



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

**Scuola di Scienze**

**Dipartimento di Informatica, Sistemistica e Comunicazione**

**Corso di laurea in Informatica**

# **Tecniche di Natural Language Processing per il riconoscimento dei discorsi d'odio sui social network**

**Relatore:** Prof.ssa Elisabetta Fersini

**Correlatore:** Prof. Antonio Candelieri

**Relazione della prova finale di:**

Daniel Scalena

Matricola 844608

**Anno Accademico 2020-2021**

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Il problema dei contenuti offensivi nei social Network . . . . .	1
1.1.1	Panoramica generale su TikTok . . . . .	2
1.2	Presentazione del lavoro svolto . . . . .	2
<b>2</b>	<b>Stato dell'arte</b>	<b>4</b>
2.1	Definizione dei discorsi d'odio . . . . .	4
2.2	Lavori relativi al riconoscimento dei discorsi d'odio . . . . .	5
2.2.1	Dizionari . . . . .	6
2.2.2	Bag of words e N-grams . . . . .	7
2.2.3	Analisi sulla sintassi . . . . .	7
2.2.4	Analisi dei contenuti . . . . .	8
2.2.5	Classificazione supervisionata e deep learning . . . . .	9
2.3	Strumenti tecnologici utilizzati . . . . .	10
2.3.1	Il modello BERT . . . . .	11
<b>3</b>	<b>Framework proposto</b>	<b>12</b>
3.1	Overview del sistema proposto . . . . .	12
3.2	Raccolta e gestione dei dati . . . . .	14
3.2.1	Web scraping . . . . .	14
3.2.2	Pulizia dei dati . . . . .	15
3.3	Analisi del lessico . . . . .	15
3.3.1	Lemmatizzazione . . . . .	16

3.4	Classificazione con BERT . . . . .	16
3.4.1	Classificazione manuale e linee guida adottate . . . . .	17
3.4.2	Scelta del modello e fine tuning . . . . .	17
3.4.3	Costruzione della rete neurale con BERT . . . . .	18
3.4.4	Ottimizzazioni effettuate sulla rete neurale . . . . .	18
<b>4</b>	<b>Risultati sperimentali</b>	<b>20</b>
4.1	Classificazione con il lessico Hurtlex . . . . .	20
4.1.1	Distribuzione delle categorie ed esempi descrittivi . . . . .	20
4.1.2	Distinzione tra lemmi conservativi e inclusivi . . . . .	21
4.2	Analisi esplorativa dei dati . . . . .	22
4.2.1	Analisi di genere e movimenti temporali . . . . .	23
4.3	Fine tuning del modello BERT base . . . . .	24
4.3.1	Valutazione della funzione obiettivo . . . . .	24
4.3.2	Metriche sul dataset di test . . . . .	26
4.4	Modifiche al modello BERT base . . . . .	30
4.4.1	BERT con layer Bi-LSTM . . . . .	30
4.4.2	BERT con layer lineare, dropout e di classificazione . . . . .	31
4.4.3	Confronto risultati con le modifiche effettuate . . . . .	32
<b>5</b>	<b>Conclusioni e sviluppi futuri</b>	<b>34</b>
5.1	Conclusioni sul lavoro svolto . . . . .	34
5.2	Possibili miglioramenti proposti e sviluppi futuri . . . . .	35
<b>References</b>		<b>37</b>

# 1

## Introduzione

### 1.1 Il problema dei contenuti offensivi nei social Network

L'uso dei social network è ormai da molti anni entrato nelle abitudini quotidiane di una gran parte della popolazione mondiale: il loro uso semplice e immediato permette a persone di ogni età di interfacciarsi con un mondo infinito di informazioni che rispecchiano la realtà che ordinariamente ci circonda. Tra le innumerevoli opportunità offerte e gli enormi vantaggi della vita online spesso la regolarizzazione dei contenuti non riesce a mantenere il passo. Risolvere questo problema rappresenta un'ardua sfida soprattutto in un mondo virtuale dove la possibilità di mantenere una pseudo anonimità rende il lavoro di gran lunga più complicato che nel mondo reale.

La quantità di dati presenti sulle piattaforme social supera di gran lunga la capacità di controllo a disposizione delle aziende proprietarie e per questo motivo, solamente negli ultimi anni, la ricerca si sta orientando verso il riconoscimento automatico dei contenuti offensivi. Il compito di supervisionare e riconoscere violazioni delle norme di servizio, o

in questo caso quello che è in gergo chiamato *hate speech*, è tuttora affidato a un gruppo ristretto di lavoratori chiamati ironicamente "Deciders" [25]. Nonostante gli aiuti forniti da algoritmi di riconoscimento automatico, rimane comunque necessaria una supervisione umana che implica un lavoro decisamente più grande di quello che si possa comunemente immaginare. Il perfezionamento di algoritmi in grado di riconoscere automaticamente contenuti offensivi risulta quindi essere sempre più una questione di urgenza e di assoluta priorità per tutelare nel miglior modo possibile gli utenti che frequentano quotidianamente le piattaforme social.

### 1.1.1 Panoramica generale su TikTok

Tra tutti i diversi social network, ormai consolidati e conosciuti da diversi anni, TikTok risulta essere non solo l'ultimo arrivato ma anche quello che sta subendo una crescita maggiore nell'ultimo periodo. Il suo ambiente, frequentato prevalentemente da giovanissimi, lo rende per sua stessa natura in continua evoluzione con trend che ne cambiano le modalità di utilizzo quasi giornalmente. Allo stato attuale il controllo dei contenuti offensivi è eseguito principalmente in maniera manuale basandosi sulla sola segnalazione degli utenti o sull'azione dei moderatori che creano i contenuti sulla piattaforma. La mancanza di un sistema automatico efficace in grado di classificare ciò che risulta essere offensivo rende complicato contenere, soprattutto in caso di creator con un grande seguito, i discorsi d'odio sulla piattaforma.

L'obiettivo è quindi migliorare gli strumenti di identificazione fornendosi delle più recenti ricerche nel campo del riconoscimento del linguaggio naturale per rendere TikTok, o più in generale ogni piattaforma social online, un luogo virtuale più sicuro per le persone che lo frequentano ogni giorno.

## 1.2 Presentazione del lavoro svolto

La ricerca di un sistema di classificazione efficace, capace di comprendere le più piccole sfumature del linguaggio sapendone interpretare ogni piccolo aspetto, è sempre stato tra i compiti più difficili per dei sistemi automatici privi di esperienza. Il lavoro svolto mira

a fornire una soluzione efficace a questo problema esplorando le varie possibilità offerte della comunità scientifica nel campo del riconoscimento del linguaggio naturale e, in particolare, vengono proposte diverse soluzioni in grado di identificare i discorsi d'odio.

Come primo approccio è stato progettato un sistema capace di riconoscere parole offensive e denigratorie nei commenti scaricati: i risultati ottenuti non riescono ad essere sufficientemente precisi nell'identificazione dei commenti negativi, problematica dovuta ad una serie di caratteristiche legate alla comunicazione su TikTok. Una valida alternativa è rappresentata dall'utilizzo di reti neurali profonde in grado di apprendere al meglio il contesto di un commento. Attualmente lo stato dell'arte è rappresentato dai modelli BERT di Google la cui capacità di codifica del testo riesce efficacemente nella rappresentazione di ogni parola nel contesto entro la quale è inserita. I risultati di questo approccio risultano essere più che validi se confrontati con gli stessi ottenuti dalla classificazione lessicale. Viene infine proposto un ultimo lavoro di ottimizzazione, modificando la rete neurale originale e cercando di migliorarne le relative prestazioni. Il sistema risultante evidenzia un'ottima performance tanto da riuscire nella maggior parte dei casi a comprendere appieno il contesto e il senso dei commenti pubblicati online.

# 2

## Stato dell'arte

### 2.1 Definizione dei discorsi d'odio

Nonostante la grande presenza di discorsi d'odio nel panorama dei social network, non è ancora presente una definizione nitida e universale che riesca a inquadrare il fenomeno entro una cornice precisa. Le diverse definizioni, di quello che è comunemente chiamato *hate speech*, sono principalmente fornite da organizzazioni sovranazionali come l'Unione Europea in [42], da organizzazioni internazionali per le minoranze (ILGA [20]), da documenti scientifici e dai termini e condizioni delle principali compagnie tecnologiche operanti nel settore social (Facebook, YouTube e Twitter in [14, 51, 38]).

In [29] viene svolto un lavoro di analisi su queste fonti e vengono sommarizzate efficacemente fornendo la seguente definizione tradotta di *hate speech*:

*Il discorso d'odio è un linguaggio che attacca o sminuisce, incita violenza o odio verso dei gruppi, basandosi su specifiche caratteristiche quali l'apparenza fisica, la religione, la discendenza, la nazionalità o le origini etniche,*

*l'orientamento sessuale, l'identità di genere o altro, e può verificarsi in diversi stili linguistici, anche in forma subdola o con l'uso di umorismo.*

Su questa definizione appena introdotta si baserà il lavoro svolto per la classificazione dei contenuti d'odio online.

## 2.2 Lavori relativi al riconoscimento dei discorsi d'odio

Il problema di riconoscimento dei discorsi d'odio sui social network, seppur sia ancora nella sua fase iniziale, è in rapida evoluzione. Ad esempio, non è ancora presente un benchmark universale per testare le performance di una qualsiasi tecnica utilizzata e la gran parte dei testi usati per la classificazione vengono generati caso per caso, variando di molto il tipo di linguaggio utilizzato [48], tuttavia è presente un vasto numero di metodologie in grado di identificare testi offensivi e denigratori. Eccezion fatta per pochi casi ([34, 44, 10, 50, 39]), la stragrande maggioranza di dataset rimangono spesso privati, limitando la possibilità di eventuali confronti tra diversi metodi di classificazione. Un'ulteriore problematica presente nel settore è costituita dalla scarsa variabilità relativa ai social network utilizzati, difatti la maggior parte degli studi usa Twitter come unica piattaforma per la raccolta di informazioni. Tra le probabili cause di questo fenomeno è sicuramente presente l'oggettiva difficoltà riscontrata durante la raccolta dei dati, molte volte limitata o perfino bloccata dalle stesse piattaforme social per tutelare la privacy dei propri utenti. Esplorando la letteratura presente è inoltre evidente come la lingua più analizzata sia l'inglese, con alcune più rare eccezioni per le lingue europee più diffuse quali tedesco, spagnolo, francese e italiano, rimanendo comunque poco sufficienti a fornire un quadro chiaro della situazione dei fenomeni di odio online nei diversi paesi del mondo.

Diversi studi sono stati effettuati con lo scopo di riconoscere i diversi tipi di hate speech online classificandoli in base ai loro attributi: nella maggior parte dei casi le statistiche descrittive evidenziano una forte presenza di razzismo [43], sessismo e misoginia [4, 34, 47], pregiudizi contro immigrati [37] e omofobia [33]. Nonostante la presenza di diverse categorie però, una gran parte degli studi si concentra su una classificazione

esclusivamente binaria, lasciando in sospeso ad eventuali sviluppi futuri il riconoscimento di tutte le sottocategorie precedentemente elencate.

Il riconoscimento dei discorsi d'odio online viene svolto attraverso tecniche di text mining, ovvero un'analisi del lessico utilizzato nei dati relativi alla ricerca. Generalmente vengono prese in considerazione diverse caratteristiche dei testi per trovare la/e combinazioni che permettono una migliore classificazione dei discorsi d'odio.

### 2.2.1 Dizionari

La strategia più semplice utilizzata per l'analisi dei testi è sicuramente l'uso dei dizionari, ossia raccolte di parole offensive e denigratorie che vengono generalmente usate in commenti contenenti discorsi d'odio. Contando il numero di occorrenze delle parole offensive o usando la normalizzazione sulla base della lunghezza del testo considerato, è possibile generare un punteggio che indichi con quanta probabilità il commento appartenga alla classe dei discorsi d'odio.

Diversi miglioramenti a questo approccio vengono evidenziati in studi [19, 6, 35] che fanno uso della *distance metric*, una tecnica in grado di mitigare il problema del mascheramento delle parole. Questa procedura, usata dagli utenti online per ingannare il riconoscimento automatico dei termini offensivi, riguarda nella maggior parte dei casi la sostituzione di un carattere all'interno di una parola pur mantenendone un aspetto visivo simile all'originale. Come esempio esplicativo, calcolando il minimo numero di modifiche da apportare la parola mascherata “*stupId0*”, è possibile associarla alla sua forma originale “*stupido*”, presente in un qualsiasi dizionario come termine chiaramente offensivo.

Il confronto tra i testi esaminati e i dizionari contenenti parole offensive come [13, 16, 24, 21] riesce nella maggior parte dei casi a individuare i commenti negativi, pur mancando ancora di precisione nella divisione delle classi e generando una grande quantità di falsi positivi. Osservando il lavoro svolto in [26] è evidente come una classificazione di questo tipo sia inefficace nel rappresentare il contesto entro il quale un determinato termine è inserito: secondo lo studio il 48% dei testi presi in esame non rientra nella classe dei negativi nonostante contenga un'alta percentuale di parole denigratorie.

Un ulteriore supporto per la classificazione viene fornito dall'analisi di diversi fattori concorrenti ad un probabile atteggiamento denigratorio da parte di un commento [7, 46]. Osservando i riferimenti esterni come URL o hashtag è generalmente possibile attribuire un contesto entro il quale il testo in analisi è inserito o, analizzando la sua punteggiatura e la sua capitalizzazione, si è in grado costruire un pattern che caratterizza i commenti di tipo negativo.

### 2.2.2 Bag of words e N-grams

Tra le tecniche che più in assoluto vengono impiegate, anche per migliorare lo scarso riconoscimento del contesto da parte dei dizionari, troviamo sicuramente le raccolte di parole (o *Bag-of-words*) in [49, 17, 43] e le *N-grams* in [41, 49, 46, 17, 15, 6, 45]: la prima consiste in un procedura in grado di creare un corpus a partire dalle parole usate nei dati in analisi ed estrarne successivamente le relative frequenze; nella seconda vengono raccolte una serie di  $N$  parole consecutive in seguito usate per l'addestramento di un modello classificatore. L'utilizzo di *bag-of-words* dimostra però ancora una scarsa efficacia nel riconoscimento del contesto che viene parzialmente migliorata dalla tecnica *N-grams*. Come messo in evidenza in [49] una delle problematiche che affligge *N-grams* riguarda la distanza tra le parole considerate che, se maggiore di  $N$ , influisce negativamente sull'efficacia di questa tecnica. Nel medesimo studio viene altresì proposta una variante della stessa tecnica che, una volta identificati i termini offensivi con l'ausilio di un dizionario, raccoglie un numero prefissato di parole nell'immediato intorno. Come evidenziato in [7] però, il problema principale di queste metodologie è rappresentato dalla numerosità di termini nella finestra considerata: al loro aumentare corrisponde un accrescimento esponenziale della complessità di calcolo necessaria alla classificazione.

### 2.2.3 Analisi sulla sintassi

Grazie ai miglioramenti effettuati su modelli in grado di svolgere analisi grammaticali dei testi, è possibile usare le informazioni sintattiche per meglio comprendere come ogni termine è messo in correlazione con tutti gli altri che compongono una determinata frase (*LSF* o *Lexical Syntactic Feature-based*). Passi avanti in questo senso sono stati svolti da

[18], riuscendo ad associare correttamente gli aggettivi e gli avverbi verso i soggetti a cui si riferiscono. Nello studio [7] viene proposta una tecnica di classificazione in grado di sfruttare queste caratteristiche concentrandosi sugli aggettivi considerati come negativi rivolti verso dei soggetti esterni; i risultati dimostrano un ottimo miglioramento rispetto ai metodi di classificazione precedentemente elencati.

Ulteriori sviluppi riguardanti l'analisi della sintassi di una frase riportati nel medesimo studio riguardano la raccolta delle dipendenze tra le diverse parole utilizzate. In particolar modo vengono analizzate le coppie composte da un termine governatore e dalla sua relativa apposizione per trovare riferimenti negativi attribuiti al soggetto della frase.

#### 2.2.4 Analisi dei contenuti

Ulteriori strategie proposte dalla comunità scientifica comprendono la possibilità di studiare non solo il comportamento di ogni termine della frase in analisi, ma anche l'argomento e il contenuto generale della stessa per aumentare la precisione nel riconoscimento dei discorsi d'odio.

L'analisi dei sentimenti, generalmente negativi in caso di hate speech, è la tecnica più utilizzata in questo senso come in [2, 46, 15, 16, 11, 27]. Metodi di classificazione di questo tipo vengono spesso usati in combinazione con altre caratteristiche come l'analisi dei soggetti di una frase. Negli studi proposti viene ad esempio dimostrato come le minoranze o le persone con delle caratteristiche specifiche, siano i soggetti più esposti al fenomeno di hate speech.

Ulteriori analisi si concentrano sugli argomenti trattati nel testo da classificare, osservando lo schieramento e la polarità di un determinato commento è possibile estrarre un'ulteriore caratteristica che aiuta nel processo di identificazione del discorso d'odio.

Insieme al riconoscimento del testo in alcuni studi vengono impiegati anche diversi elementi che concorrono contemporaneamente alla classificazione di testi denigratori. È il caso di [12] dove sono state analizzate le immagini insieme alla loro descrizione usando diverse tecniche e algoritmi.

Una delle ultime tecniche di text mining usate è la misurazione dell'importanza di un termine analizzando il testo dal quale è stato estratto (*TF-IDF* o *Term Frequency-Inverse*

*Document Frequency*). Osservando la frequenza con cui una parola occorre in una frase è possibile attribuire un punteggio che indichi la sua rilevanza rispetto a tutti gli altri termini presenti. Nello studio [24] viene sfruttata questa pratica dimostrando come riesca a migliorare i risultati ottenuti per la classificazione dei discorsi d'odio relativi al caso di cyberbullismo.

### 2.2.5 Classificazione supervisionata e deep learning

Con il passare del tempo e il conseguente aumento della potenza di calcolo, diverse tecniche più complesse sono state applicate al riconoscimento dei discorsi d'odio: l'utilizzo di modelli ad apprendimento supervisionato quali Support Vector Machine (in [19]) e Naïve Bayes (in [32]), analizzando la rappresentazione vettoriale dei testi, costituivano fino a poco tempo fa lo standard per classificazioni di questo tipo [29].

Solo più di recente sono stati introdotti modelli basati su reti neurali profonde in grado di apprendere al meglio il contesto di una frase data in input: è il caso di [37] dove una rete neurale ricorrente è stata impiegata per il riconoscimento del testo. La caratteristica principale di questa rete è quella di “ricordare” parzialmente ciò che riceve in input, permettendo quindi una buona contestualizzazione di ogni parola nella frase che la racchiude.

Altri tentativi di classificazione sono stati effettuati con reti neurali convoluzionali (o CNN) e confrontati con le reti neurali ricorrenti: i risultati ottenuti in [22] dimostrano come, nonostante un'ottima capacità nel riconoscimento del contesto, le reti ricorrenti mantengano comunque una leggera superiorità in termini di punteggi ottenuti per il riconoscimento dell'hate speech online.

Attualmente lo stato dell'arte nella comprensione del linguaggio naturale è rappresentato dai modelli basati su reti di tipo transformer addestrati su corpus estremamente grandi: i loro punteggi in termini di precisione superano le alternative precedenti nella maggior parte dei dati analizzati. La numerosità degli studi che sfruttano reti di tipo transformer per identificare i discorsi d'odio online è ancora bassa data la recente diffusione di questi modelli ma, per i pochi già presenti come [9, 1], i risultati sembrano essere promettenti per questo genere di classificazione. Mancano tuttavia dei confronti che permettano di

comparare le performance ottenute dai modelli di tipo transformer con altri metodi di classificazione elencati precedentemente.

Passi avanti nella ricerca e nell'ottimizzazione dei modelli di questo tipo sono stati portati avanti principalmente da Google in [31] e da OpenAI in [5], grandi aziende e associazioni che hanno la possibilità tecnologica per poter addestrare modelli così estesi e onerosi di dati.

## 2.3 Strumenti tecnologici utilizzati

Il processo di text mining consente la trasformazione del testo non strutturato in un formato ordinato per una successiva schematizzazione e analisi della sintassi. Gli strumenti offerti da [40] permettono un'efficace scomposizione del testo e dei suoi contenuti, riuscendo opportunamente nella riduzione delle forme flessive o correlate di una parola alla loro forma base comune. A questo scopo esistono due principali tecniche: lo *stemming*, meno complesso della *lemmatizzazione* e per questo largamente più utilizzato come in [23, 46, 15], si riferisce ad un semplice processo euristico di troncamento dell'ultima parte della parola, auspicando di estrarre una sequenza che si avvicina il più possibile alla forma base. La lemmatizzazione è invece riferita all'analisi morfologica del termine per ottenerne, attraverso l'uso di un vocabolario, il lemma (o forma base) della parola originale.

La rappresentazione del testo è invece ben generalizzata dai modelli BERT presenti nella libreria di [8]. Il successivo compito di classificazione, come nel caso del riconoscimento dei discorsi d'odio, viene generalmente effettuato attraverso l'uso di tecniche di *transfer learning* (o *domain adaptation*): il *fine tuning* è sicuramente uno tra gli approcci più diffusi considerati i suoi vantaggi in termini di risparmio di tempo e risorse computazionali durante la fase di addestramento. Mediante il processo di fine tuning i pesi relativi alle connessioni tra i vari neuroni nei modelli vengono lievemente modificati rispetto alla loro versione originale, adattandoli ai dati presenti in input ma mantenendo comunque un tasso di apprendimento più basso rispetto alla normale fase di pre-addestramento. È ulteriormente possibile "congelare" alcuni layer e conseguentemente aggiungerne di altri di diverso tipo per provare ad ottenere diversi risultati in base alle caratteristiche degli

stessi layer aggiunti.

### 2.3.1 Il modello BERT

BERT, acronimo di Bidirectional Encoder Representations from Transformers [3], è un modello sviluppato e pre-addestrato da Google di tipo Transformer che adotta meccanismi di attenzione pesando ogni elemento fornito in input. A differenza delle reti neurali ricorrenti, largamente utilizzate nel riconoscimento del linguaggio naturale, le reti di tipo Transformer non necessitano di elaborare i dati in ordine ma riescono a fornire un contesto per ogni posizione presente nella sequenza data in input.

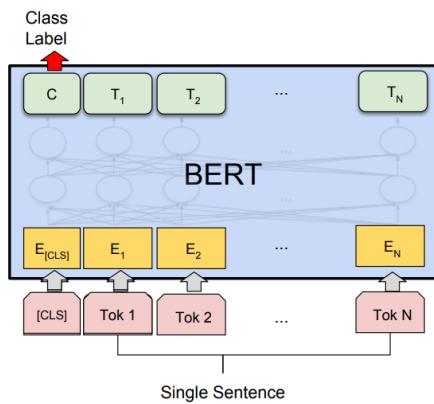


Figura 2.1 Single sentence classification task, BERT

Tra i diversi impieghi del modello BERT è possibile usare il token specialece *[CLS]* come illustrato in 2.1 per indicare il task di classificazione di una determinata frase. Dopo una prima fase di conversione delle parole in tokens il modello ne seleziona uno casualmente e, mascherandolo, cerca di predirlo autonomamente: questa procedura, chiamata dagli autori *Masked LM*, forza la rete nel mantenere una rappresentazione del contesto distribuita su tutto il corpo della frase e non solo sui token antecedenti e successivi rispetto a quello considerato. La rappresentazione finale del testo, ottenuta leggendo gli output dell'ultimo layer del modello, permette una classificazione efficace in una delle classi che meglio descrive la frase fornita in input.

# 3

## Framework proposto

### 3.1 Overview del sistema proposto

La ricerca svolta copre l'intero processo di raccolta e classificazione dei commenti partendo dalla raccolta dei dati fino al proponimento di diversi sistemi di riconoscimento dei discorsi d'odio. In figura 3.1 è proposto un diagramma di flusso descrittivo del percorso seguito durante lo studio.

La prima fase fondamentale, utile a raccogliere il maggior numero di dati, è l'estrazione dei commenti da TikTok; successivamente le informazioni raccolte devono essere pulite dagli elementi poco utili ai fini della classificazione come messaggi di spam o eventuali errori.

Dopodiché vengono applicati e analizzati due metodi di classificazione differenti: il primo ha come obiettivo la segnalazione di lemmi offensivi o denigratori all'interno dei commenti, il secondo invece prevede il fine tuning di un modello per il riconoscimento del linguaggio naturale.

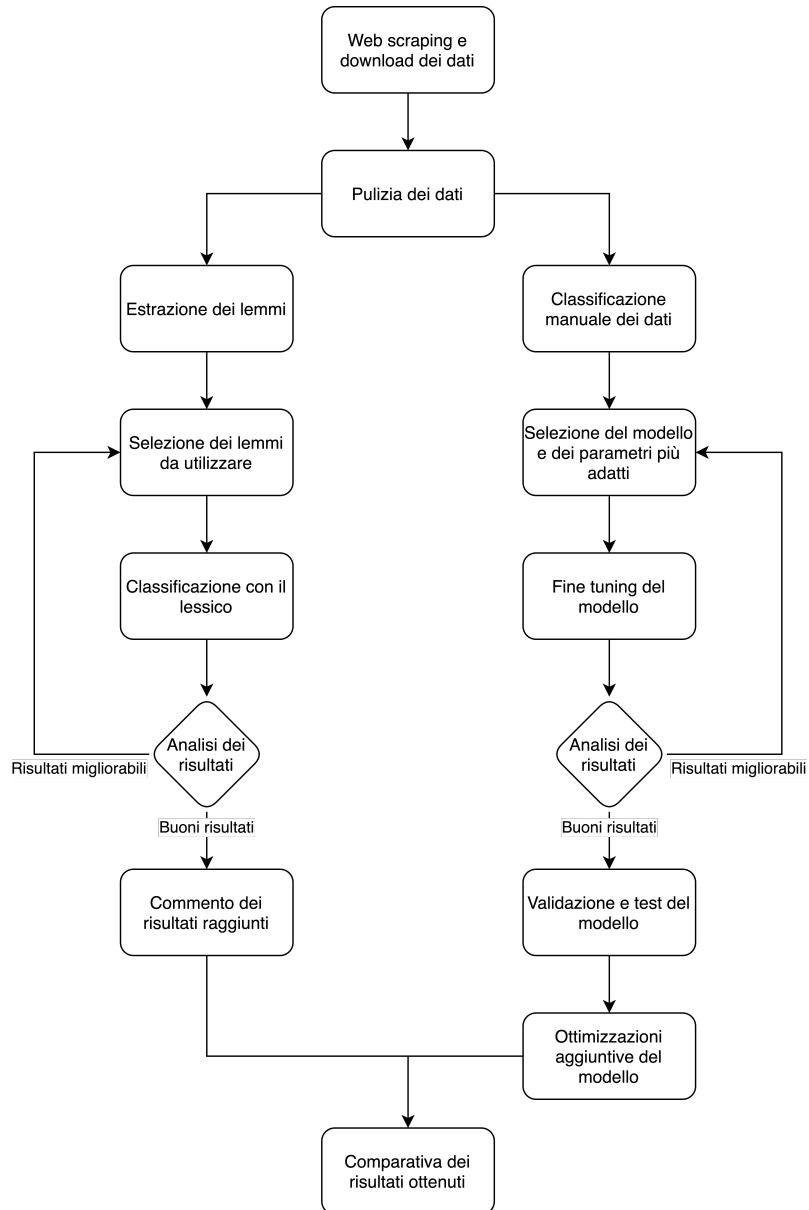


Figura 3.1 Diagramma di flusso riepilogativo del lavoro svolto

Il primo metodo di classificazione prevede dapprima l'estrazione dei lemmi da ogni commento scaricato e, successivamente, un confronto con un lessico di termini offensivi e denigratori. Una volta ottenuti i risultati sarà possibile effettuare una valutazione qualitativa degli stessi per giudicare quanto il metodo di classificazione sia stato preciso nel riconoscere i commenti positivi da quelli offensivi.

Il secondo metodo di classificazione previsto dallo studio comprende l'utilizzo di un modello in grado di comprendere il linguaggio naturale sul quale effettuare fine tuning.

Per questa fase è necessario classificare manualmente un sottoinsieme di commenti presi dai dati originariamente scaricati seguendo una serie di linee guida specificate in seguito. Come già visto per il metodo di classificazione precedente, viene effettuata una valutazione dei risultati in output adottando diverse configurazioni dei parametri della rete neurale profonda. Una volta trovata la migliore combinazione vengono svolte alcune ulteriori ottimizzazioni modificando la rete neurale per migliorarne ulteriormente le prestazioni.

Tutti i risultati ottenuti vengono infine messi a confronto per determinare quale sia stato il miglior metodo di classificazione trovato.

## 3.2 Raccolta e gestione dei dati

In seguito alle restrizioni sulla privacy degli utenti nei social network nel corso degli ultimi anni, il processo di estrazione dati non può più essere svolto utilizzando delle API ufficiali. Nella maggior parte dei casi sono presenti tuttavia delle API non ufficiali che permettono di ottenere informazioni relative a singoli post o ad un singolo account. Nel caso specifico di TikTok, social network nato solo in tempi recenti, non sono presenti né API ufficiali né API alternative che permettono l'estrazione delle informazioni per la costruzione dei dataset utili alla ricerca. Per questo motivo è stato necessario costruire un tool in Python in grado di scaricare tutti i commenti da ogni post di un determinato account.

I dati raccolti vengono gestiti ed analizzati con l'ausilio di librerie quali Pandas e NumPy, universalmente utilizzati in presenza di grandi quantità di informazioni.

### 3.2.1 Web scraping

Il Web scraping permette l'estrazione di dati da una pagina web simulando una navigazione di un normale utente. È stato utilizzato Selenium e BeautifulSoup rispettivamente per la simulazione della navigazione e la raccolta dei dati dalla struttura HTML della sezione commenti di ogni post. Si è reso altresì necessario adattare il più possibile il tool alle diverse configurazioni della pagina e ai vari sistemi di protezione che TikTok implementa per limitare la raccolta di dati automatica (Captcha in primis). Il processo per il download dei dati implica quindi l'apertura di un determinato post relativo ad un account, il caricamento

di tutti i commenti e infine l’analisi dell’albero DOM (una rappresentazione a oggetti della pagina web) per l’estrazione del commento e dei suoi metadati. Per ogni commento vengono salvati il nome, l’identificativo univoco relativo all’account che ha commentato, l’immagine di profilo (utile per una successiva classificazione del genere) e il numero di like presenti al commento stesso.

### 3.2.2 Pulizia dei dati

Prima dell’introduzione dei sistemi di identificazione dei discorsi d’odio ogni commento viene quanto più possibile pulito da ogni informazione accessoria non utile al processo di classificazione. Questa fase di pulizia dei dati è eseguita principalmente per minimizzare per quanto possibile la numerosità delle parole da analizzare e, di conseguenza, ottimizzare al meglio i tempi di calcolo richiesti nelle fasi successive. Per ogni commento vengono eliminati tutti i tag verso altri profili (e.g. @*nomeprofilo*) e caratteri non standard che potrebbero creare problemi nella fase di analisi. Vengono anche riconosciuti ed eliminati manualmente messaggi di spam che risultano poco utili nei vari passaggi di classificazione.

## 3.3 Analisi del lessico

Per una prima classificazione dei commenti scaricati viene utilizzato il lessico Hurtlex [13] contenente una raccolta di lemmi generalmente utilizzati nei commenti offensivi, aggressivi e denigratori. In particolare, ogni lemma appartiene ad una o più delle 17 categorie (quali ad esempio stereotipi etnici, disabilità fisiche, disabilità cognitive, omofobia, ecc.) e può essere classificato come conservativo o inclusivo. Nel primo caso il lemma ha un significato originale offensivo, nel secondo può assumere un significato offensivo solo se usato in determinati contesti. Si procede quindi con un confronto tra i lemmi che compongono i commenti scaricati e gli stessi riportati nel lessico, registrando la loro tipologia, la categoria di appartenenza e un identificativo univoco generato automaticamente.

### 3.3.1 Lemmatizzazione

I dati originali vengono ulteriormente puliti prima di affrontare la fase di lemmatizzazione: punteggiatura, emoji, sequenze non appartenenti all’alfabeto e tutte le stop words, ovvero parole poco significative al senso finale della frase, non sono presenti nel lessico e pertanto sono inconcludenti ai fini della classificazione. Il risultato del precedente processo viene successivamente analizzato da Stanza [40], una raccolta di modelli pre-addestrati dall’università di Stanford in grado di effettuare analisi sulla sintassi in varie lingue.

Per ogni parola di un commento viene estratto il lemma ottenendo la sua forma base e, successivamente, viene confrontato con tutto il lessico Hurtlex. Tutti i commenti sono dunque classificati come non offensivi se nessun lemma ha trovato corrispondenza nel lessico o, come offensivi, se i commenti presentano uno o più lemmi segnalati nel lessico come potenzialmente denigratori. Per ciascun commento sono specificati gli id univoci dei lemmi potenzialmente offensivi e un punteggio basato sul numero di occorrenze per ogni categoria.

## 3.4 Classificazione con BERT

Una seconda possibilità di classificazione è data dall’utilizzo di reti neurali profonde in grado di riconoscere e interpretare il linguaggio naturale dei commenti. Viene utilizzata la tecnica del fine tuning che consente di usare una rete neurale già addestrata per un task generico e, ritoccandone i pesi, specializzarla per un task più specifico.

Data la complessità dei calcoli da svolgere e vista la dimensione della rete neurale profonda utilizzata, è stato necessario utilizzare hardware dedicato in grado di effettuare computazione parallela su schede grafiche Nvidia. Tra le diverse alternative si è scelta la piattaforma Colab di Google che permette la computazione in edge computing gratuitamente seppur con alcune limitazioni trascurabili sull’hardware e sui tempi di calcolo.

### 3.4.1 Classificazione manuale e linee guida adottate

Durante il processo di fine tuning il modello necessita di dati etichettati da poter essere successivamente suddivisi in un dataset di train e un secondo dataset di test. È stato quindi necessario classificare manualmente un sottoinsieme di commenti originali in due classi: positivi e negativi. In questa fase delicata si è scelto di conservare quanto più possibile la tipologia di linguaggio e il gergo comune utilizzato su TikTok definendo delle linee guida da tenere a mente durante il processo di classificazione.

La classe di commenti positivi comprende non solo commenti generici ma anche quelli che mirano ad una difesa dell'influencer attaccato/a dai commenti classificati come negativi. Questa scelta rende sicuramente più difficile la fase di apprendimento visto l'utilizzo di un lessico molto simile tra commenti difensivi, classificati positivamente, e quelli offensivi. Per descrivere al meglio la tipologia di classificazione appena esposta è di seguito fornito un esempio esplicativo a riguardo: "*Wow mi hai migliorato la giornata*" e "*Non ascoltare chi ti da della stupida*", sono entrambi classificati come commenti positivi mentre "*Sei stupida e dovresti vergognarti*" è stato classificato come offensivo.

La numerosità dei commenti negativi è risultata essere inferiore a quella dei commenti positivi a causa della moderazione effettuata già dal social network stesso al momento della pubblicazione. In totale sono stati classificati più di 2300 commenti di cui, circa il 15% sono offensivi.

### 3.4.2 Scelta del modello e fine tuning

La scelta del modello è sicuramente vincolata dalla lingua utilizzata nei commenti. Tra i diversi modelli a disposizione sulla piattaforma Huggingface [8] ne sono stati selezionati due per la lingua italiana: *BERT base italian uncased* e *BERT base italian xxl cased*: entrambi sono stati precedentemente addestrati dai laboratori di ricerca Google sul testo integrale di Wikipedia e su un corpus composto da articoli e testi ottenuti da pubblicazioni online. La principale differenza tra i due riguarda la dimensione del dataset utilizzato nella fase di addestramento iniziale e la capacità di saper riconoscere i caratteri maiuscoli da quelli minuscoli. Sono stati considerati anche modelli capaci di riconoscere più lingue ma

non verranno presi in considerazione nella fase sperimentale vista la loro bassa performance per la sola lingua italiana.

### 3.4.3 Costruzione della rete neurale con BERT

Per poter controllare al meglio ogni parametro della rete neurale è stata scelta la libreria PyTorch [30], ampiamente utilizzata nel mondo della ricerca per applicazioni riguardanti la visione digitale e l'elaborazione del linguaggio naturale.

Dopo aver diviso il dataset prodotto dalla precedente fase di classificazione manuale ogni commento viene scomposto in tokens interpretabili dal primo layer di input del modello BERT. Più precisamente viene stimata una lunghezza massima fissata di 128 caratteri per commento anche se, mediamente, la lunghezza è di molto inferiore. Una volta generati gli input, ed effettuate le dovute conversioni in tensori, viene costruita una rete neurale semplice in grado di interpretare l'output del modello BERT selezionato precedentemente. La scelta dei parametri prende spunto dai consigli forniti dagli autori del modello stesso, sono state tuttavia apportate delle leggere modifiche al numero di epoch previste e alla dimensione di batch per cercare la combinazione che portasse al risultato migliore.

### 3.4.4 Ottimizzazioni effettuate sulla rete neurale

Ottenuti i risultati dai due modelli BERT citati precedentemente si è scelto di ottimizzare al meglio la rete neurale per cercare di aumentare il più possibile la capacità di classificazione dei commenti offensivi e non. Vengono quindi effettuate una serie di modifiche eliminando l'ultimo layer da BERT dedicato alla classificazione (chiamato *pooler layer*) in modo da aumentare la rete neurale di base con dei layer aggiuntivi, la cui scelta è influenzata dalle caratteristiche peculiari di quest'ultimi.

L'elaborazione del linguaggio naturale, prima dell'arrivo dei modelli basati su reti di tipo transformer, era affidata alle reti neurali ricorrenti in grado di "ricordare", seppur in forma parziale, le informazioni date come input. Cercando di sfruttare questa caratteristica viene quindi introdotto un layer aggiuntivo di tipo Bi-LSTM, acronimo di *Bi-directional long short term memory*. Lo scopo è quello di ottenere una rappresentazione eseguita dal

modello BERT della frase in input e sfruttare le caratteristiche di memoria di una rete neurale ricorrente. Come anticipato si è scelto di utilizzare un layer di tipo Bi-LSTM che, grazie alla sua bidirezionalità, permette di accedere in ogni istante alle informazioni precedenti e successive riuscendo quindi a conservare il lavoro svolto da BERT nella rappresentazione del testo in input contemporaneamente rafforzando i pesi delle parole adiacenti a quella considerata.

Una seconda opzione per migliorare le prestazioni è data dall'aggiunta di un layer lineare in coda al modello BERT. Questa tecnica permette di specializzare la rete per un task ancora più specifico migliorando quindi la rappresentazione del testo già effettuata da BERT. In seguito ad alcuni test è stato necessario inserire anche un layer di regolarizzazione in grado mitigare il fenomeno di overfitting. Verranno riportati i risultati ottenuti con un livello aggiuntivo di Dropout in grado di eliminare alcuni nodi presi in maniera casuale con una probabilità fissata dello 0.2%.

# 4

## Risultati sperimentali

### 4.1 Classificazione con il lessico Hurtlex

I primi risultati ottenuti sono relativi alla classificazione con il lessico Hurtlex. Il lessico, composto da quasi 7000 lemmi molti dei quali inclusivi, riesce solo in parte a classificare correttamente i commenti in base alla loro categoria di appartenenza. Vengono quindi svolte alcune operazioni in grado di mitigare, anche se solo parzialmente, questo problema.

#### 4.1.1 Distribuzione delle categorie ed esempi descrittivi

Come descritto precedentemente ogni lemma contenuto nel lessico appartiene a una o più categorie. Tra le 17 categorie quella che ha una frequenza maggiore è quella relativa a *parole dispregiative* seguita da termini relativi alla categoria *omosessualità*. Tutte le altre categorie mantengono una frequenza che si attesta sullo stesso livello.

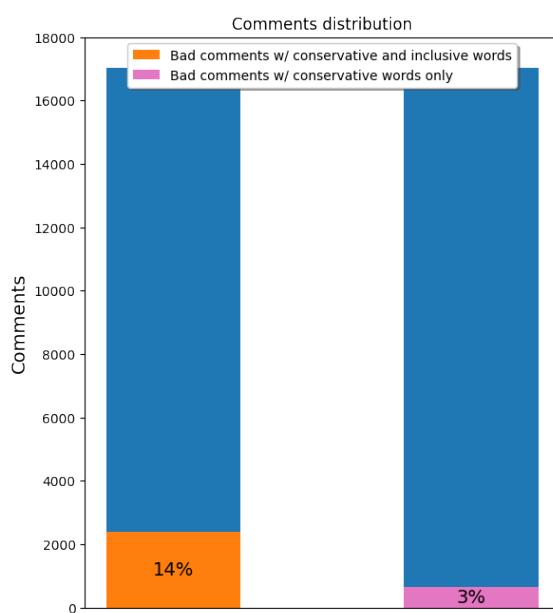
Osservando i commenti, nello specifico quelli che presentano più segnalazioni, è possibile notare come il lavoro svolto dalla classificazione con il lessico Hurtlex non sia

molto efficace nel trovare commenti con un senso offensivo. Nella maggior parte dei casi è possibile trovare commenti che contengono parole considerate offensive dal lessico ma che nella realtà sono applicate ad un contesto totalmente diverso da quello atteso. I falsi positivi risultano quindi essere gran parte dell'output prodotto dalla classificazione con il lessico Hurtlex.

Di seguito vengono riportati degli esempi esplicativi su quanto affermato in precedenza: nel commento "*Sono cretina perché ancora non ti seguivo PS ti adoro*" il lemma segnalato è *cretina* mentre invece in "*sei bravissima e adoro il tuo stile faccio hip hop anche io, vorrei un sacco averti come insegnante ahaha*" viene segnalato il lemma *insegnante*; in entrambi i casi però i commenti risultano essere totalmente inoffensivi.

#### 4.1.2 Distinzione tra lemmi conservativi e inclusivi

Per cercare di arginare il problema relativo alla segnalazione di commenti inoffensivi si è provato ad eseguire l'algoritmo di classificazione sfruttando solamente i termini conservativi che, a differenza dei termini inclusivi, assumono nella maggior parte dei contesti significati prettamente offensivi. In questo caso il numero di commenti segnalati è più basso rispetto alla classificazione precedente come osservabile dal grafico in figura 4.1.



*Figura 4.1 Distribuzione dei commenti classificati come negativi usando lemmi conservativi e inclusivi o esclusivamente lemmi conservativi*

Nonostante l'utilizzo di lemmi solamente conservativi, la classificazione con il lessico non risulta comunque in grado di separare efficacemente i commenti offensivi da quelli innocui. I commenti segnalati quindi sono del tutto simili a quelli visti negli esempi precedenti.

Dopo una veloce ispezione visiva dei commenti in output è possibile notare come, nonostante vengano selezionati commenti con parole offensive, spesso il contesto cambia il senso della frase. In altri casi invece i lemmi classificati come conservativi non hanno molto a che fare con contesti di offesa come è evidente da alcuni esempi che seguono: "*Io sono due persone diverse praticamente*" e "*Ma adesso che sono chiuse le discoteche come fai a lavorare? (Sembra una domanda aggressiva, ma sono solo curiosa)*" sono stati segnalati per la presenza del lemma *diverso* nel primo commento e *curioso* nel secondo. Inoltre è importante indicare come i commenti di difesa verso il/la creator di contenuti su TikTok vengano comunque segnalati come offensivi: nel commento "*Ma che cafoni ignoranti! Sei bellissima!*" la presenza dei lemmi *cafone* e *ignorante* classifica automaticamente lo stesso come potenzialmente denigratorio.

In generale, tra tutti i commenti segnalati, sono sicuramente presenti dei commenti offensivi ma, nonostante questo, il numero di falsi positivi rende la classificazione con il lessico Hurtlex poco efficace nell'isolare solamente i commenti negativi.

## 4.2 Analisi esplorativa dei dati

Una volta ottenuto un semplice metodo di classificazione dei commenti è possibile esplorare superficialmente le informazioni raccolte per riassumerne le principali caratteristiche. La numerosità del dataset complessivo supera i 250 000 elementi e, per ognuno, sono state registrate non solo le informazioni relative al corpo del commento ma anche i relativi metadati come le informazioni di base sul profilo proprietario e la data di pubblicazione dello stesso. Di seguito vengono quindi analizzati questi aspetti utili ad avere un quadro più chiaro sui dati a disposizione.

### 4.2.1 Analisi di genere e movimenti temporali

Una prima considerazione viene svolta sulla distribuzione di genere di chi commenta. TikTok purtroppo non rende pubbliche le informazioni relative all'identificazione di genere di un account ed è stato quindi necessario classificare manualmente un campione di circa 10 000 account presi casualmente in base alla loro immagine profilo e ai contenuti caricati sulla piattaforma. La distribuzione risultante dall'analisi è osservabile nel grafico in figura 4.2 dove è evidente come la presenza femminile sia di molto superiore rispetto alla controparte maschile.

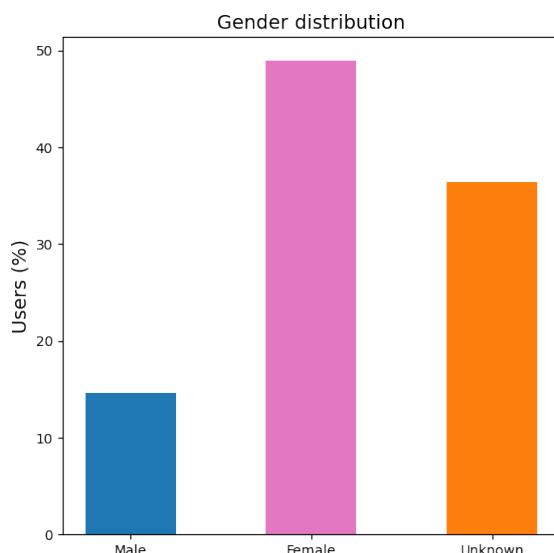


Figura 4.2 Distribuzione di genere di un campione di 10 000 account

Viene ulteriormente considerata la quantità di commenti segnalati durante il corso del tempo. Dopo aver collegato ogni commento al relativo post di appartenenza e conseguentemente diviso i dati in 10 regioni temporali, è stato possibile notare come la presenza di commenti negativi per ogni account si concentri soprattutto in presenza di post controversi che, diventando virali, attirano nuovi utenti che commentano negativamente. I picchi relativi a questo fenomeno sono osservabili nella figura 4.3 suddivisi per i quattro account più popolari analizzati.

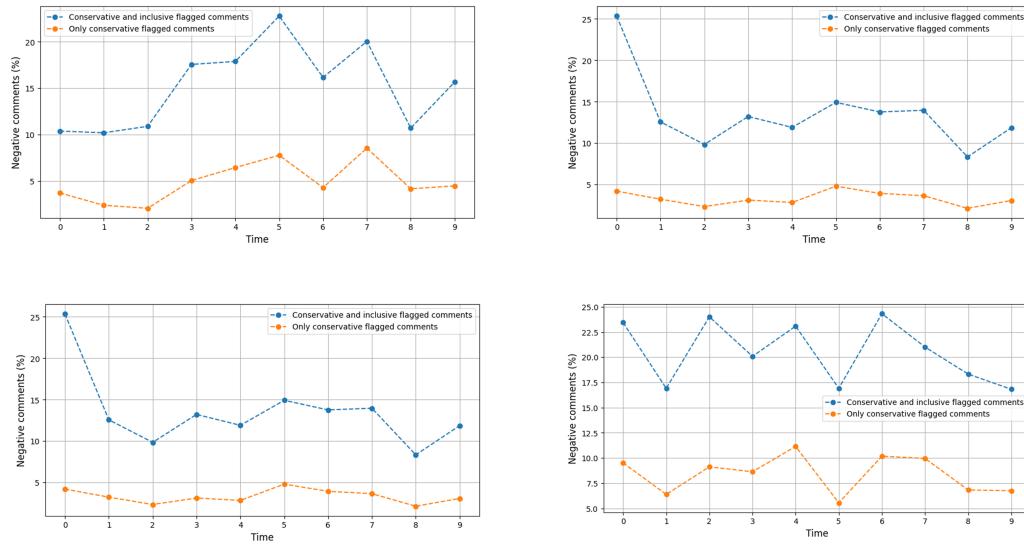


Figura 4.3 Movimenti temporali dei quattro account più popolari analizzati

## 4.3 Fine tuning del modello BERT base

Vengono proposti di seguito i risultati ottenuti dal processo di fine tuning dei modelli BERT descritti nel capitolo relativo al framework proposto. Per misurare la performance dei modelli e delle ottimizzazioni effettuate durante la fase di addestramento viene considerato l’output della funzione *cross-entropia*, una funzione obiettivo (*loss function*) che permette di avere una rappresentazione efficace sullo stato di apprendimento del modello. In seguito, per ogni epoca di addestramento effettuata, vengono considerate metriche quali F1, precisione e recupero osservando le rispettive matrici di confusione. L’accuratezza dei modelli non viene considerata ai fini della descrizione delle performance visto l’impiego di un dataset sbilanciato per la fase di addestramento e di test.

### 4.3.1 Valutazione della funzione obiettivo

La valutazione della funzione obiettivo durante la fase di addestramento è essenziale per comprendere come il modello sta apprendendo le informazioni dai dataset di train e di test. Osservando i dati in output è possibile stabilire se la rete neurale si adatta eccessivamente ai dati di train. Questo fenomeno è definito come overfitting e può portare

a basse performance per il dataset di test mantenendo allo stesso tempo delle ottime performance nel dataset di addestramento.

La prima fase di test è stata effettuata variando lievemente i parametri consigliati dagli autori dei modelli basati su BERT per cercare quale configurazione di adattasse al meglio al task di classificazione considerato. Il training consiste in otto epochi totali dove, per ogni epoca, viene registrato l'output della loss function sia sul dataset di train che su quello di test. Viene inoltre considerata anche la lunghezza di un singolo batch facendola variare tra 16, 32 e 64. I risultati del primo modello *BERT base italian uncased* vengono mostrati in figura 4.4.

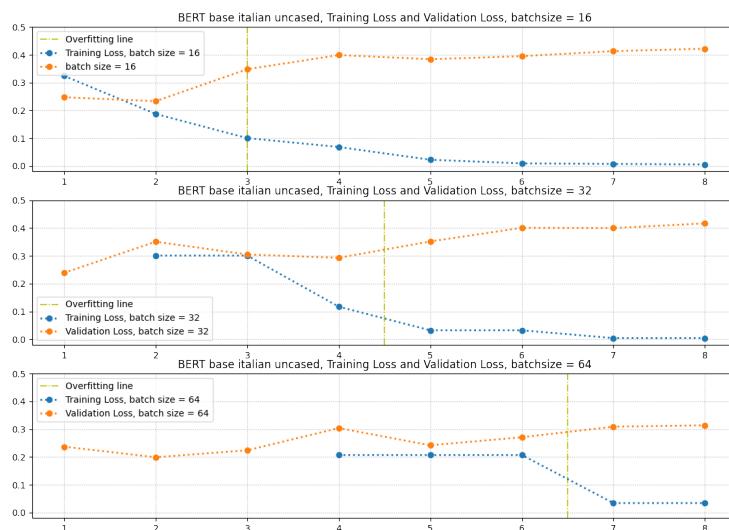


Figura 4.4 BERT base italiano uncased; risultati funzione obiettivo per il dataset di train e di test

I grafici ottenuti sono suddivisi secondo la lunghezza del batch utilizzata nella fase di addestramento. Come si può notare, osservando la crescita della funzione obiettivo per il dataset di test e la decrescita della stessa per il dataset di train, la probabilità di overfitting aumenta considerevolmente con l'aumentare delle epochi. Viene quindi rappresentata, attraverso l'utilizzo di una linea gialla, il limite entro il quale è consigliabile rimanere per il numero di epochi di addestramento necessarie al modello prima di incorrere in problemi di overfitting. I parametri ottimi ottenuti coincidono perfettamente con quelli consigliati dagli autori che suggeriscono un numero di epochi comprese nell'intorno di 4.

Il secondo modello utilizzato è *BERT base italian xxl cased*. A differenza del modello visto precedentemente viene utilizzato un corpus maggiore nella fase di pre-training (un incremento di circa il 17%) e viene aggiunta la capacità di riconoscere i caratteri minuscoli da quelli maiuscoli. L'output della funzione obiettivo è rappresentata in figura 4.5.

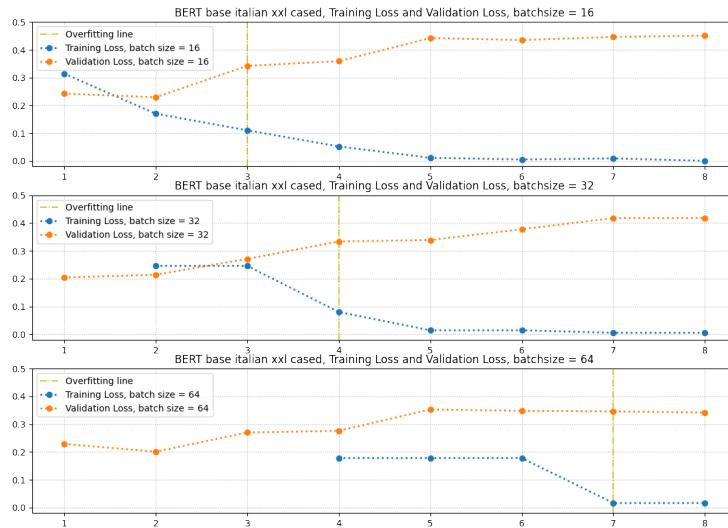


Figura 4.5 *BERT base italian xxl cased*; risultati funzione obiettivo per il dataset di train e di test

In entrambi i casi i risultati ottenuti riguardo il processo di apprendimento risultano essere molto simili. A seguito di ulteriori prove, aumentando fino a venti il numero di epoche, è possibile generalizzare l'andamento osservato: la probabilità di overfitting si presenta dopo un numero sempre maggiore di epoche proporzionalmente alla dimensione di batch.

### 4.3.2 Metriche sul dataset di test

La valutazione del modello, come accennato precedentemente, è stata effettuata sul dataset di test dove ogni commento presente non è mai stato utilizzato per l'apprendimento nelle fasi precedenti. La scelta delle metriche è condizionata da alcuni fattori fondamentali dovuti alla distribuzione delle diverse classi nel dataset stesso. Nello specifico, avendo a disposizione un dataset sbilanciato, non è possibile usare l'accuratezza per descrivere al meglio il comportamento del modello nella fase di classificazione. Prendendo ad esempio il dataset a disposizione con solamente il 15% di commenti negativi, il modello potrebbe classificare tutti i commenti come positivi e riuscire a raggiungere comunque un livello di

accuratezza pari ad un 85% senza in realtà classificare correttamente nessun commento negativo.

La scelta della metrica ricade quindi sull'F1 score in grado di misurare l'accuratezza di un modello anche in caso di dataset sbilanciato. L'F1 score considera contemporaneamente sia la precisione che il recupero, due metriche utili a capire rispettivamente quanti commenti sono stati classificati correttamente sul totale dei commenti etichettati con una classe dal modello e quanti commenti sono stati correttamente classificati sul numero totale di commenti effettivamente appartenenti a quella classe. In termini più formali la precisione rappresenta il rapporto tra i veri positivi e la somma tra i veri positivi e i falsi positivi; il recupero è invece il rapporto tra i veri positivi e la somma dei vero positivi con i falsi negativi. Il punteggio F1 viene infine calcolato con la media armonica tra precisione e recupero definito di seguito:

$$\text{Precisione} = \frac{VP}{VP + FP}$$

$$\text{Recupero} = \frac{VP}{VP + FN}$$

$$F_1 = \frac{2}{\frac{1}{R} + \frac{1}{P}} = 2 \cdot \frac{P \cdot R}{P + R}$$

Le metriche sopra descritte sono rappresentate in figura 4.6 per il modello *BERT base italian uncased* e in figura 4.7 per il modello *BERT base italian xxl cased*.

Epoch	BERT base italian uncased			BERT base italian xxl cased		
	16 batch	32 batch	64 batch	16 batch	32 batch	64 batch
1	0.571	0.601	0.613	0.606	0.660	0.666
2	0.699	0.662	0.678	0.756	0.736	0.716
3	0.686	0.625	0.703	0.713	0.716	0.690
4	0.714	0.743	0.691	<b>0.760</b>	0.752	0.700
5	0.746	0.705	<b>0.758</b>	0.735	0.737	0.699
6	0.758	0.720	0.722	0.743	0.737	0.711
7	0.754	0.736	0.754	0.735	0.728	0.733
8	0.733	0.720	0.743	0.735	0.724	0.717

Tabella 4.1 Risultati numerici per ogni epoca usando diverse dimensioni di batch

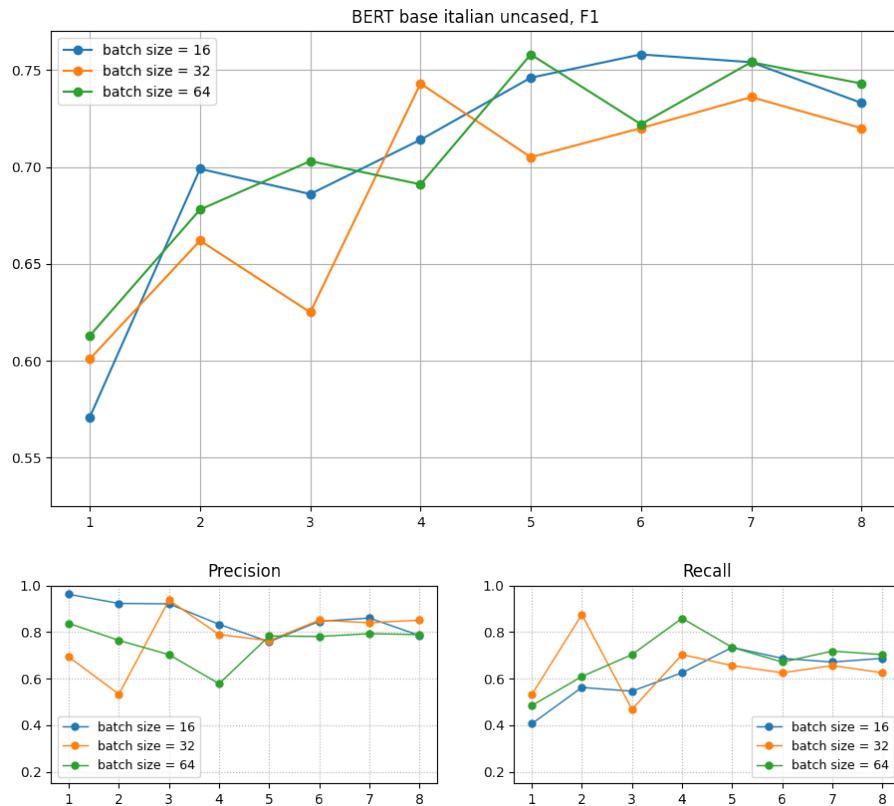


Figura 4.6 BERT base italian uncased; F1 score, precisione e recupero

Confrontando i risultati ottenuti dai due modelli emerge una leggera superiorità per quanto riguarda il punteggio F1 ottenuto dal modello più grande *BERT base xxl cased*. Osservando i valori numerici riportati in tabella 4.1 sono evidenziati i punteggi F1 più alti ottenuti dai due modelli. Se si considera anche il numero di epoche, cercando di preferire il minor numero di iterazioni per evitare overfitting e abbassare i tempi di calcolo, il modello *xxl* si conferma come il migliore tra i due considerati.

La libreria PyTorch permette il salvataggio dei pesi trovati durante il processo di fine tuning del modello. Ricostruendo la rete neurale e importando i pesi precedentemente calcolati è possibile svolgere alcuni test manuali per osservare da più vicino il comportamento del modello nei diversi scenari di utilizzo. Le frasi più banali, la cui classe di appartenenza è ovvia anche con un semplice controllo del lessico, vengono classificate correttamente senza problemi mantenendo una probabilità restituita dalla funzione sigmoidea generalmente superiore al 85%. Per mettere in difficoltà il modello è quindi necessario usare frasi che condividono lo stesso tipo di linguaggio ma con un significato diametralmente

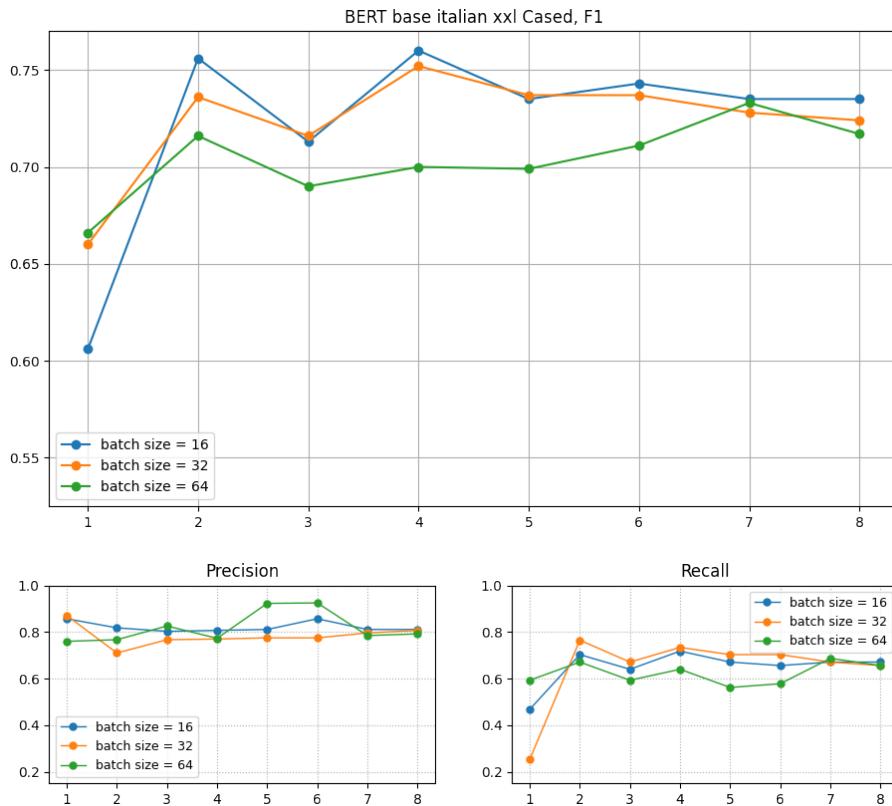


Figura 4.7 BERT base italiano xxl cased; F1 score, precisione e recupero

opposto. L'esempio riportato nella tabella 4.2 mostra il risultato restituito dal modello in entrambi i casi. Nel secondo esempio è evidente come BERT non svolga solo un ruolo di classificazione basato sui termini utilizzati ma riesce efficacemente a comprendere il loro contesto di utilizzo interpretando correttamente il senso della frase. Viene ulteriormente segnalato come il problema dell'unintended bias nella classificazione dei discorsi d'odio (presentato e studiato in [28]) risulta essere efficacemente gestito dalla rete neurale testata.

Rimane comunque importante sottolineare che la buona precisione riscontrata negli esempi riportati è dovuta alla forte presenza di termini simili nel dataset di train che permettono quindi un apprendimento efficace del modello stesso. Lo stesso comportamento non è sempre osservabile se si utilizzano vocaboli non presenti nel dataset di train dimostrando come, soprattutto quando si parla di reti neurali profonde, una grande quantità di dati risulta essere indispensabile per ottenere un modello che riesca a mantenere delle buone prestazioni anche nel mondo reale.

	Commento	Classe	Output funzione sigmoidea
1	<i>Sei bravissima, complimenti</i>	Good	0.06
	<i>Sei bruttissimo, vergognati</i>	Offensive	0.97
2	<i>Dai, povero...</i>	Good	0.38
	<i>Sei un povero</i>	Offensive	0.78

Tabella 4.2 Esempi di classificazione manuale per verificare la comprensione del contesto del modello BERT

## 4.4 Modifiche al modello BERT base

I risultati ottenuti con l'utilizzo del modello base di BERT sono più che soddisfacenti per un semplice task di classificazione ma è possibile migliorare il risultato ottenuto aggiungendo dei layer che, invece di considerare l'output della classificazione (o pooler layer), raccolgono l'output dell'ultimo layer di BERT (768 nodi) e continuano il processo di fine-tuning.

Come accennato nel capitolo relativo framework proposto vengono proposte due modifiche diverse al modello BERT di base. La prima riguarda l'aggiunta di un layer di tipo Bi-LSTM in grado di memorizzare gli input antecedenti e seguenti, caratteristica peculiare delle reti neurali ricorrenti; la seconda riguarda l'aggiunta di un layer lineare, uno di dropout e infine un ultimo layer di output in grado di classificare nelle due classi gli input ricevuti.

Come banchmark per il confronto delle prestazioni viene scelto il modello con i migliori risultati trovati nei capitoli precedenti: *BERT base italian xxl cased* con una dimensione di batch pari a 16.

### 4.4.1 BERT con layer Bi-LSTM

La prima modifica apportata al modello è stata aggiungere un layer Bi-LSTM in coda a BERT. Durante la fase di addestramento, osservando i risultati della funzione obiettivo per il dataset di train e di test in figura 4.8, si può notare come i valori della loss function sul dataset di test aumentino dopo ogni epoca segnalando la probabile occorrenza di un problema di overfitting. Contemporaneamente la performance del modello, rappresentata dalla spezzata a destra, misurata dal punteggio F1 è in leggera discesa ma riesce a mantenere

comunque dei valori in linea, o di poco inferiori, rispetto a quelli visti dal modello base di BERT.

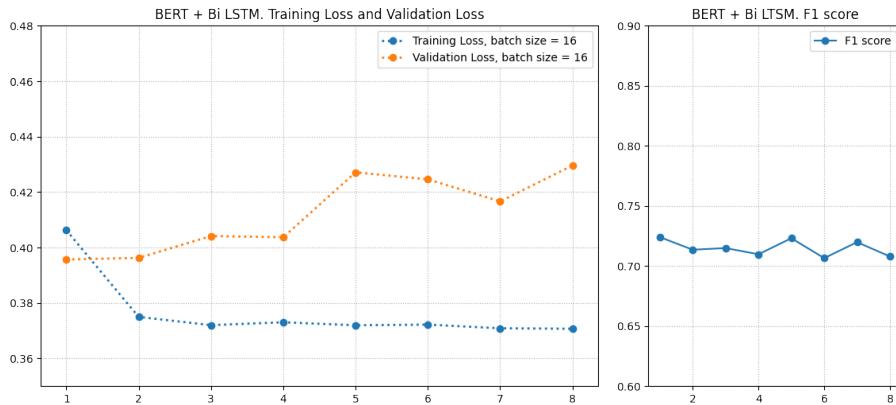


Figura 4.8 Metriche del modello BERT base con un layer Bi-LSTM in coda

A fronte di una validation loss maggiore e un punteggio F1 quasi identico al modello base di BERT è possibile concludere che questa modifica effettuata, almeno in questo caso, non porta a nessun vantaggio effettivo rispetto all'impiego del modello base. Questo comportamento è probabilmente attribuibile alla struttura delle reti neurali ricorrenti spiegate nel capitolo relativo al framework proposto. La loro capacità di ricordare è ristretta ai token vicini (destra e sinistra nel caso di una LSTM bidirezionale) e risulta essere quindi limitante rispetto all'ottima capacità di encoding di un'intera frase da parte delle reti neurali di tipo transformer di cui BERT fa parte.

#### 4.4.2 BERT con layer lineare, dropout e di classificazione

In figura 4.9 vengono rappresentati i risultati ottenuti dalla seconda modifica effettuata al modello base di BERT. In questo caso sono state effettuate diverse prove prima di ottenere il miglior risultato possibile. Inizialmente era stato utilizzato un singolo layer denso dimostratosi però inefficace nel migliorare le performance a causa del presentarsi di overfitting già dalla prima epoca. Si è quindi scelto di introdurre un layer di regolarizzazione, più nello specifico di dropout, per cercare di mitigare il problema di overfitting e subito dopo un semplice layer di classificazione in grado di restituire valori compatibili con le due classi dei commenti. Dopo diversi tentativi si è trovata la miglior percentuale di dropout fissata a 0.2.

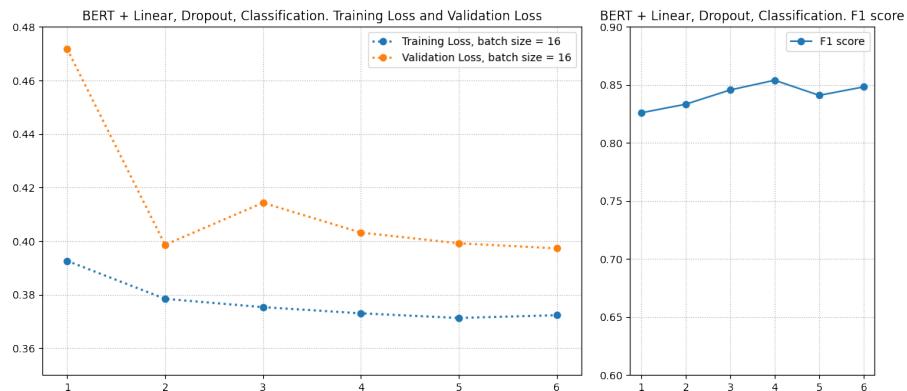


Figura 4.9 Metriche del modello BERT base con un layer lineare e uno di dropout in coda

I dati di classificazione ottenuti dimostrano come, nonostante una validation loss più alta della training loss in fase di training, le metriche sul dataset di test rimangano più che accettabili nel corso delle epoche. È possibile supporre che, in presenza di un dataset più grande nella fase di addestramento, i risultati della funzione obiettivo sarebbero stati sicuramente migliori: la rete neurale, con anche l'aggiunta dei nuovi layers visti precedentemente, risulta essere troppo grande per una quantità di dati così bassa.

#### 4.4.3 Confronto risultati con le modifiche effettuate

Viene infine proposto un grafico riassuntivo in figura 4.10 dove vengono illustrati i diversi punteggi F1 ottenuti sia dal modello base di BERT sia dai modelli con le modifiche apportate. Tra tutti i modelli proposti il migliore risulta essere BERT con i layer lineari e di dropout aggiunti in coda. Come anticipato il modello BERT base e il modello BERT con l'aggiunta di un layer Bi-LSTM si mantengono sulle stesse performance con un leggero vantaggio per il secondo a fronte però di una validation loss più alta vista precedentemente.

Nella tabella riepilogativa 4.3 sono vengono ulteriormente riportati i punteggi F1 relativi alla prima classificazione con il lessico Hurtlex. Come previsto da una prima ispezione manuale, il lessico non è stato in grado di classificare efficacemente i commenti offensivi ottenendo un punteggio decisamente più basso dei modelli analizzati.

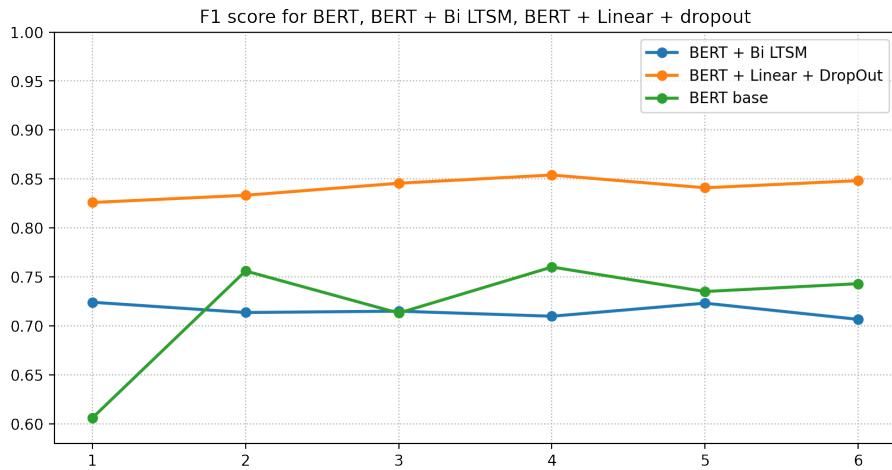


Figura 4.10 Risultati ottenuti dal miglior modello BERT base e dai modelli modificati con i layer aggiuntivi

Epoch	BERT base	BERT Bi-LSTM	BERT Linear dropout	Hurtlex	Hurtlex only conservative
1	0.606	0.724	0.825	-	-
2	0.756	0.713	0.833	-	-
3	0.713	0.714	0.845	-	-
4	0.760	0.709	<b>0.853</b>	-	-
5	0.735	0.723	0.840	-	-
6	0.743	0.706	0.848	-	-
	-	-	-	0.121	0.154

Tabella 4.3 Tabella riepilogativa dei migliori risultati numerici ottenuti con tutti i metodi di classificazione utilizzati

# 5

## Conclusioni e sviluppi futuri

### 5.1 Conclusioni sul lavoro svolto

La ricerca presenta una serie di valide proposte per la classificazione di commenti relativi ai discorsi d'odio online: i risultati ottenuti mostrano diverse strade percorribili ognuna con i propri vantaggi e svantaggi. È stato preso in analisi TikTok, un social network nato solo di recente, le cui dinamiche differiscono completamente da quelle tradizionali di piattaforme più conosciute e consolidate come Instagram, Twitter e Facebook.

Dopo una prima fase di raccolta di dati, diversi sistemi di classificazione sono stati messi a confronto partendo dalla semplice analisi del lessico, fino ad arrivare all'uso di modelli rappresentanti lo stato dell'arte nel riconoscimento del linguaggio naturale. Per la fase di addestramento della rete neurale è stato generato un dataset di commenti classificati manualmente definendo delle linee guida che permettessero al modello di apprendere la differenza tra un commento positivo e uno offensivo. È stata ulteriormente enfatizzata la differenza tra i commenti difensivi e offensivi che, pur condividendo gran parte del

lessico, sono state distinte due classi diverse: questa scelta ha sicuramente influenzato le prestazioni del modello, il quale però ha dimostrato una buona capacità nel comprendere il contesto ivi una singola parola è inserita.

È stato successivamente applicato un processo di fine tuning che ha permesso la specializzazione della rete neurale in un task specifico riuscendo a migliorare di molto la precisione della classificazione. Una volta effettuati i diversi test ed aver trovato la combinazione di parametri migliori è stato altresì svolto un lavoro di ottimizzazione aggiungendo diversi tipi di layer dopo BERT che permettessero una maggiore specializzazione nel compito da svolgere. Tra le due prove effettuate solo la seconda, con l'aggiunta di layers lineari e di regolarizzazione, ha ottenuto prestazioni superiori, migliorando a tutti gli effetti la performance peraltro già molto buona di BERT. La prima prova invece, effettuata aggiungendo un layer di tipo Bi-LSTM, ha mantenuto delle performance quasi al pari di BERT per via dei suoi aspetti tecnici analizzati nei capitoli precedenti.

## 5.2 Possibili miglioramenti proposti e sviluppi futuri

È possibile estrapolare dal lavoro svolto alcuni possibili miglioramenti.

In primis, come accennato durante l'analisi dei risultati delle modifiche apportate a BERT, si è dimostrata necessaria l'analisi di un dataset dalle dimensioni maggiori. Le reti neurali profonde, come quelle utilizzate per la classificazione vista nei capitoli precedenti, necessitano di una grande quantità di dati per essere funzionali e mantenere le buone performance in un contesto reale, prescindendo da una classica fase di validazione.

Inoltre, con una maggiore quantità di dati, sarebbe possibile effettuare nuovi esperimenti partendo dalla base delle modifiche proposte o utilizzare nuovi modelli di base che migliorano le già ottime prestazioni di BERT (e.g. RoBERTa sviluppatto dal Facebook AI in [36]). Il problema principale, in entrambi i casi, è rappresentato dal fenomeno di overfittig che ha limitato la possibilità di ampliare ulteriormente la rete neurale o sfruttare appieno le caratteristiche intrinseche dei vari layer aggiunti.

Ultimo possibile miglioramento è rappresentato dal numero di features considerato dalla rete neurale. Nello studio riportato in tesi il modello riceve in input solamente il testo

del commento, tuttavia viene ignorato il contesto entro il quale lo stesso è stato pubblicato. Di conseguenza in molti casi conoscere delle informazioni aggiuntive relative al post di provenienza o il commento principale in caso di una risposta a cascata, aiuterebbe nel compito di classificazione, fornendo ulteriori dettagli come base di partenza.

# References

- [1] Gaurav Rajput - Narinder Singh punn - Sanjay Kumar Sonbhadra - Sonali Agarwal. *Hate speech detection using static BERT embeddings*. URL: <https://arxiv.org/pdf/2106.15537.pdf>.
- [2] Swati Agarwal e Ashish Sureka. *Characterizing Linguistic Attributes for Automatic Classification of Intent Based Racist/Radicalized Posts on Tumblr Micro-Blogging Website*. 2017. arXiv: 1701.04931 [cs . IR].
- [3] Ashish Vaswani - Ming-Wei Chang - Kenton Lee - Kristina Toutanova - Google AI e Google Brain. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. URL: <https://arxiv.org/pdf/1810.04805.pdf>.
- [4] Sarah Hewitt - Thanassis Tiropanis - Christian Bokhove. *The problem of identifying misogynist language on Twitter (and other online social spaces)*. URL: <https://dl.acm.org/doi/pdf/10.1145/2908131.2908183>.
- [5] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs . CL].
- [6] Chikashi Nobata - Joel Tetreault - Achint Thomas - Yashar Mehdad - Yi Chang. *Abusive Language Detection in Online User Content*. URL: <https://dl.acm.org/doi/pdf/10.1145/2872427.2883062>.
- [7] Ying Chen et al. “Detecting Offensive Language in Social Media to Protect Adolescent Online Safety”. In: *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*. 2012, pp. 71–80. DOI: 10.1109/SocialCom-PASSAT.2012.55.
- [8] Huggingface community. *State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0*. URL: <https://huggingface.co/transformers/index.html>.
- [9] Marzieh Mozafari - Reza Farahbakhsh - Nöel Crespi. *A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media*. URL: <https://arxiv.org/pdf/1910.12574.pdf>.
- [10] CrowdFlower. *Data for everyone*. URL: <https://www.crowdflower.com/data-for-everyone/>.
- [11] Fabio Del Vigna et al. “Hate me, hate me not: Hate speech detection on Facebook”. In: gen. 2017.
- [12] Fersini Elisabetta - Gasparini Francesca - Corchs Silvia Elena. *Detecting Sexist MEME On The Web: A Study on Textual and Visual Cues*. URL: <http://hdl.handle.net/10281/298213>.
- [13] Viviana Patti Elisa Bassignana Valerio Basile. *Hurtlex: A Multilingual Lexicon of Words to Hurt*. URL: <http://ceur-ws.org/Vol-2253/paper49.pdf>.

- [14] Facebook. *What does Facebook consider to be hate speech?* URL: [https://www.facebook.com/communitystandards/hate\\_speech](https://www.facebook.com/communitystandards/hate_speech).
- [15] Shuhua Liu - Thomas Forss. *Combining N-gram based Similarity Analysis with Sentiment Analysis in Web Content Classification.* URL: <https://www.scitepress.org/papers/2014/51703/51703.pdf>.
- [16] Shuhua Liu - Thomas Forss. *New Classification Models for Detecting Hate and Violence Web Content.* URL: <https://www.scitepress.org/Papers/2015/56367/56367.pdf>.
- [17] Edel Greevy e Alan F. Smeaton. “Classifying Racist Texts Using a Support Vector Machine”. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR ’04. Sheffield, United Kingdom: Association for Computing Machinery, 2004, pp. 468–469. ISBN: 1581138814. DOI: 10.1145/1008992.1009074. URL: <https://doi.org/10.1145/1008992.1009074>.
- [18] NLP Stanford Group. *The Stanford NLP Group.* URL: <https://nlp.stanford.edu>.
- [19] William Warner - Julia Hirschberg. *Detecting hate speech on the world wide web.* URL: <https://dl.acm.org/doi/pdf/10.5555/2390374.2390377>.
- [20] ILGA. *Hate crime and hate speech.* URL: <https://www.ilga-europe.org/what-we-do/our-advocacy-work/hate-crime-hate-speech>.
- [21] Maral Dadvar - Franciska de Jong - Roeland Ordelman - Dolf Trieschnigg. *Improved cyberbullying detection using gender information.* URL: [https://www.researchgate.net/publication/230701861\\_Improved\\_Cyberbullying\\_Detection\\_Using\\_Gender\\_Information](https://www.researchgate.net/publication/230701861_Improved_Cyberbullying_Detection_Using_Gender_Information).
- [22] Younghun Lee - Seunghyun Yoon - Kyomin Jung. *Comparative Studies of Detecting Abusive Language on Twitter.* URL: <https://aclanthology.org/W18-5113.pdf>.
- [23] Burnap Peter - Williams Matthew Leighton. *Hate speech, machine classification and statistical modelling of information flows on Twitter: interpretation and communication for policy decision making.* URL: <https://orca.cardiff.ac.uk/65227/>.
- [24] Karthik Dinakar - Roi Reichart - Henry Lieberman. *Modeling the detection of textual cyberbullying.* URL: <https://ie.technion.ac.il/~roiri/papers/3841-16937-1-PB.pdf>.
- [25] Francesco Marazzo. *Deciders. Chi decide sulla rete.* URL: [https://www.researchgate.net/publication/309014634\\_Deciders\\_Ch\\_i Decide\\_sulla\\_rete](https://www.researchgate.net/publication/309014634_Deciders_Ch_i Decide_sulla_rete).
- [26] Homa Hosseinmardi - Sabrina Arredondo Mattson - Rahat Ibn Rafiq - Richard Han - Qin Lv - Shivakant Mishra. *Detection of Cyberbullying Incidents on the Instagram Social Network.* URL: <https://arxiv.org/pdf/1503.03909.pdf>.
- [27] Dennis Njagi et al. “A Lexicon-based Approach for Hate Speech Detection”. In: *International Journal of Multimedia and Ubiquitous Engineering* 10 (apr. 2015), pp. 215–230. DOI: 10.14257/ijmue.2015.10.4.21.
- [28] Debora Nozza, Claudia Volpetti e Elisabetta Fersini. “Unintended Bias in Misogyny Detection”. In: *IEEE/WIC/ACM International Conference on Web Intelligence.* WI ’19. Thessaloniki, Greece: Association for Computing Machinery, 2019, pp. 149–155. ISBN: 9781450369343. DOI: 10.1145/3350546.3352512. URL: <https://doi.org/10.1145/3350546.3352512>.
- [29] Paula Fortuna - Sèrgio Nunes. *A Survey on Automatic Detection of Hate Speech in Text.* URL: <https://dl.acm.org/doi/pdf/10.1145/3232676>.

- [30] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. A cura di H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [31] Ashish Vaswani - Noam Shazeer - Niki Parmar - Jakob Uszkoreit - Llion Jones - Aidan N. Gomez - Łukasz Kaiser - Illia Polosukhin. *Attention Is All You Need*. URL: <https://arxiv.org/pdf/1706.03762.pdf>.
- [32] Amir Razavi et al. “Offensive Language Detection Using Multi-level Classification”. In: mag. 2010, pp. 16–27. ISBN: 978-3-642-13058-8. DOI: 10.1007/978-3-642-13059-5\_5.
- [33] Vasu Reddy. *Perverts and sodomites: homophobia as hate speech in Africa*. URL: <https://www.tandfonline.com/doi/abs/10.2989/16073610209486308>.
- [34] Maria Anzovino - Elisabetta Fersini - Paolo Rosso. *Automatic Identification and Classification of Misogynistic Language on Twitter*. URL: [http://personales.upv.es/prossro/resources/AnzovinoEtAl\\_NLDB18.pdf](http://personales.upv.es/prossro/resources/AnzovinoEtAl_NLDB18.pdf).
- [35] B. Sri Nandhini - J.I. Sheeba. *Cyberbullying Detection and Classification Using Information Retrieval Algorithm*. URL: <https://dl.acm.org/doi/pdf/10.1145/2743065.2743085>.
- [36] Yinhan Liu - Myle Ott - Naman Goyal - Jingfei Du - Mandar Joshi - Danqi Chen - Omer Levy - Mike Lewis - Luke Zettlemoyer - Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. URL: <https://arxiv.org/pdf/1907.11692.pdf>.
- [37] Yashar Mehdad - Joel Tetreault. *Do Characters Abuse More Than Words?* URL: <https://aclanthology.org/W16-3638.pdf>.
- [38] Twitter. *Regole di Twitter*. URL: <https://help.twitter.com/it/rules-and-policies/twitter-rules>.
- [39] UCSM. *IWG hatespeech public*. URL: <https://github.com/UCSM-DUE/>.
- [40] Stanford University. *Stanza – A Python NLP Package for Many Human Languages*. URL: <https://stanfordnlp.github.io/stanza/>.
- [41] Pinkesh Badjatiya - Shashank Gupta - Manish Gupta - Vasudeva Varma. *Deep learning for hate speech detection in tweets*. URL: <https://arxiv.org/pdf/1706.00188.pdf>.
- [42] Christian Wigand - Melanie Voin. *Speech by Commissioner Jourová - 10 years of the EU Fundamental Rights Agency: a call to action in defence of fundamental rights, democracy and the rule of law*. URL: [https://ec.europa.eu/commission/presscorner/detail/en/SPEECH\\_17\\_403](https://ec.europa.eu/commission/presscorner/detail/en/SPEECH_17_403).
- [43] Irene Kwok - Yuzhou Wang. *Locate the hate: Detecting tweets against blacks*. URL: <https://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6419/6821>.
- [44] Zeerak Waseem. *Hate Speech Twitter annotations*. URL: <https://github.com/ZeerakW/hatespeech>.
- [45] Zeerak Waseem e Dirk Hovy. *Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter*. URL: <https://aclanthology.org/N16-2013.pdf>.

- [46] Thomas Davidson - Dana Warmsley - Michael Macy - Ingmar Weber. *The Automated Hate Speech Detection and the Problem of Offensive Language*. URL: <https://arxiv.org/pdf/1703.04009.pdf>.
- [47] Jamie Bartlett - Richard Norrie - Sofia Patel - Rebekka Rumpel - Simon Wibberley. *Misogyny on Twitter*. URL: [https://www.demos.co.uk/files/MISOGYNY\\_ON\\_TWITTER.pdf](https://www.demos.co.uk/files/MISOGYNY_ON_TWITTER.pdf).
- [48] Anna Schmidt - Michael Wiegand. *A Survey on Hate Speech Detection using Natural Language Processing*. URL: <https://aclanthology.org/W17-1101.pdf>.
- [49] Pete Burnap - Matthew L. Williams. *Identifying cyber hate on Twitter across multiple protected characteristics*. URL: <https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-016-0072-6>.
- [50] Yahoo. *Webscope datasets*. URL: <https://webscope.sandbox.yahoo.com>.
- [51] YouTube. *Hate speech policy*. URL: <https://support.google.com/youtube/answer/2801939>.