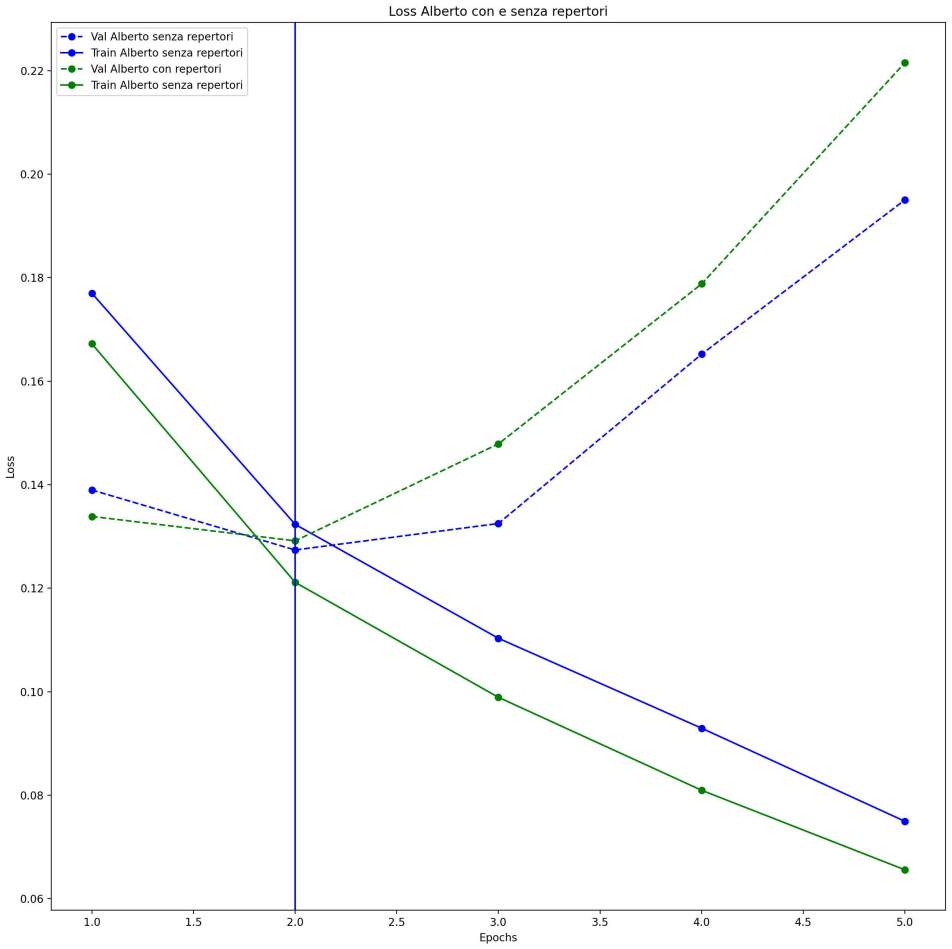

Figura 4.5: Loss set di test e di validazione soggettività.

Positività

Nel caso della positività, in Tabella 4.5, non si possono riscontrare risultati significativi, né per la classe positività né per la classe neutralità. Il valore massimo è prodotto dal modello che non utilizza i repertori, mentre l'approccio con i repertori riduce la deviazione standard. Gli stessi risultati sono visibili in Figura 4.6, dove si può notare che i due boxplot hanno posizioni simili, anche se la mediana risulta leggermente più alta per il modello con i repertori e il modello senza repertori ha una maggiore varianza nei risultati. La maggiore varianza anche in questo caso comporta la capacità di poter raggiungere il modello massimo migliore.

In questo caso possiamo dire che i repertori hanno semplicemente una funzione di regolarizzazione senza alcun miglioramento importante di altri risultati.

Il risultato riportato nell'articolo già citato è perfettamente plausibile rispetto alla serie dei valori riprodotti.

Positività	Alberto senza repertori	Alberto con repertori
F1 non positivo media	0.9029	**0.9074**
F1 positivo media	0.5463	**0.5495**
F1 macro media	0.7246	**0.7284**
F1 macro massimo	**0.7426**	0.7361
F1 deviazione standard	0.0081	**0.0041**

Tabella 4.5: Risultati Alberto con e senza repertori positività.

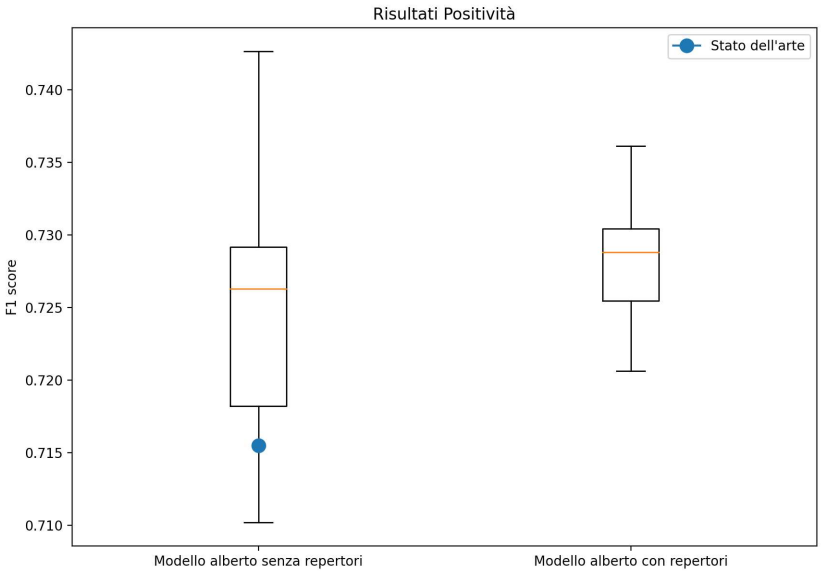

Figura 4.6: Boxplot predizione modello con e senza repertori positività.

In Figura 4.7 si può notare come nei casi precedenti il modello allenato con i repertori parta inizialmente con una loss minore, che alla seconda epoca tende a convergere allo stesso valore. Il fatto che i due modelli convergano allo stesso punto in fase di validazione della loss potrebbe spiegare perché non c'è un grande miglioramento in termini di valore medio.

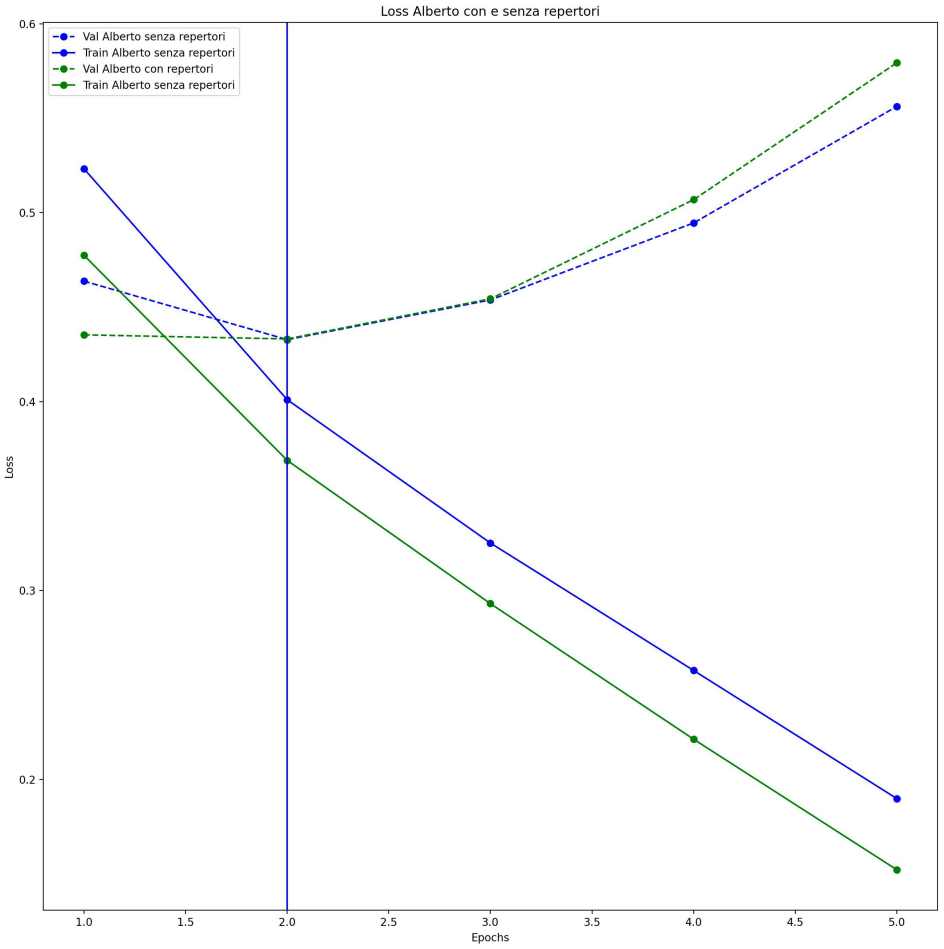

Figura 4.7: Loss set di test e di validazione positività.

Negatività

In maniera similare al caso del task della positività, si può mostrare che l'approccio adottato non mostra particolari miglioramenti nemmeno per la negatività. L'unico parametro per la quale il secondo modello risulta migliore è la deviazione standard, anche se non per un valore significativo.

Questa somiglianza è visibile anche nel boxplot in Figura 4.8, dove sono estremamente simili le due rappresentazioni.

Per quanto riguarda lo stato dell'arte, i valori riprodotti non sono coerenti con i valori segnati nell'articolo, visto che il nostro modello in generale produce risultati nettamente migliori.

Negatività	Alberto senza repertori	Alberto con repertori
F1 non negativo media	**0.8127**	0.8079
F1 negativo media	**0.7474**	0.7438
F1 macro media	**0.7801**	0.7759
F1 macro massimo	**0.7894**	0.7875
F1 deviazione standard	0.0101	**0.0078**

Tabella 4.6: Risultati Alberto con e senza repertori negatività.

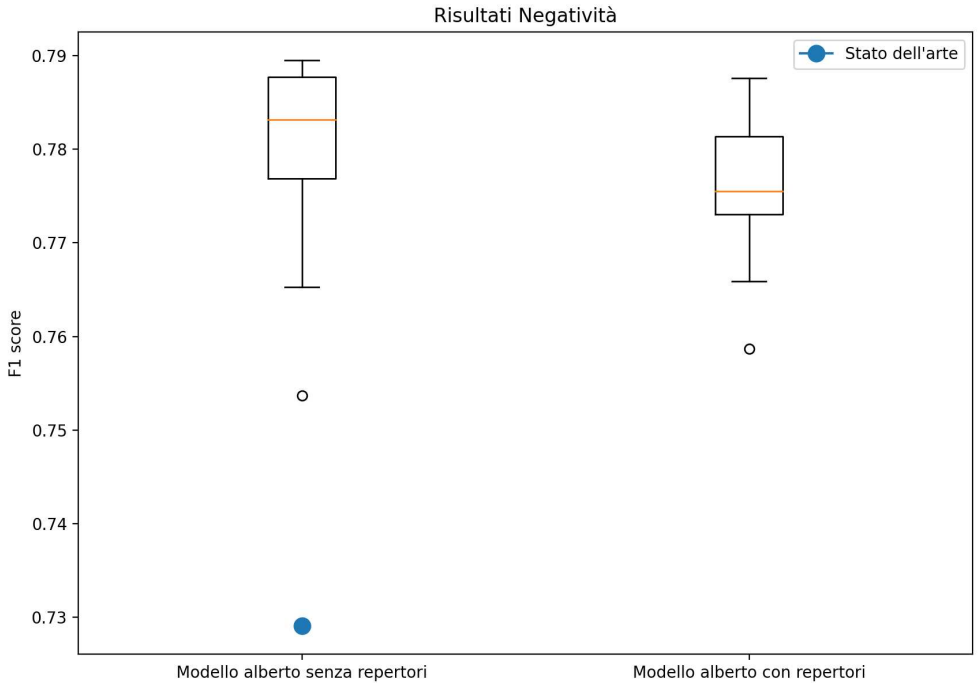

Figura 4.8: Boxplot predizione modello con e senza repertori negatività.

La funzione della loss in Figura 4.9 non ci fornisce particolari informazioni, a parte la tendenza condivisa anche dagli altri task di partire da un valore più basso e mantenere questo intervallo per tutto il processo di allenamento.

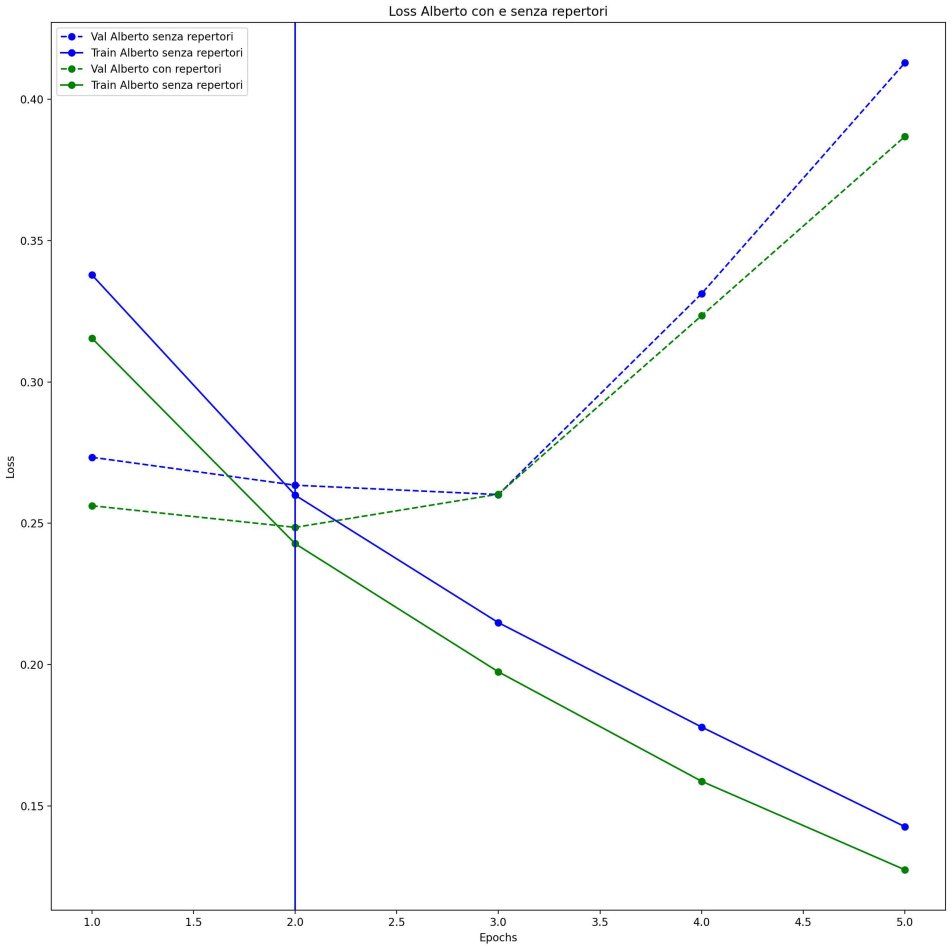

Figura 4.9: Loss set di test e di validazione negatività.

4.2.2 Prime conclusioni

Si può sicuramente notare come la soggettività e l'ironia subiscano un innegabile miglioramento generale delle prestazioni, a differenza della polarizzazione. Questo può dipendere da diversi fattori: ad esempio, può succedere che la soggettività e l'ironia siano fortemente correlate e quindi dipendere da caratteristiche intrinseche del task stesso.

4.3 Esplorazione di altri metodi di integrazione

Visto il successo di questo approccio per l'ironia e la soggettività, si è deciso di proseguire ed esplorare nuovi metodi per integrare l'informazione relativa ai repertori

parallelamente all'utilizzo del modello BERT. La prima soluzione sperimentata è stata quella di aggiungere l'informazione codificando il testo sotto forma di repertorio con il formato One Hot Encoder, un formato utilizzato per rappresentare variabili categoriche in formato numerico. Viene creato un vettore con tante celle quante sono le classi presenti. Si assegna ad ogni cella il valore 1/k (k che rappresenta il numero di classi per quel esempio) alla o alle celle corrispondente alla o alle classi presenti.

Una volta codificato in questo modo il testo viene concatenato al token [CLS] che passerà successivamente per il livello denso come mostrato in Figura 4.10.

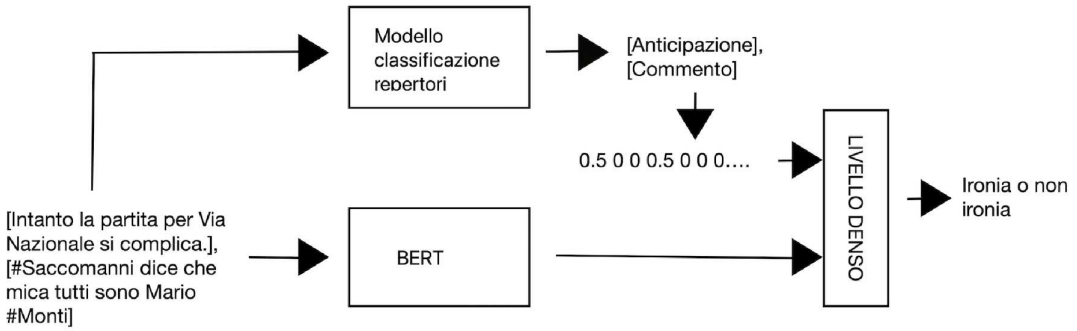

Figura 4.10: One Hot Encoder con livello denso.

Questo primo approccio non ha restituito dei buoni risultati, probabilmente per due ragioni principali: la prima, perché andando ad inserire la codifica prima del livello denso si rischia che il modello basi la classificazione sopratutto sulla codifica, questo lo renderebbe estremamente simile o addirittura inferiore all'approccio utilizzato con la SVM, che ha già mostrato i suoi limiti.

La seconda è legata al fatto che questa informazione non passerà per il meccanismo di attenzione del modello, che permetterebbe un aumento importante delle informazioni andando ad oscurare la parte non informativa.

La soluzione, per evitare i limiti elencati, è quella di inserire l'informazione fornita dalla codifica del testo all'interno del modello BERT, ovvero allegare esplicitamente l'informazione dei repertori, e non in maniera indiretta tramite il transfert learning.

L'idea è quella di concatenare l'informazione codificata al testo prima dell'input nel modello. Sono state prese in considerazione due possibili codifiche:

- One Hot Encoder: dopo aver classificato un testo utilizzando il modello BERT che rileva i repertori, trasforma le variabili categoriche nel formato One Hot Encoder.

- Ultimo livello denso dopo la sigmoide: viene usato il risultato del modello BERT nell'ultimo livello dopo aver applicato la sigmoide. Questa tecnica è implementata per aumentare la capacità informativa rispetto a quella del formato One Hot Encoder.

E' stato deciso di implementare questa tecnica sia per il modello BERT originale sia per il modello pre-allenato sui repertori, tutto questo per ognuno dei 4 task. Come nel caso dei modelli precedenti, si è verificato quale fosse il miglior modello con il set di validazione, e i miglior iper-parametri sono risultati gli stessi del modello precedente. La pipeline è visibile in Figura 4.11.

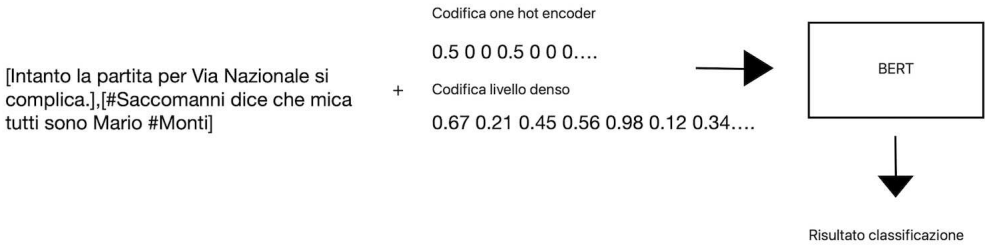

Figura 4.11: Pipeline con modello che usa la codifica del testo.

4.3.1 Risultati

Ironia

Per l'ironia questo approccio risulta estremamente soddisfacente: come nelle modalità precedenti, ha fornito un valore sostanzialmente invariato per quanto riguarda la classe di maggioranza, riscontrando invece un notevole miglioramento nella classe di minoranza.

L'utilizzo della tecnica di aggiunta al testo della sua versione codificata porta miglioramenti graduali, secondo la complessità e la quantità di informazione presente nella codifica stessa; questo avviene però solo per il modello pre-allenato sui repertori, come mostra la Figura 4.12.

Per quanto riguarda il modello che alla base usa semplicemente Alberto, non si possono notare particolari miglioramenti. Si nota anzi un leggero peggioramento per tutte le tipologie di parametri. Per quanto riguarda la deviazione standard, essa rimane inalterata. Dall'immagine è possibile però apprezzare che questo, nel caso dei modelli che usano il pre-allenamento con i repertori, è causato da 3 valori anomali che aumentano fortemente la deviazione standard. Se osserva bene, buona parte dei valori sono racchiusi in un area inferiore a quella occupata nel caso dei modelli che non usano i repertori.

Si può inoltre notare, come riportato nella Tabella 4.7, che con i repertori si ottengono i migliori risultati in termini di modello con le massime prestazioni, e che la bontà del modello tende a crescere in base alla codifica fornita.

Ironia	Alberto	Alberto One Hot	Alberto Denso
F1 non ironia media	**0.9278**	0.9240	0.9240
F1 ironia media	0.3314	0.3164	0.3108
F1 macro media	0.6296	0.6202	0.6174
F1 macro massimo	0.6644	0.6564	0.6431
F1 deviazione standard	0.0271	0.0264	0.0272

Ironia	Repertori	Repertori One Hot	Repertori Denso
F1 non ironia media	0.9249	0.9230	0.9190
F1 ironia media	0.3570	0.3655	**0.3853**
F1 macro media	0.6410	0.6443	**0.6521**
F1 macro massimo	0.6704	0.6730	**0.6855**
F1 deviazione standard	**0.0186**	0.0285	0.0286

Tabella 4.7: Risultati Alberto con e senza repertori con codifica ironia.

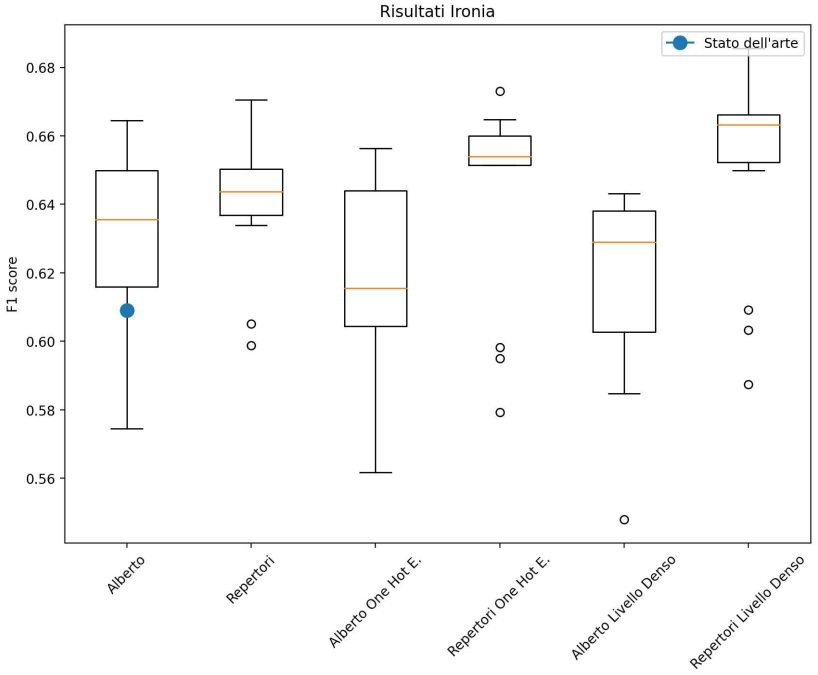

Figura 4.12: Boxplot predizione modello con e senza repertori ironia.

Soggettività

Nel caso della soggettività si può notare in Tabella 4.8 che la deviazione standard restituisce un valore che rappresenta meglio la realtà, grazie al numero inferiore di valori anomali presenti. I risultati in questo caso mostrano che l'aggiunta della codifica all'informazione testuale non porta un grande miglioramento: restano inalterati i risultati del modello pre-allenato sui repertori, e peggiorano quelli del modello che utilizza l'approccio dello stato dell'arte (i valori della media e della deviazione standard peggiorano considerevolmente). L'unica miglioria considerevole che si può notare è l'incremento del valore del modello massimo per il caso del modello Alberto con la codifica del livello denso. Questa cosa è possibile vederla anche nel boxplot in Figura 4.13.

Si può notare un leggero miglioramento per quanto riguarda la media ma non abbastanza per non considerarlo una piccola variazione data dalla casualità.

Soggettività	Alberto	Alberto One Hot	Alberto Denso
F1 oggettività media	0.6185	0.5685	0.5532
F1 soggettività media	**0.8518**	0.8457	0.8461
F1 macro media	0.7352	0.7071	0.6997
F1 macro massimo	0.7879	0.7917	**0.8009**
F1 deviazione standard	0.0327	0.0517	0.0714
Soggettività	Repertori	Repertori One Hot	Repertori Denso
F1 oggettività media	**0.6649**	0.6615	0.6635
F1 soggettività media	0.8493	0.8487	0.8511
F1 macro media	0.7571	0.7551	**0.7573**
F1 macro massimo	0.7741	0.7934	0.7762
F1 deviazione standard	**0.0137**	0.0183	0.0220

Tabella 4.8: Risultati Alberto con e senza repertori con codifica soggettività

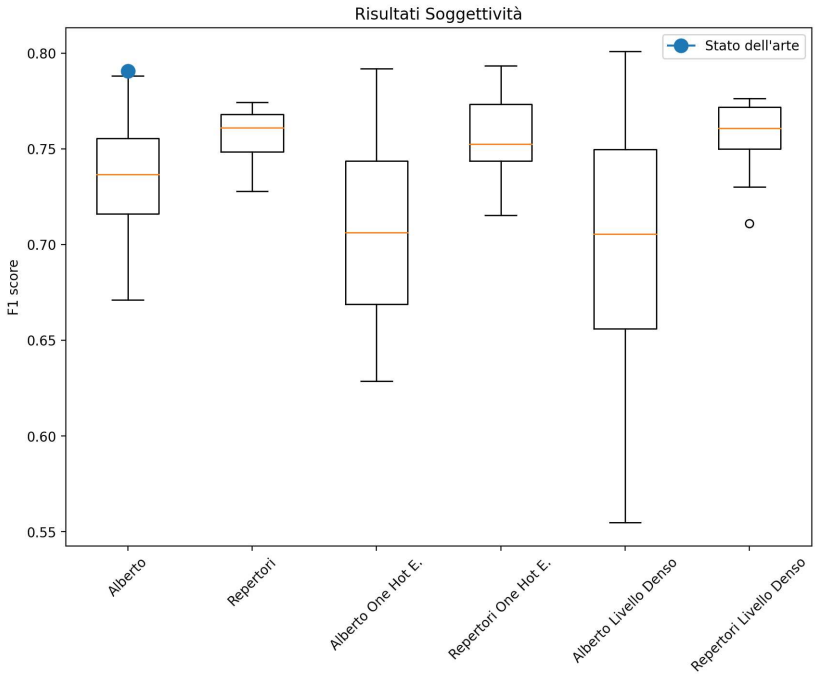

Figura 4.13: Boxplot predizione modello con e senza repertori soggettività.

Positività

Nel caso della positività, generalmente non si riscontrano grandi differenze: i modelli che si basano sul transfert learning nella classificazione dei repertori tendono ad avere una deviazione standard più bassa. É interessante vedere come tutti gli approcci che utilizzano la concatenazione della codifica incrementano in maniera importante la deviazione standard, in maniera minore il modello che non usa il transfer learning ed in maniera maggiore quello che lo fa.

Questo influisce fortemente anche sul valore massimo, che tendenzialmente apparterrà al modello con la varianza più ampia. In questo caso, infatti, l'approccio migliore per ottenere il miglior modello è Alberto classico con l'aggiunta della codifica One Hot Encoder. Tutto questo in Tabella 4.9.

In questo caso, diversamente da quanto accade in relazione a ironia e soggettività, non c'è nessuna differenza, né per la classe di minoranza, che corrisponde alla classe con prestazioni peggiori, né per quella di maggioranza. Questo avviene per tutte le tipologie di approcci.

Dal boxplot in Figura 4.14 è possibile notare che i risultati confermano quanto definito dalla deviazione, dato che tendono a raggrupparsi maggiormente nel caso dei modelli che utilizzano i repertori discorsivi.

Negatività

Dai risultati della negatività emerge chiaramente che gli approcci basati sulla scienza dialogica non portano miglioramenti. Questi valori sono perfettamente confermati dal boxplot in Figura 4.15.

Positività	Alberto	Alberto One Hot	Alberto Denso
F1 non positivo media	0.9029	0.8955	0.8995
F1 positivo media	0.5463	0.5546	**0.5564**
F1 macro media	0.7246	0.7251	0.7280
F1 macro massimo	0.7426	**0.7505**	0.7442
F1 deviazione standard	0.0081	0.0173	0.0117
Positività	Repertori	Repertori One Hot	Repertori Denso
F1 non positivo media	**0.9074**	0.9064	0.9043
F1 positivo media	0.5495	0.5485	0.5423
F1 macro media	**0.7284**	0.7274	0.7233
F1 macro massimo	0.7361	0.7379	0.7317
F1 deviazione standard	**0.0041**	0.0058	0.0061

Tabella 4.9: Risultati Alberto con e senza repertori con codifica positività.

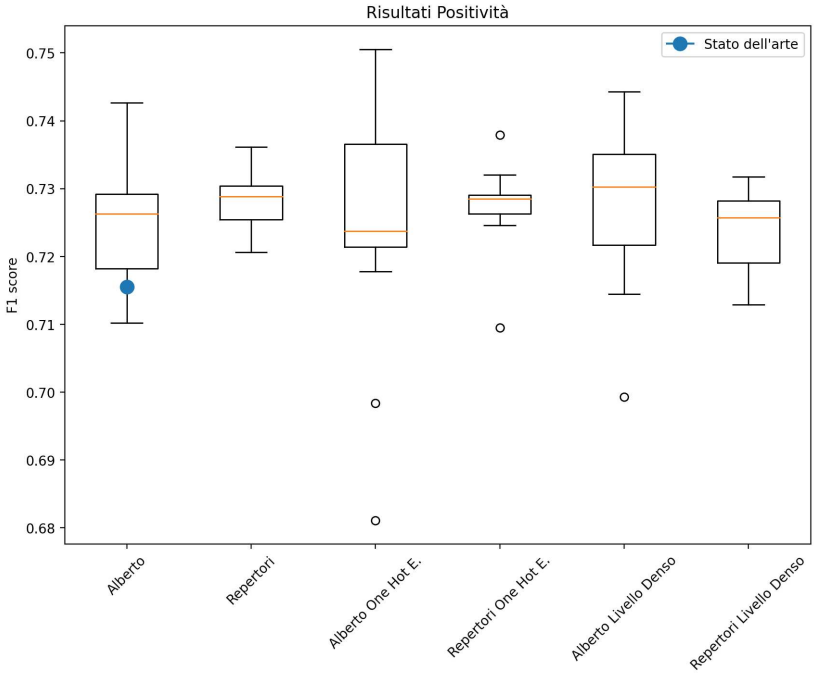

Figura 4.14: Boxplot predizione modello con e senza repertori positività

Si può anche notare che l'approccio che non usa il transfer learning ma implementa la codifica One Hot Encoder, restituisce il modello con il valore massimo migliore. Ciò è sicuramente dato dal fatto che tale approccio ha deviazione standard maggiore e di conseguenza presenta un numero maggiore di modelli peggiori. La deviazione standard minore è ottenuta con l'aggiunta della codifica più complessa, sia nel modello base, sia nel modello a cui è applicato il transfer learning.

Nel caso del valore medio, il modello del transfer learning con l'aggiunta delle codifiche rimane inalterato, mentre l'utilizzo dell'approccio classico comporta una leggera decrescita. Il calo della media è dovuto alla diminuzione delle predizioni corrette della classe con le prestazioni migliori, ovvero la non negatività. Mentre la classe con le prestazioni peggiori, ovvero la negatività, non subisce alcuna modifica.

Negatività	Alberto	Alberto One Hot	Alberto Livello Denso
F1 non negativo media	**0.8127**	0.8075	0.8124
F1 negativo media	**0.7474**	0.7457	0.7405
F1 macro media	**0.7801**	0.7766	0.7765
F1 macro massimo	0.7894	**0.7954**	0.7865
F1 deviazione standard	0.0101	0.0123	**0.0075**
Negatività	Repertori	Repertori One Hot	Repertori Livello Denso
F1 non negativo media	0.8079	0.8007	0.8023
F1 negativo media	0.7438	0.7422	0.7420
F1 macro media	0.7759	0.7715	0.7721
F1 macro massimo	0.7875	0.7825	0.7808
F1 deviazione standard	0.0078	0.0102	**0.0075**

Tabella 4.10: Risultati Alberto con e senza repertori con codifica negatività.

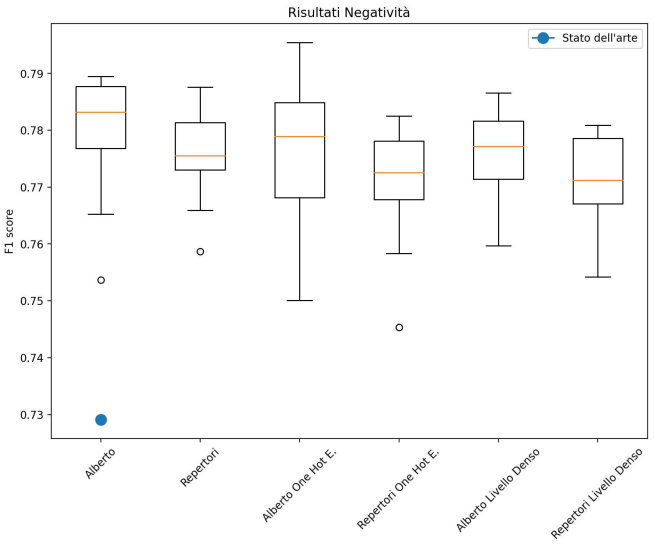

Figura 4.15: Boxplot modello con e senza repertori negatività codifica.

Conclusioni

É interessante notare come l'utilizzo della codifica, nel caso dell'ironia, porti a migliori risultati solo se utilizzata in combinazione con il transfer learning. Ovvero, per avere prestazioni più alte è necessario combinare i due approcci; il solo utilizzo della codifica senza il transfer learning comporta il peggioramento dei risultati rispetto allo stato dell'arte.

Per la soggettività invece non si possono riscontrare particolari miglioramenti tramite l'utilizzo della codifica insieme al transfer learning. Nel caso in cui la si utilizzi da sola, possiamo notare nel modello un importante aumento della deviazione standard.

É chiaro, quindi, come l'utilizzo della codifica porti miglioramenti unicamente per quanto riguarda la classificazione dell'ironia. Invece, con l'utilizzo dei repertori, si ha in generale un miglioramento per quanto riguarda la classificazione dell'ironia e della soggettività, mentre per la positività e negatività non si nota alcun miglioramento.

Questo può dipendere da molti fattori. Un'ipotesi potrebbe essere che ironia e soggettività sono qualcosa di concettualmente molto più complesso rispetto alla semplice positività e negatività. Questo indicherebbe che i repertori sono efficaci solamente nel caso di task che richiedono un certo grado di complessità.

L'idea per il prossimo capitolo è quella di esplorare altre tipologie di task per esaminare come si comportano i modelli che utilizzano le informazioni generate dai repertori discorsivi. Per fare ciò, è stato scelto un task di classificazione di tweet con la presenza di stereotipi e hate speech. É fondamentale provare a mantenere la stessa tipologia di testo, ovvero quello prodotto con i social network, per evitare che questo possa influenzare la classificazione.

Capitolo 5

Classificazione dei repertori per hate speech e stereotipi

Come già introdotto nel capitolo precedente, la scelta è stata quella di continuare ad esplorare il contributo che i repertori possono dare nella classificazione del testo.

Il motivo per cui la scelta è ricaduta su questi due specifici task è legata al fatto che fosse disponibile un dataset composto unicamente da tweet, la stessa tipologia di testo sulla quale sono stati allenati gli altri modelli. Questo perché in qualche modo l'ironia e la soggettività possono essere correlate a questi due task.

L'ironia è largamente utilizzata nei social network ed è uno degli elementi maggiormente persuasivi. Comporta l'utilizzo di parole opposte a quello che si intende. Solitamente è utilizzata per esprimere sentimenti negativi riguardanti la sfera privata, però spesso può essere uno strumento per coprire commenti offensivi, odio e stereotipi [13].

É innegabile come sia però necessario riuscire a distinguere tra uso di ironia, hate speech e stereotipi. Per questo motivo un fattore determinante nella classificazione efficace di hate speech e stereotipi potrebbe essere quello di sfruttare lo stesso meccanismo che il modello utilizza per riconoscere l'ironia. L'ironia potrebbe essere quello che differenza un commento di odio da uno che non lo è, visto che con l'uso dell'ironia si intende il significato opposto.

5.1 Distribuzione repertori e albero di decisione

Come nel caso della classificazione della soggettività, ironia, positività e negatività si è deciso di utilizzare un modello intermedio prima di implementare direttamente il modello BERT.

In maniera analoga a quanto fatto precedentemente, è stato considerato il modello Alberto allenato a classificare i repertori e nella divisione del testo. Sono stati utilizzati questi due modelli per trasformare il testo in una codifica secondo i repertori discorsivi.

La classificazione degli stralci di testo in repertori permette di analizzarne la distribuzione, ovvero vedere per ogni tipologia di testo come si differenzia sulla base della composizione dei repertori. In Figura 5.1 possiamo notare questa distribuzione: è facilmente evidenziabile che i testi con stereotipi ed hate speech tendono ad avere

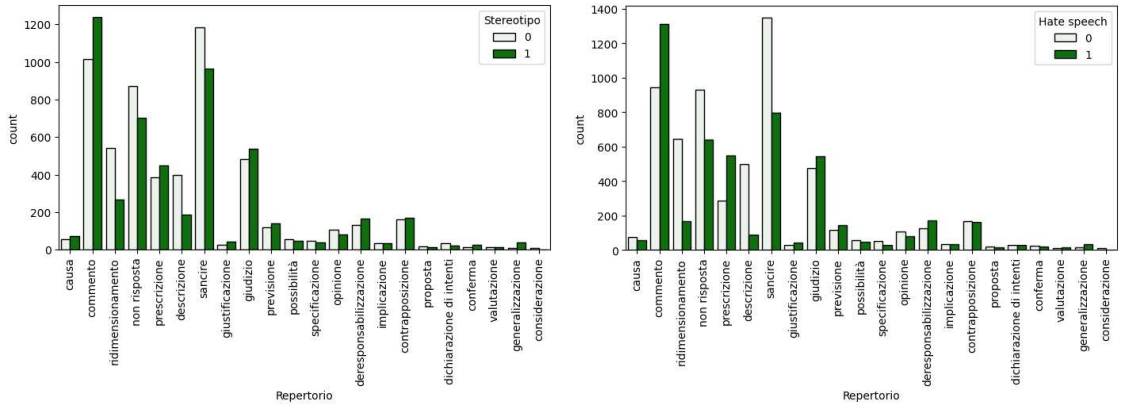

Figura 5.1: (a) Distribuzione repertori secondo stereotipi (b) Distribuzione repertori secondo hate speech

maggiormente il repertorio del commento, mentre tende ad essere meno presente il repertorio sancire, ridimensionamento e descrizione. Per il resto dei repertori non si possono definire altri pattern particolari.

Per ogni testo sono state implementate due possibili codifiche:

- Assoluta: simile al formato One Hot Encoder, ma con la somma del vettore che corrisponde al numero di classi rappresentate. Ovvero: se il testo contiene due classi, nella cella corrispondete a quelle due classi saranno presenti due 1.
- Relativa: corrisponde al formato One Hot Encoder

Successivamente, le stesse codifiche sono state utilizzate per classificare i vari esempi disponibili. Per fare ciò è stato usato un albero di decisione, che spesso ha prestazioni limitate rispetto ad altri modelli ma porta con se la possibilità di spiegare ogni decisione e comprendere quali sono i repertori più determinanti.

Capire i meccanismi utilizzati potrebbe essere fondamentale per avere un riscontro anche dal punto di vista teorico, e verificare se i parametri presi in considerazione per ogni decisione siano gli stessi tra macchina e umani.

Sono stati creati due alberi di decisione diversi per ognuno dei singoli task, ognuno dei quali definito da determinati iper parametri, visibili in Tabella 5.1.

Iperparametri	Criterion	Splitter	Max Depth	Min Samples Split	Class Weight
Stereotipi	entropy	best	766	508	Balanced
Hate speech	entropy	random	714	116	Balanced

Tabella 5.1: Iperparametri alberi di decisione.

I risultati ottenuti sono stati confrontati con i risultati ottenuti durante la competizione, visibili in Tabella 5.2. É stato testato il modello per le due tipologie di dataset di test, il primo contenente tweet e il secondo contenete articoli di giornale.

	Hs notizie F1	Hs tweet F1		St notizie F1	St tweet F1
1	0,774	0,808	1	0,720	0,771
2	0,731	0,789	2	0,718	0,767
3	0,725	0,789	3	0,716	0,761
4	0,718	0,780	4	0,685	0,741
5	0,702	0,778	5	0,681	0,738
6	0,698	0,776	6	0,670	0,707
7	0,692	0,775	7	0,646	0,688
8	0,682	0,771	8	0,641	0,667
9	0,675	0,768	9	0,605	**0,606**
10	.....	.....	10	0,538	0,603
11	.....	.....	11	**0,375**	0,507
...	.....	.....	12	0,367	0,467
...	.....	.....	13	0,307	0,336
	.....	.....			
22	.....	.....			
23	.....	0,6878			
24	0,50	**0,656**			
25	**0,44**	0,505			
26	0,38				

Tabella 5.2: Risultati competizione hate speech e stereotipi.

I risultati dell'albero di decisione sono quelli evidenziati: è possibile notare che, come ci si aspettava, con le notizie tende a restituire risultati pessimi in entrambi i task, mentre per quanto riguardo i tweet sembra fare meglio. Si deve considerare che il modello non utilizza direttamente alcuna tipologia di testo, che invece in un contesto come questo potrebbe essere estremamente informativo.

In Figura 5.2 è possibile vedere il contributo che ogni repertorio fornisce alla classificazione dell'hate speech. I repertori più determinanti sono sancire e descrivere. Il riscontro è anche individuabile in Figura 5.1, dove i citati repertori hanno presenza diversa a seconda se il testo è classificato come hate speech oppure no. Stessa cosa si può notare per i repertori di non risposta e ridimensionamento. Una eccezione la fa specificazione, che risulta molto determinante. Tuttavia non possiamo riscontrare

una forte differenza tra le due tipologie di testo. Queste informazioni sono coerenti anche dal punto di vista teorico: i repertori presenti sono quelli che ci si aspetta dalla natura del testo.

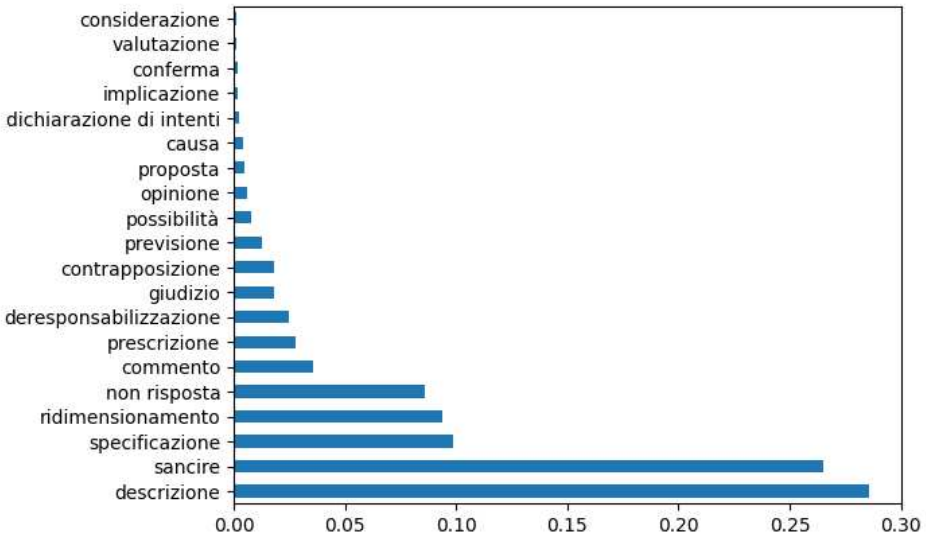

Figura 5.2: Importanza feature hate speech.

Parlando dei testi che hanno presenti stereotipi è facile notare come tutti repertori influiscono nella classificazione, a differenza dell'hate speech dove ne erano presenti alcuni praticamente irrilevanti.

É interessante prendere atto che i primi sei repertori in entrambe le situazione sono gli stessi, ma in diverso ordine. Questo ci può dare la misura della similarità dei due task e di come i repertori riescano a percepirla.

Fondamentale però è anche considerare che i repertori di commento, sancire e descrizione sono classi molto presenti nel dataset originale, quindi il modello tende a predirle facilmente; più specifici sono invece i repertori di non risposta, ridimensionamento.

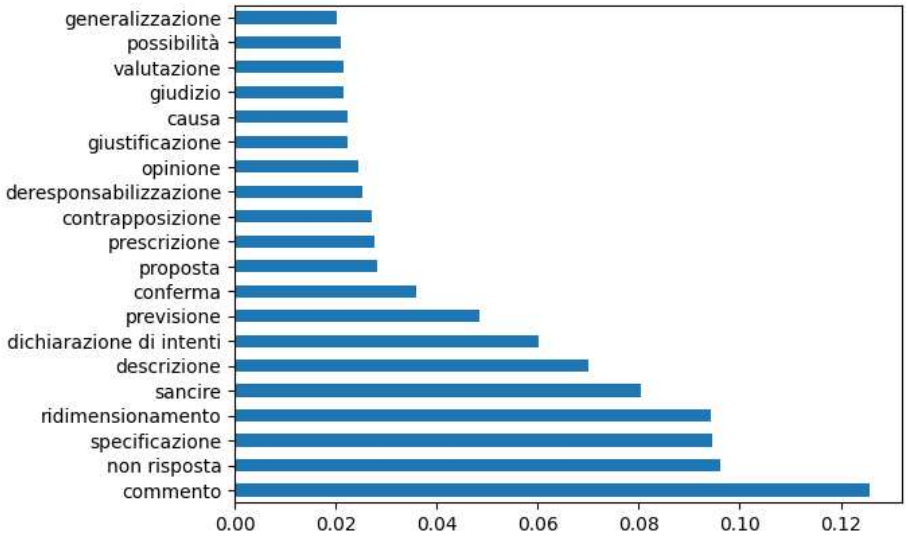

Figura 5.3: Importanza feature stereotipi.

5.2 Modello Alberto e repertori

Per la risoluzione di questi task è stato utilizzato lo stesso approccio utilizzato nel Capitolo 4, ovvero utilizzo del modello BERT con il pre allenamento definito con Alberto. Si è provato con il metodo del transfert learning a pre allenare il modello nella classificazione dei repertori, poi si è proseguito il test usando il metodo dell'aggiunta del testo codificato.

L'obiettivo non è tanto superare lo stato dell'arte, ma superare le prestazioni fornite dal modello pre-train Alberto. Di conseguenza si terrà in considerazione il risultato della competizione, mantenendo però come riferimento le prestazioni che Alberto ha per questo task. Questo approccio ha il fine di comprendere se la classificazione dei repertori fornisce informazioni che il modello BERT non riesce ad includere.

In questo caso il modello ha avuto bisogno di parametri differenti, con una batch size inferiore ed un learning rate più basso come possiamo vedere in Tabella 5.3. Gli iperparametri sono stati scelti andando a selezionare il modello migliore su un set di validazione, cercando i parametri del modello che produca la loss minore.

Iperparametri modello	Learning rate	Batch size	Epochs	Loss Weight
Stereotipi e hate speech	5e-6	32	2	None

Tabella 5.3: Iperparametri modello BERT.

5.2.1 Risultati

Hate speech

Nel caso del riconoscimento dell'hate speech si può sicuramente notare dai risultati in Tabella 5.4 che il modello Alberto con la codifica One Hot encoder preforma meglio di tutti gli altri, sotto tutti i punti di vista. La deviazione standard invece rimane invariata, eccezion fatta per il caso del modello Alberto con l'aggiunta della codifica del livello denso. Questo è spiegabile dal boxplot in Figura 5.4 che mostra la presenza di valori anomali che fanno risultare la deviazione maggiore, come si può vedere dai plot che sono tutti simili anche se situati ad altezze diverse.

Anche in questo caso la classe per cui abbiamo un miglioramento dei risultati è la classe di minoranza, mentre la classe di maggioranza rimane invariata. In questo caso il transfert learning non porta a particolari migliorie, e anzi tendono ad essere più efficienti i modelli che usano semplicemente il modello Alberto e la codifica.

Rispetto allo stato dell'arte, il modello migliore tende a risultare leggermente inferiore, ma si deve tenere a mente che questo approccio è estremamente semplice: il modello Alberto è stato sviluppato nel 2015 e solitamente può essere considerato il primo approccio provato. In queste tipologie di competizioni, invece, si tende ad applicare tecniche più complesse. Questo per dire che con l'utilizzo di tecniche più complesse e l'utilizzo della scienza dialogica si potrebbero raggiungere risultati più significativi.

Hate speech	Alberto	Alberto One Hot	Alberto Denso
F1 non hate media	0.8380	**0.8397**	0.8347
F1 hate media	0.5797	**0.6248**	0.5745
F1 macro media	0.7088	**0.7322**	0.7046
F1 macro massimo	0.7666	**0.7807**	0.7627
F1 deviazione standard	**0.0324**	**0.0324**	0.0610
Hate speech	Repertori	Repertori One Hot	Repertori Denso
F1 non hate speech media	0.8318	0.8300	0.8308
F1 hate speech media	0.5425	0.5443	0.5460
F1 macro media	0.6871	0.6871	0.6884
F1 macro massimo	0.7634	0.7551	0.7406
F1 deviazione standard	0.0352	0.0352	0.0353

Tabella 5.4: Risultati Alberto con e senza repertori con codifica hate speech.

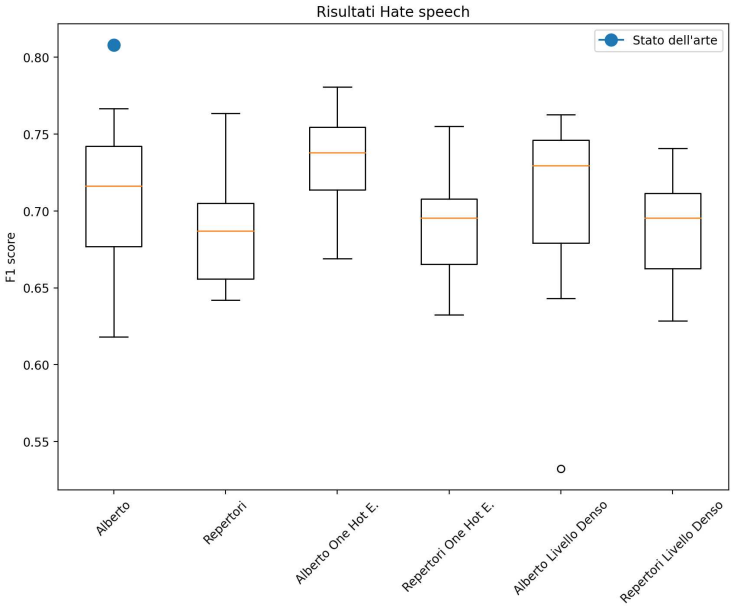

Figura 5.4: Boxplot modello con e senza repertori hate speech codifica

Stereotipi

I risultati prodotti dalla classificazione di testi con stereotipi sono all'interno della Tabella 5.5. Si può notare che, come negli altri task, anche per la classificazione di repertori tendono a migliorare i risultati della classe che performa peggio, che anche in questo caso corrisponde a quella di minoranza.

La classe di maggioranza rimane praticamente inalterata. Si può notare l'effetto positivo su tutti i parametri dell'utilizzo delle informazioni portate dai repertori. Nel caso dell'utilizzo del classico modello Alberto con l'aggiunta della codifica One Hot Encoder possiamo notare come la deviazione standard rimane inalterata, la media risulta sostanzialmente più alta e di conseguenza produce anche il modello con il valore massimo.

La soluzione migliore globale in termini di media la produce la tecnica pura del transfert learning sulla classificazione dei repertori. Come tutti i modelli che usano di base questa tecnica, produce i risultati migliori anche per la deviazione standard.

Stereotipi	Alberto	Alberto One Hot	Alberto Denso
F1 non stereotipi media	0.8219	0.8194	**0.8284**
F1 stereotipi media	0.4678	0.5010	0.5340
F1 macro media	0.6448	0.6602	0.6812
F1 macro massimo	0.7454	**0.7572**	0.7424
F1 deviazione standard	0.0686	0.0686	0.0587
Stereotipi	Repertori	Repertori One Hot	Repertori Denso
F1 non stereotipi media	0.8266	0.8258	0.8268
F1 stereotipi media	**0.5629**	0.5616	0.5480
F1 macro media	**0.6948**	0.6937	0.6874
F1 macro massimo	0.7276	0.7456	0.7370
F1 deviazione standard	0.0464	0.0464	0.0468

Tabella 5.5: Risultati Alberto con e senza repertori con codifica stereotipi.

Nel boxplot in Figura 5.5 si può vedere come i risultati in tabella vengono confermati; si può anche notare come il miglior modello raggiunge valori molto simili a quelli ottenuti nello stato dell'arte e in questo caso la tecnica del transfert learning ha un buon effetto, sopratutto nella distribuzione dei risultati, che sono più raggruppati.

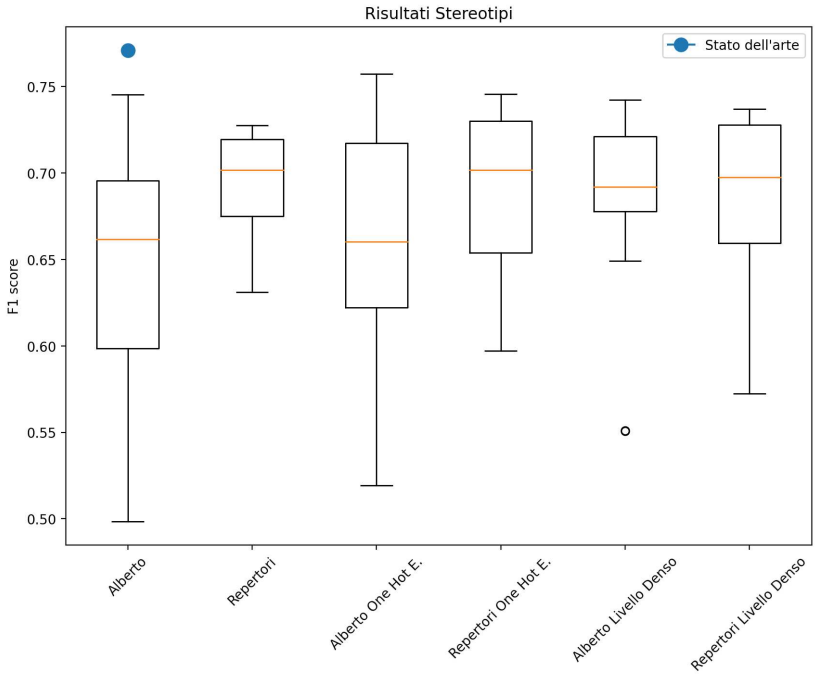

Figura 5.5: Boxplot modello con e senza repertori codifica stereotipi.

5.2.2 Conclusioni

I risultati prodotti per entrambi i task sono migliori con l'utilizzo dei repertori, quindi si può constatare che i repertori possono apportare miglioramenti non solo nei contesti di classificazione di ironia e soggettività, ma anche in contesti diversi, o dove vengono comunque implementati meccanismi che ricordino l'ironia. Una caratteristica interessante e di come anche in questo caso condividono la tendenza a migliorare la classe di minoranza.

Capitolo 6

Analisi delle predizioni e metodi ensemble

Una volta definito che anche per l'hate speech e per la classificazione di testi con stereotipi i repertori discorsivi offrono un importante aiuto, sarebbe interessante scoprire quali sono le caratteristiche che vengono favorite.

Per fare ciò è necessario focalizzarsi sull'ironia, sulla soggettività, sull'hate speech e sulla classificazione di stereotipi, essendo i task che hanno visto un incremento importante delle prestazioni usando la teoria dialogica.

Un fattore interessante sarà sicuramente analizzare la tendenza di tutti i modelli che utilizzano la scienza dialogica a predire in maniera migliore la classe di minoranza. Si andrà ad implementare anche l'analisi utilizzando l'accuratezza, che può arricchire le informazioni date dalla F1-score.

6.1 Ironia

La prima cosa interessante da controllare è il confronto tra F1-score e l'accuratezza per analizzare se l'incremento delle prestazioni sia dato da un aumento effettivo degli esempi predetti correttamente o se è stato uno spostamento, passando a predire in maniera maggiore la classe di minoranza. Il confronto è esposto in Tabella 6.1.

Ironia	F1-score media	Accuratezza media
Modello senza repertori	0.6296	**0.8705**
Modello con repertori	**0.6521**	0.8576

Tabella 6.1: Accuratezza vs F1-score ironia.

Si può notare che l'accuratezza media tende a calare, a differenza del F1-score; questo è dovuto al fatto che il modello tende a predire in valore assoluto un numero minore di esempi corretti, che però appartengono maggiormente alla classe di minoranza.

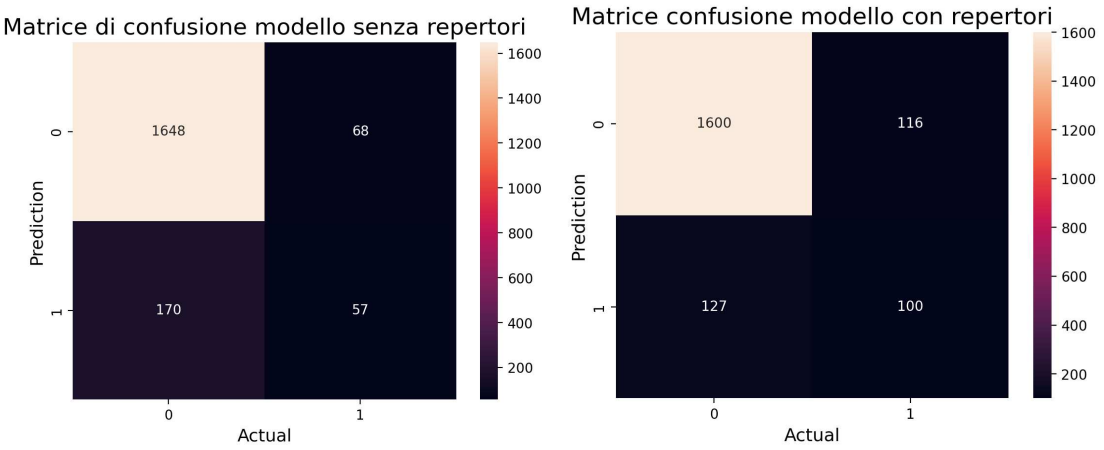

Figura 6.1: (a) Distribuzione repertori secondo ironia (b) Distribuzione repertori secondo ironia

Per vedere in maniera più oculata le modalità con cui il modello predice le due classi in Figura 6.1 si è stampato le due matrici di confusione, una per ogni tipologia di modello. É facile notare che il modello con i repertori tende a predire maggiormente la classe di minoranza, ecco perché le prestazione della F1 crescono notevolmente, e in maniera maggiore per il task sbilanciato, ovvero l'ironia. La F1 favorisce particolarmente un modello che predice meglio la classe con le prestazioni peggiori.

6.2 Soggettività

I risultati della accuratezza relativa alla soggettività crescono leggermente, come l'F1-score medio. Dalla matrice di confusione in Figura 6.2 si può notare che anche in questo caso viene predetta maggiormente la classe di minoranza, come nel caso dell'ironia; però c'è una crescita assoluta anche nelle predizione corrette, come possiamo vedere dall'accuratezza in Tabella 6.2.

Soggettività	F1-score media	Accuratezza media
Modello senza repertori	0.7307	0.7850
Modello con repertori	**0.7571**	**0.7924**

Tabella 6.2: Accuratezza vs F1-score soggettività.

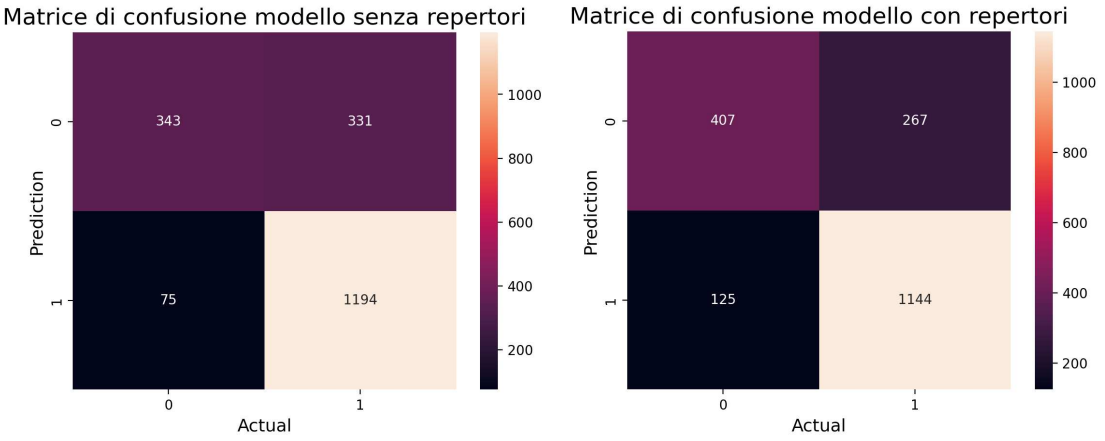

Figura 6.2: (a) Distribuzione repertori secondo soggettività (b) Distribuzione repertori secondo soggettività

6.3 Hate speech

Per quanto riguarda l'hate speech, è chiaro che in questo caso l'accuratezza cresce in maniera importante. Questo incremento è accompagnato anche da un'ottima crescita della F1-score.

Hate speech	F1-score media	Accuratezza media
Modello senza repertori	0.7088	0.7671
Modello con repertori	**0.7322**	**0.7768**

Tabella 6.3: Accuratezza vs F1-score hate speech.

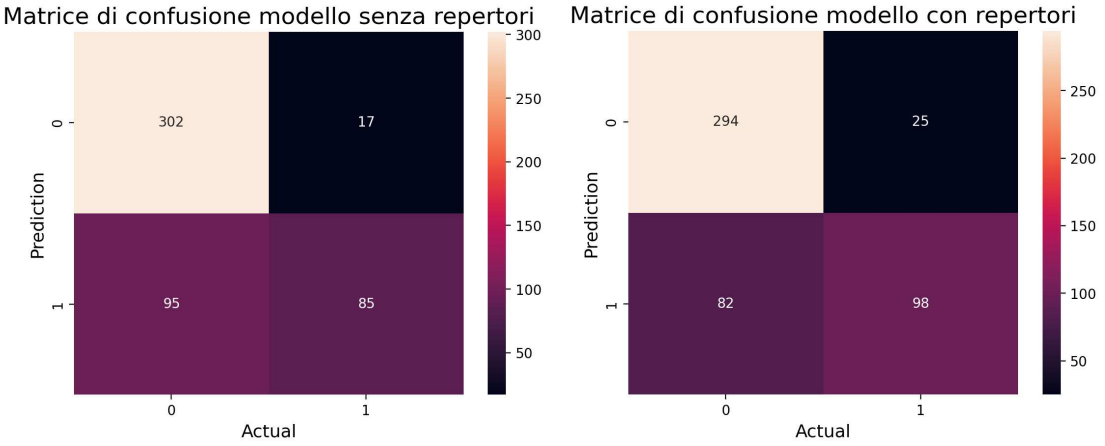

Figura 6.3: (a) Matrice di confusione modello senza repertori hate speech(b) Matrice di confusione modello con repertori hate speech

6.4 Stereotipi

Anche gli stereotipi si comportano in maniera analoga rispetto ai task precedenti: i risultati della classe di minoranza subiscono un incremento e, come nel caso dell'hate speech, cresce il numero assoluto di predizioni corrette.

Stereotipi	F1-score media	Accuratezza media
Modello senza repertori	0.6448	0.7350
Modello con repertori	**0.6948**	**0.7524**

Tabella 6.4: Accuratezza vs F1-score stereotipi.

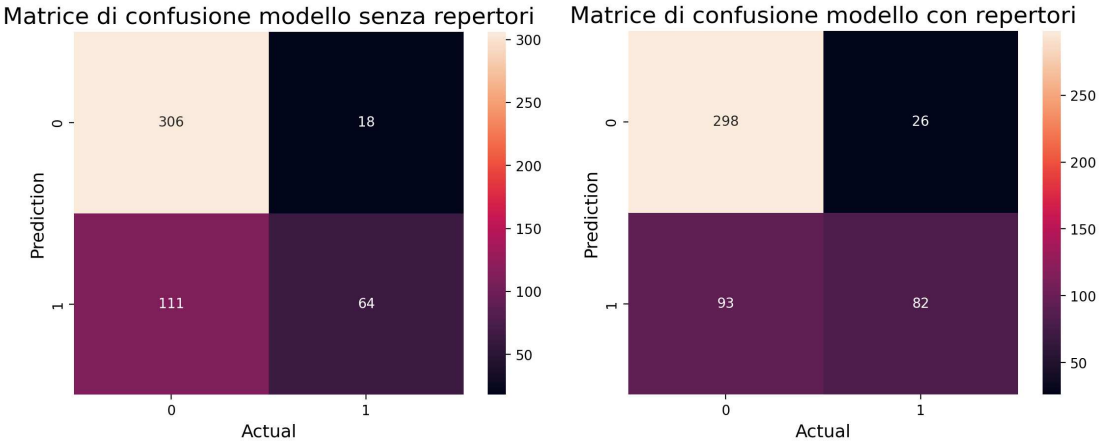

Figura 6.4: (a) Distribuzione repertori secondo stereotipi (b) Distribuzione repertori secondo stereotipi

6.5 Modello ensemble

Dalle analisi precedenti si è preso atto di come i modelli che utilizzano i repertori tendono a predire in maniera migliore le classi di minoranza rispetto a quelli che non li usano. Questo però indicherebbe che le predizioni tra i due modelli variano significativamente, di conseguenza potrebbe essere utile applicare un meccanismo di ensemble, ovvero, anzichè basarsi sulla predizione di un unico modello, si allenano modelli differenti per poi farli votare. La predizione corrisponderà alla classe dell'esempio più votato.

Quindi il primo approccio che si è deciso di adottare è quello di far votare un modello che è allenato nella predizione dei repertori ed un modello che non lo è. Se si implementa un sistema con due modelli si deve avere qualche accortezza: ad esempio, per classificare un testo come ironia, si deve definire se si voglia il voto di entrambi, o è sufficiente uno dei due.

Nel primo caso si tende a favorire la predizione dell'ironia -se il modello è "indeciso" tende a predirla comunque-, nel secondo caso invece la predice solo se ne è estremamente sicuro, ovvero quando il risultato dei due modelli corrisponde. Queste sono caratteristiche che dipendono fortemente dall'obiettivo che ci si è posti. Se ad esempio in un social voglio eliminare tutti i commenti che reputo offensivi, però voglio essere sicuro che non siano ironici, preferirò un modello che predice in maniera estremamente corretta la non ironia.

Questo approccio dal punto di vista delle prestazioni non ha dato buoni risultati, di conseguenza si è scelto un approccio più complesso che coinvolgesse più modelli nella classificazione. Visto che per ogni approccio, con l'utilizzo dei repertori o senza, sono stati prodotti 15 modelli, si è deciso di farli votare tutti, 15 per uno e 15 per l'altro.

Questo permette di avere una varianza maggiore e evitare che le scelte prese da un solo modello influenzino eccessivamente la scelta finale. Infatti tendenzialmente i modelli di ensemble sono maggiormente efficienti nel caso i cui alla votazione partecipino un buon numero di componenti. Anche in questo caso le prestazioni sono state deludenti, producendo risultati simili alla media di tutti i modelli. Di conseguenza si è provato a separare le due tipologie di approcci facendoli votare per ogni tipologia, ovvero fare votare i 15 modelli che utilizzano il pre-allenamento sui repertori e quelli che non lo fanno.

In questo caso però può essere comunque estremamente utile per confrontare i due sistemi di votazione a 15, permettendo di definire quale dei due approcci produce maggiore varietà nei risultati, ovvero se uno dei due modelli tende a predire correttamente gli stessi risultati oppure predice correttamente risultati differenti.

In più i modelli di ensemble possono essere estremamente utili come regolarizzatori. Per motivi computazionali non è stato possibile provare più di 15 votazioni in parallelo per ogni tipologia.

Esistono diverse tecniche di votazione all'interno dei modelli: si può imporre che ognuno abbia lo stesso peso o variare il peso in base alle prestazioni che producono nel set di validazione. Il più efficiente è risultato quello più semplice, che associa lo stesso peso ad ogni modello. Per quanto riguarda i modelli pesati, sono stati provati differenti approcci. In un primo momento, si è utilizzato un approccio di premiazione lineare, dando più potere di voto ai modelli con le prestazioni migliori nel set di validazione.

Successivamente si è provato a lavorare in maniera esponenziale (sempre premiando i modelli con prestazioni migliori). Questa tecnica sembrava efficace e portava buoni risultati. Vi era però una criticità: aumentando il peso in maniera esponenziale, si finisce per far valere troppo il voto del modello con le prestazioni migliori, finendo con l'avere un modello uguale al modello migliore.

Dalla Tabella 6.5 possiamo notare come generalmente l'approccio ensemble è particolarmente efficace per i modelli che utilizzano i repertori discorsivi. Questo evidenzia che l'utilizzo dei repertori permette di aumentare la varianza delle predizioni: mentre il modello classico tende a predire correttamente gli stessi esempi, il modello con i repertori tende a predire esempi estremamente diversi tra di loro.

L'incremento in termini di F1-score è sempre positivo, mentre invece per l'accuratezza è più contenuto ed avviene solo per l'hate speech e gli stereotipi.

La Tabella 6.5 mostra come la media dei modelli performa meglio del modello ensemble che non usa i repertori, mentre è sempre inferiore rispetto al modello ensemble che usa i repertori.

Inoltre è interessante notare come per tutti i task, ad eccezione dell'ironia, l'utilizzo del modello ensemble porta miglioramenti anche nel caso in cui non si utilizzino i repertori.

La Figura 6.5 mostra come nel caso dell'ironia il risultato del modello ensemble restituisca un valore maggiore rispetto al valore massimo, per gli altri task invece occupa una posizione superiore alla mediana e alla media dei 15 modelli.

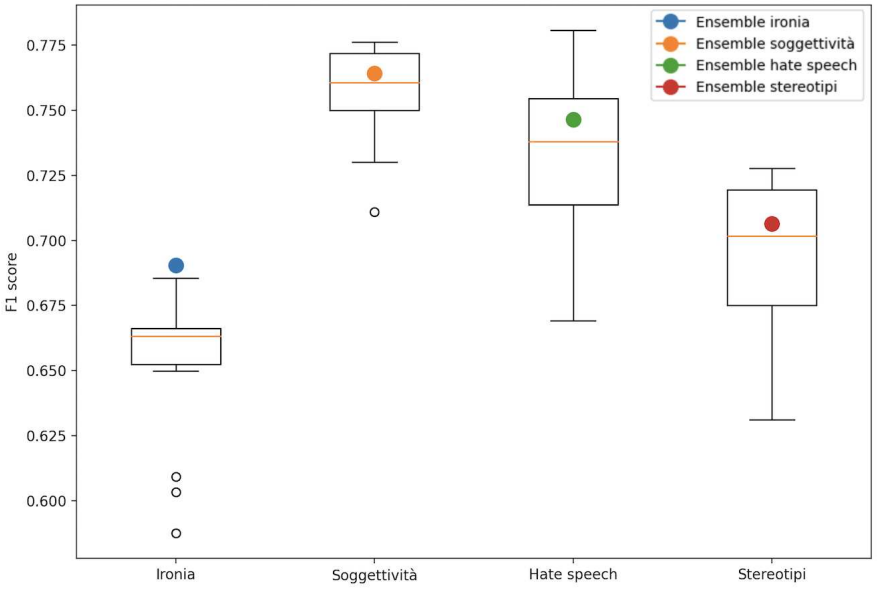

Figura 6.5: Risultati ensemble vs serie di modelli

6.6 Conclusioni

É palese come questi risultati ci confermano il pattern che si è seguito, dove l'ironia è il task che ha ricevuto il miglior incremento dal punto di vista delle prestazioni. É sicuramente interessante notare come l'utilizzo del modello ensemble è un approccio vincente perché migliora le prestazioni, a prescindere dall'utilizzo dei repertori. É inoltre facile notare dalla Tabella 6.5 che l'approccio con l'ensemble permette di raggiungere risultati superiori rispetto alla media dei singoli modelli che usano i repertori.

Andando poi ad analizzare la combinazione tra ensemble e scienza dialogica, si può notare che oltre ad un aumento delle prestazioni medie si ottiene una forte regolarizzazione.

Ironia	F1-score	Accuratezza
Media modelli senza repertori	0.6296	0.8705
Media modelli con repertori	0.6521	0.8576
Modello ensemble senza repertori	0.6282	**0.8775**
Modello ensemble con repertori	**0.6904**	0.8749
Soggettività	F1-score	Accuratezza
Media modelli senza repertori	0.7307	0.7850
Media modelli con repertori	0.7571	0.7924
Modello ensemble senza repertori	0.7414	0.7910
Modello ensemble con repertori	**0.7643**	**0.7982**
Hate speech	F1-score	Accuratezza
Media modelli senza repertori	0.7088	0.7671
Media modelli con repertori	0.7322	0.7768
Modello ensemble senza repertori	0.7232	0.7755
Modello ensemble con repertori	**0.7464**	**0.7855**
Stereotipi	F1-score	Accuratezza
Media modelli senza repertori	0.6448	0.7350
Media modelli con repertori	0.6948	0.7524
Modello ensemble senza repertori	0.6619	0.7414
Modello ensemble con repertori	**0.7065**	**0.7615**

Tabella 6.5: Accuratezza vs F1-score.

6.7 Generalizzazione

Un punto fondamentale su cui si è basato questo lavoro è la generalizzazione che è possibile ottenere attraverso l'utilizzo dei repertori, ovvero come questi rendano possibile ottenere una soluzione più generale e meno vincolata al dataset che si usa come esempio.

Per provare ciò però bisogna affidarsi ad un dataset diverso rispetto a quello al quale ci siamo affidati precedentemente nella classificazione dell'ironia, per vedere come il modello vecchio performa su un nuovo dataset.

Nel 2018 sempre il gruppo EVALITA ha organizzato la competizione ironITA, una competizione dove dato un testo l'obbiettivo era riconoscere se poteva essere etichettato come ironico o no.

Di conseguenza, in questo lavoro, si è deciso di utilizzare il dataset di test fornito per testare le prestazioni del modello BERT per l'ironia, con repertori e senza, nel caso di nuove tipologie di dati. Nel dataset precedente l'argomento principale era la riforma Buona Scuola di Renzi, in questo caso gli argomenti riguarderanno in maniera più generale la posizione politica e l'hate speech.

Un'altro punto estremamente interessante è anche il fatto che questo dataset è estremamente bilanciato, contenendo circa il 50% di frasi ironiche e 50% di frasi non ironiche. Questo potrebbe mostrare se i miglioramenti osservati nel dataset precedente dell'ironia siano frutto anche di un maggior numero di esempi predetti per la classe di minoranza.

Si è quindi preso i 15 modelli che non usano i repertori precedentemente allenati e i 15 modelli che usano il transfert learning dai repertori e che concatenano la codifica del livello di output (l'approccio che dava i risultati migliori) e sono stati testati per il dataset di ironITA.

6.7.1 Risultati

In Figura 6.6 si possono apprezzare i risultati di questo test: si può notare come anche in questo caso il modello che sfrutta i repertori funziona in maniera migliore rispetto al modello che non lo fa, tanto dal punto di vista della varianza nella distribuzione dei risultati, quanto per quello che riguarda la mediana e il valore massimo.

Nel caso dei repertori sono presenti 3 valori anomali, per i quali il modello preforma estremamente peggio, cosa che però succedeva anche nei test sul dataset originale, visibili in Figura 4.2.

Ulteriormente si può notare come anche in questo caso far votare i 15 modelli porta risultati positivi, e che ancora una volta il modello che usa i repertori ha un notevole vantaggio.

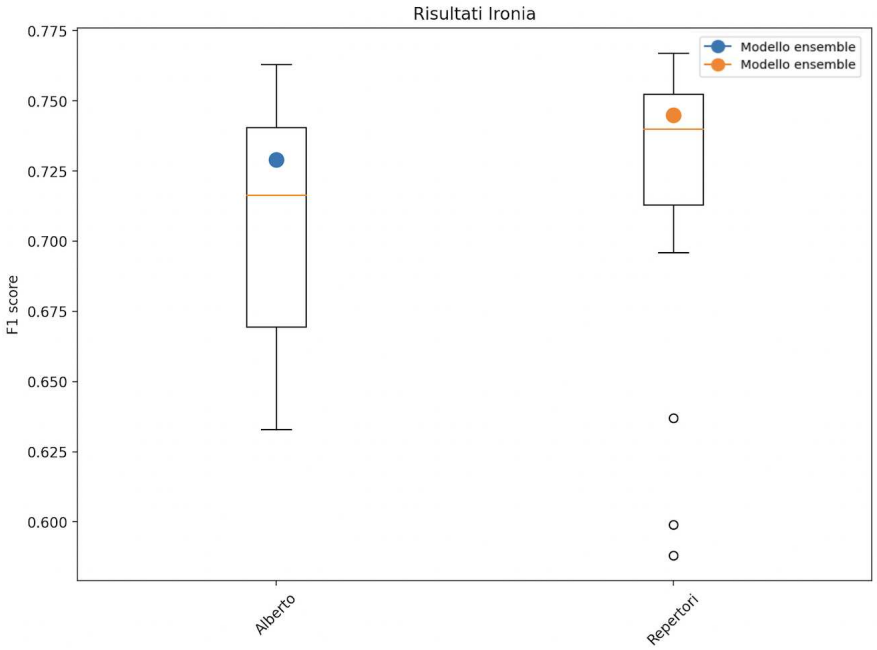

Figura 6.6: Modello con e senza repertori sul dataset ironITA 2018

I risultati prodotti dal modello ensemble sono visibili in Tabella 6.6 e possono essere considerati estremamente positivi. Se li andiamo a confrontare con i risultati ottenuti nella competizione, il modello Alberto ensemble senza repertori ottiene valori simili al miglior risultato, mentre il modello che considera i repertori ottiene addirittura valori migliori.

Questo può essere considerato notevole perché questo modello, a differenza di quelli della competizione, non è mai stato allenato su quella tipologia di testi, il che implica un'ottima capacità di generalizzazione.

Ironia	F1 macro
Miglior risultato competizione IronIta 2018	0.731
Modello ensemble senza repertori	0.729
Modello ensemble con repertori	**0.754**

Tabella 6.6: Risultati modello sul dataset ironITA 2018.

Capitolo 7

Conclusioni

Questo lavoro mostra come la scienza dialogica possa dare un innegabile contributo ai principali task NLP legati all'ironia. Infatti elementi come ironia, soggettività, hate speech e stereotipi mostrano un innegabile miglioramento sia in termini di media, che in termini di deviazione standard.

Dal punto di vista della media c'è un aumento dei valori condiviso da tutti i principali task, mentre, per quel che riguarda la deviazione standard, il miglioramento avviene solo in alcuni casi. Questo indica che il modello ha comunque una varianza importante, ma con l'utilizzo dei repertori c'è certamente una stabilizzazione dei risultati. Questi risultati, oltre che a variare meno tendono anche ad essere maggiori.

In oltre è chiaro come i repertori discorsivi non aiutino nel caso del task della polarità, ovvero la positività e la negatività.

Inoltre in questo lavoro sono state esplorate diverse tecniche per aggregare informazioni testuali a delle informazioni che riguardano l'appartenenza di testi a determinate classi; questo è stato fatto in maniera indiretta tramite l'utilizzo del transfert learning, o in maniera diretta, concatenando la rappresentazione del testo tramite i repertori discorsivi appartenenti a quegli elementi.

Queste due tipologie di approcci hanno dato risultati considerevoli, e potrebbero essere applicati indipendentemente dai repertori discorsivi, ovvero utilizzando differenti regole linguistiche o proprietà.

Infine, abbiamo mostrato come i modelli BERT si prestino bene all'implementazione dei metodi ensemble, grazie alla varietà di predizioni effettuate per ogni istanza del modello. Questi risultati sono sembrati ancora più efficaci nel caso in cui il modello BERT considerasse anche l'informazione fornita dai repertori.

Si è mostrato che l'utilizzo delle informazioni fornite dalla scienza dialogica sono rimaste determinanti per la classificazione dell'ironia anche con l'utilizzo di un altro dataset contenente argomenti diversi, e con un bilanciamento perfetto tra le due classi.

Sarebbe utile esplorare altri task, magari anche indipendenti dall'ironia, per vedere in che altri contesti i repertori possono dare il loro contributo. Potrebbe essere estremamente interessante anche cambiare la tipologia di testi utilizzati per la classificazione. Grazie alla loro facilità di reperimento, nelle competizioni -sopratutto in lingua italiana- si fa un grande utilizzo di tweet; questo però comporta alcuni

limiti. I tweet sono testi abbastanza brevi, che nel migliore dei casi contengono al massimo due o tre repertori discorsivi differenti. L'utilizzo di testi più lunghi potrebbe portare notevoli miglioramenti perché porterebbe maggiori informazioni per ogni singolo testo.

Infine, si potrebbe considerare il fatto di estendere tutto il lavoro svolto alla lingua inglese, dal momento che la scienza dialogica definisce le regole del linguaggio, non della lingua. Di conseguenza tutta la teoria dialogica, anche se è stata sviluppata utilizzando la lingua italiana, potrebbe avere applicazione anche in lingua inglese, visto che le regole definite non sono strettamente legate alla grammatica italiana, ma al significato che il testo porta con se.

Bibliografia

[1] Treccani.it - vocabolario treccani on line, 2011. URL `http://www.treccani.it`.

[2] Valerio Basile and Malvina Nissim. Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107, Atlanta, 2013.

[3] Valerio Basile, M Di Maro, D Croce, L Passaro, et al. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In *CEUR WORKSHOP PROCEEDINGS*, volume 2765. CEUR-ws, 2020.

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[5] Michele Bortone. A deep learning approach for discursive repertoires prediction in online texts.

[6] Alessandra Teresa Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti, and Paolo Rosso. In *Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)*, volume 2263, pages 1–6. CEUR-WS, 2018.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] Elisabetta Fersini, Debora Nozza, Paolo Rosso, et al. Overview of the evalita 2018 task on automatic misogyny identification (ami). In *CEUR workshop proceedings*, volume 2263, pages 1–9. CEUR-WS, 2018.

[9] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[10] Turchi G. P. Orrù L. *Metodologia per l'Analisi dei Dati Informatizzati Testuali. Fondamenti di teoria della misura per la Scienza Dialogica*. Edises, 2014.

[11] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*, 2020.

[12] Luisa Orrù, Christian Moro, Marco Cuccarini, Monia Paita, Marta Silvia Dalla Riva, Davide Bassi, Giovanni Da San Martino, Nicolò Navarin, and Gian Piero Turchi. Machine learning and madit methodology for the fake news identification: the persuasion index. In *4th International Conference on Advanced Research Methods and Analytics (CARMA 2022)*, pages 165–172. Editorial Universitat Politècnica de València, 2022.

[13] Reynier Ortega-Bueno, Berta Chulvi, Francisco Rangel, Paolo Rosso, and Elisabetta Fersini. Profiling irony and stereotype spreaders on twitter (irostereo). 2021.

[14] Turchi G. P. *Dati senza numeri. Per una metodologia dell'analisi dei dati informatizzati testuali MADIT*. Monduzzi Editore, 2009.

[15] Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR, 2019.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Ringraziamenti

Ringrazio il Prof. Nicolò Navarin che mi ha seguito con pazienza durante tutto il percorso di studi come responsabile nella borsa di studio, e successivamente come relatore della tesi. Ringrazio la Dott.ssa Orrù e il Prof. Turchi per avermi dato la possibilità di conoscere il mondo della Scienza Dialogica.

Ringrazio i miei coinquilini, Emanuele, Lucrezia e Valentina con cui ho condiviso momenti semplici, ma belli di vita quotidiana.

Ringrazio tutti gli amici che ho conosciuto a Padova: Francesca, Angelica, Claudia, Davide, Leonardo, Benedetto, Roberto e Sol, persone che mi hanno permesso di vivere a pieno questa città.

Ringrazio gli amici del gruppo Fede, Youssef, Jacopino. Persone con cui ho condiviso ideali e sogni. Alessandro e Giulio, due amici meravigliosi sempre presenti nei momenti più importanti e su cui potrò sempre contare.

Alla mia famiglia, mia madre Franca, mia sorella Elena, mio padre Orfeo e mio fratello Davide per il loro supporto, che da sempre mi permette di inseguire i miei sogni.

Grazie a Padova per avermi accolto, ospitato e fatto sempre sentire a casa.