Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Toxic Comment Classification

Authors:
Simone Monti - 807994 - s.monti21@campus.unimib.it
Vittorio Maggio - 817034 - v.maggio5@campus.unimib.it

18 giugno 2021

Sommario

L'obiettivo di questo paper è quello di presentare un'analisi completa di un approccio di deep learning al fine di prendere parte alla sfida proposta dalla piattaforma *Kaggle*: *Toxic Comment Classification Challenge*. Nel corso della trattazione verrà effettuata un'analisi esplorativa del dataset, verrà presentata l'architettura di diverse tipologie di reti neurali con cui poi verranno costruiti i modelli predittivi. Infine quest'ultimi verranno valutati e confrontati.

1 Introduzione

Negli ultimi anni i social networks e le comunità online hanno visto una larga crescita di interazioni sociali, dalla condivisione delle proprie esperienze personali alle discussioni politiche. Questo ha fatto si che essi abbiano assunto un ruolo sempre più centrale all'interno della vita sociale delle persone. Purtroppo però, i social non si sono rivelati essere solo luoghi di condivisione di idee e pensieri, ma anche luoghi in cui sono sempre più frequenti commenti tossici, d'odio, di minacce e di insulti. Questo sta portando non solo problemi di convivenza all'interno di queste comunità ma anche problemi più seri, di natura psicologica, alle vittime di tali commenti. L'individuazione e la gestione di questi, dunque, è diventata fondamentale.

L'obiettivo proposto dalla challange di Kaggle *Toxic Comment Classification Challenge* [1] è quello di classificare correttamente ciascun commento in una o più categorie (tra cui per esempio *minacce, oscenità, insulti,* e *odio razziale*) in base al tipo di tossicità presente. Da notare che un commento può appartenere a più categorie contemporaneamente come può non presentare nessuna di queste e quindi essere esente dalla classificazione.

Dopo una fase di analisi esplorativa del dataset e di pre-processing del dataset, l'attenzione è stata posta sull'individuazione nella letteratura delle migliori architetture neurali per la text-classification. Sono stati dunque implementati modelli basati su *LSTM* e *DistilBERT*.

2 Dataset

Kaggle fornisce un ricco dataset di training (esente da missing value) formato da commenti presi dal web (nello specifico commenti tratti da Wikipedia), categorizzati manualmente in sei diverse label binarie in base all'abuso presente: *toxic, severe*

toxic, oscene, threat, insult e identity hate. Ciascun commento può appartenere a più categorie contemporaneamente (oppure nessuna di esse). Il dataset di training è formato da 159571 commenti e nella *Figura 1* è possibile osservare la distribuzione per ciascuna etichetta.

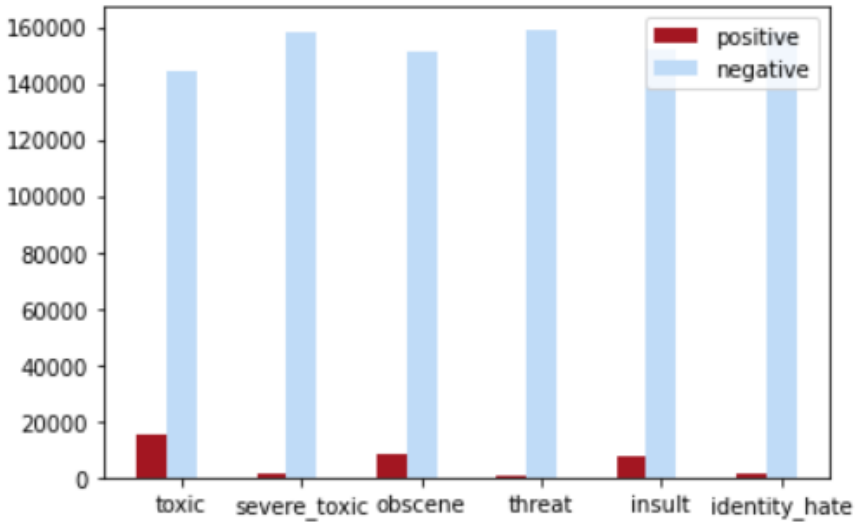

Figura 1: confronto presenza/assenza di tossicità

Data la possibilità di un commento di essere categorizzato in diverse classi contemporaneamente abbiamo individuato la presenza di relazioni fra le etichette utilizzando la matrice di correlazione (Figura 2). Si evidenzia:

- (sorprendentemente) una bassa correlazione fra *toxic* e *severe toxic*,
- una forte correlazione fra le categorie *insult* e *obscene* (Figura 3),
- una correlazione fra *toxic* e rispettivamente *insult* e *obscene*.

Sono state individuate le parole che, effettivamente, discriminano un commento ad appartenere ad una categoria, per far ciò è stato utilizzato WordCloud [7], un tool di visualizzazione che permette di evidenziare le parole più frequenti all'interno di un testo. Il tool è stato utilizzato con l'insieme dei commenti di ciascuna categoria dopo l'applicazione di un pre-processing dedicato (tokenization, stemming, stopword removing e conversione in lower case).

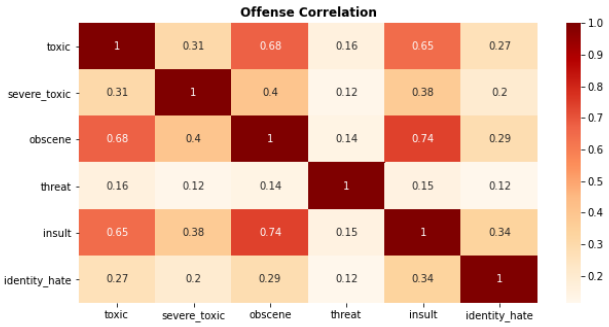

Figura 2: matrice di correlazione

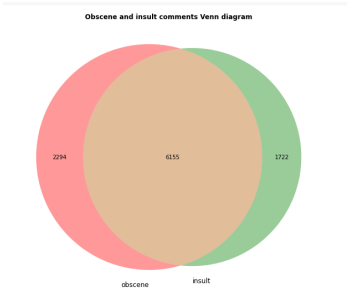

Figura 3: diagramma di Venn tra obscene e insult

Figura 4: identity hate

Figura 5: obscene

Figura 6: insult

Figura 7: threat

Figura 8: severe toxic

Figura 9: toxic

Si nota come le classi che sono fortemente correlate hanno evidenziato, all'interno del proprio diagramma Wordcloud, le stesse parole.

Come ultimo step della fase di data exploration è stata calcolata la distribuzione della lunghezza dei commenti *Figura 10*, questa misura è stata utilizzata per scegliere la lunghezza di input dei modelli neurali, per la quale si è cercato di mantenere una certa efficienza preservando il maggior grado di informazione possibile.

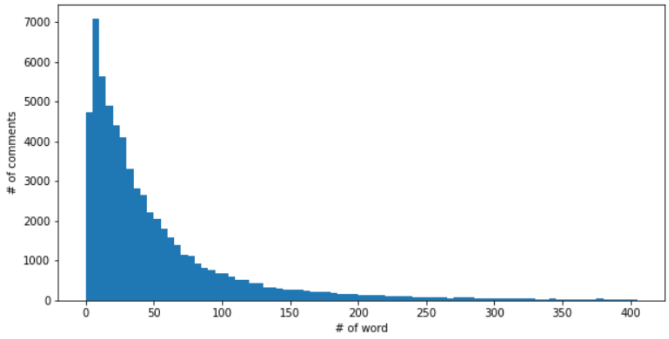

Figura 10: distribuzione della lunghezza dei commenti

3 L'approccio metodologico

L'obiettivo della trattazione, come anticipato, è quello di partecipare in maniera competitiva alla challenge proposta da *Kaggle*: *Toxic Comment Classification*. La Figura 11 raffigura la pipeline seguita.

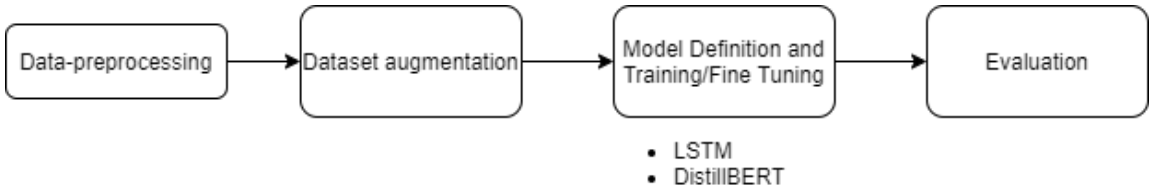

Figura 11: flowchart del processo

Durante il pre-processing il testo è stato trasformato in minuscolo, sono stati rimossi tutti i segni di punteggiatura e gli spazzi in eccesso. Data la natura fortemente sbilanciata del dataset, osservabile in *Figura 11*, è stata applicata la tecnica di data augmentatio EDA (Easy Data Augmentation)[2] sulle classi con un numero di campioni molto basso:

- *severe toxic*: ciascun commento è stato aumentato 10 volte,
- *threat*: ciascun commento è stato aumentato 20 volte,
- *indentity hate*: ciascun commento è stato aumentato 10 volte,

Data la natura multiclasse del dataset e la correlazione fra alcune label, anche il numero di commenti delle altre categorie sono risultat aumentati, la *Figura 12* mostra la distribuzione del nuovo dataset aumentato.

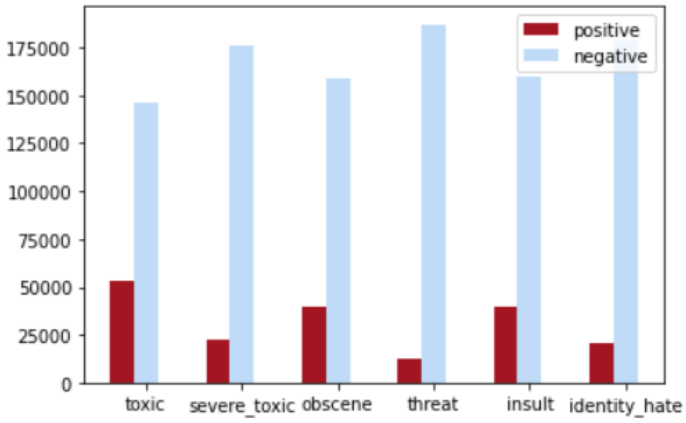

Figura 12: Distribuzione del dataset aumentato

In seguito, il testo dei vari commenti è stato tokenizzato. La dimensione di input dei modelli è stata fissata a 128, dunque è stato effettuato il padding dei commenti con un numero di token inferiore, mentre sono stati troncati al 128esimo token i commenti di lunghezza superiore. La scelta è stata fatta per cercare di conservare il maggior dettaglio possibile senza però andar a perdere d'efficienza dei modelli.

Secondo la letterature disponibile, i modelli più adatti a questa tipologia di task sono quelli in grado di apprendere relazioni contestuali fra parole. Quindi, le architetture utilizzate sono state: LSTM (Long short-term memory) Bidirezionale e DistilBERT, una versione "distillata" (40% di parametri in meno) di BERT (Bidirectional Encoder Representations from Transformers), rete neurale basata sui Transformer (meccanismo di attenzione in grado di imparare e rappresentare relazioni tra parole in base al loro contesto). Nello specifico:

- **LSTM** [8]: un layer Embedding (input_dim: 128, output_dim=128), un layer SpatialDropout1D (con rate uguale a 0.3), un layer LSTM bidirezionale con 42 neuroni, un layer BatchNormalization, un layer GlobalMaxPool1d, un layer Dropout (con rate uguale a 0.3), un layer Dense con 18 neuroni che utilizza *Relu* come activation function, un layer Dense con 6 neuroni che utilizza *Sigmoid* come activation function (avendo come output 6 differenti valori binari).
  Totale parametri addestrabili: 2,619,436;

- **DistilBERT** [6]: è stato effettuato il fine-tuning del modello pre-addestrato *distilbert-base-uncased*. Tale modello ha la stessa architettura generale di BERT, per ridurre il numero di parametri sono però stati rimossi l'embedding del *token-type*, il *pooler* e il numero di layer è stato ridotto di un fattore di 2 (6-layer, 768-hidden, 12-heads, 66M di parametri). Per il suo addestramento è stata applicata la tecnica della *Knowledge distillation*: DistilBERT è stato addestrato usando la supervisione della rete più grande BERT. In questo modo, il modello impara la stessa rappresentazione interna della lingua inglese rispetto al modello del suo insegnante (BERT), pur essendo più veloce per i compiti di inferenza. Il dataset usato per l'addestramento comprende testo scaricato da Wikipedia (2.5B di parole) e da BookCorpus (800M di parole).
  Per effettuare il fine-tuning, all'ultimo layer del modello pre-addestrato sono stati aggiunti: 2 Dense layer da 256 e 32 neuroni con funzione di attivazione *Relu* e un layer Dense di output con 6 neuroni e funzione di attivazione *Sigmoid* (avendo come output 6 differenti valori binari). Fra ogni layer aggiuntivo è stato aggiunto un layer di Dropout (con rate uguale a 0.2).

Entrambe le architetture sono state addestrate sui due differenti dataset disponibili (originale e aumentato) e valutate sul dataset di test proposto dalla challenge.

I parametri utilizzati per LSTM sono i seguenti:

- epoche: 50;
- batch size: 256;
- validation split: 0.2;
- optimizer: Adam(con rate pari a 0.0001);
- loss: BinaryCrossEntropy;

Per prevenire un possibile overfitting, sono state utilizzate le tecniche:

- Dropout [9];
- EarlyStopping [10]:
  - monitor: validation loss;
  - patience: 3;
  - restore_best_weights: True.

I parametri utilizzati per il fine tuning di DistilBERT sono i seguenti:

- epoche: 6;
- batch size: 64;
- validation split: 0.2;
- optimizer: Adam(con rate pari a 0.00005);
- loss: BinaryCrossEntropy;

Per prevenire un possibile overfitting, sono state utilizzate le tecniche:

- Dropout [9];
- EarlyStopping [10]:
  - monitor: validation loss;
  - patience: 2;
  - restore_best_weights: True.

Come misura di *loss* è stata scelta la *Binary Cross Entropy* essendo questo un problema di classificazione binario. I modelli sono stati valutati principalmente sulla metrica AUC (che è stata, inoltre, utilizzata dalla competizione stessa) e sullo score F1.

4 Risultati e valutazioni

Di seguito sono riportati i risultati ottenuti sul test set addestrando i modelli sia sul dataset originale che sulla versione aumentata.

LSTM (dataset originale)

		Predicted	
	Toxic	0	1
Actual	0	54992	2896
	1	1521	4569

		Predicted	
	Severe toxic	0	1
Actual	0	63532	179
	1	272	95

		Predicted	
	Obscene	0	1
Actual	0	58906	1381
	1	1006	2685

		Predicted	
	Threat	0	1
Actual	0	63762	5
	1	211	0

		Predicted	
	Insult	0	1
Actual	0	59193	1358
	1	1251	2176

		Predicted	
	Identity hate	0	1
Actual	0	63264	2
	1	709	3

Tabella 1: Matrici di confusione LSTM (dataset originale) test set

	Precision	**Recall**	**F1-score**	**support**
Toxic	0.61	0.75	0.67	6090
Severe_toxic	0.35	0.26	0.30	367
Obscene	0.66	0.73	0.69	3691
Threat	0.00	0.00	0.00	211
Insult	0.62	0.63	0.63	3427
Identity_hate	0.60	0.00	0.01	712
Micro avg	0.62	0.66	0.64	14498

Tabella 2: Perf. report LSTM (dataset originale) test set

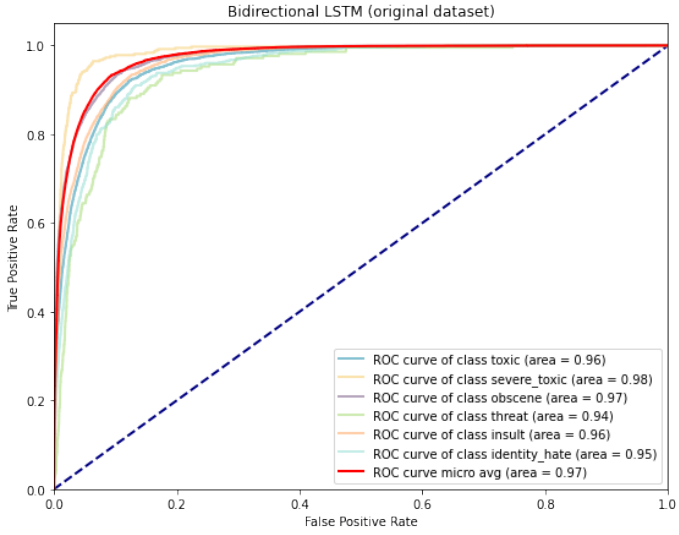

Figura 13: ROC e AUC LSTM (dataset originale) test set

LSTM (dataset aumentato)

Actual \ Predicted	Toxic 0	Toxic 1
0	53154	4734
1	902	5188

Actual \ Predicted	Severe toxic 0	Severe toxic 1
0	63542	69
1	318	49

Actual \ Predicted	Obscene 0	Obscene 1
0	58544	1743
1	836	2855

Actual \ Predicted	Threat 0	Threat 1
0	63759	8
1	207	4

Actual \ Predicted	Insult 0	Insult 1
0	59074	1477
1	1141	2286

Actual \ Predicted	Identity hate 0	Identity hate 1
0	63237	29
1	612	100

Tabella 3: Matrici di confusione LSTM (dataset aumentato) test set

	Precision	Recall	F1-score	support
Toxic	0.52	0.85	0.65	6090
Severe_toxic	0.42	0.13	0.20	367
Obscene	0.62	0.77	0.69	3691
Threat	0.33	0.02	0.04	211
Insult	0.61	0.67	0.64	3427
Identity_hate	0.78	0.14	0.24	712
Micro avg	0.57	0.72	0.63	14498

Tabella 4: Perf. report LSTM (dataset aumentato) test set

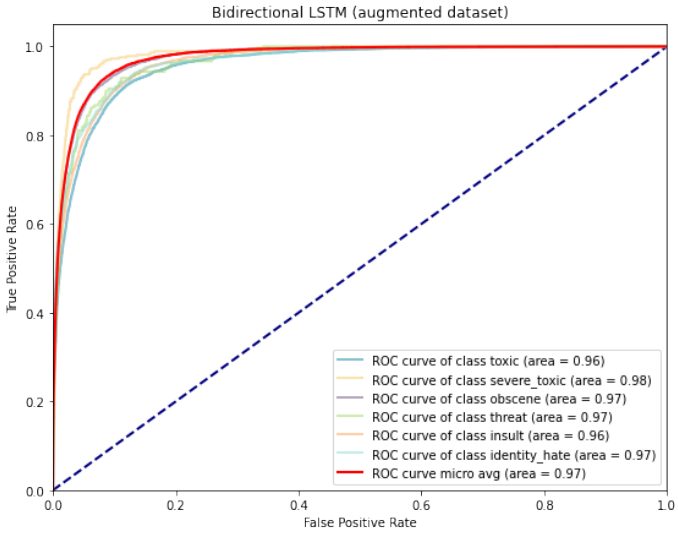

Figura 14: ROC e AUC LSTM (dataset aumentato) test set

DistillBERT (dataset originale)

Actual \ Predicted	Toxic 0	Toxic 1
0	52382	5506
1	486	5604

Actual \ Predicted	Severe toxic 0	Severe toxic 1
0	62843	768
1	90	277

Actual \ Predicted	Obscene 0	Obscene 1
0	58042	2245
1	617	3074

Actual \ Predicted	Threat 0	Threat 1
0	3602	165
1	81	130

Actual \ Predicted	Insult 0	Insult 1
0	58861	1690
1	772	2655

Actual \ Predicted	Identity hate 0	Identity hate 1
0	62928	338
1	279	433

Tabella 5: Matrici di confusione DistillBERT (dataset originale) test set

	Precision	Recall	F1-score	support
Toxic	0.50	0.92	0.65	6090
Severe_toxic	0.27	0.75	0.39	367
Obscene	0.58	0.83	0.68	3691
Threat	0.44	0.62	0.51	211
Insult	0.61	0.77	0.68	3427
Identity_hate	0.56	0.61	0.58	712
Micro avg	0.53	0.84	0.65	14498

Tabella 6: Perf. report DistillBERT (dataset originale) test set

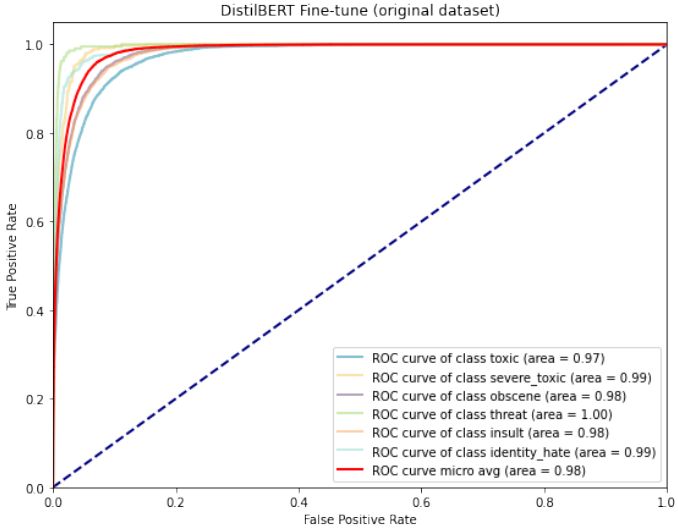

Figura 15: ROC e AUC DistillBERT (dataset originale) test set

DistillBERT (dataset aumentato)

Actual \ Predicted	Toxic 0	Toxic 1
0	52109	5779
1	353	5737

Actual \ Predicted	Severe toxic 0	Severe toxic 1
0	62877	734
1	103	264

Actual \ Predicted	Obscene 0	Obscene 1
0	57260	3027
1	421	3270

Actual \ Predicted	Threat 0	Threat 1
0	63628	139
1	91	120

Actual \ Predicted	Insult 0	Insult 1
0	57725	2826
1	431	2996

Actual \ Predicted	Identity hate 0	Identity hate 1
0	62817	449
1	236	476

Tabella 7: Matrici di confusione DistillBERT (dataset aumentato) test set

	Precision	Recall	F1-score	support
Toxic	0.50	0.94	0.65	6090
Severe_toxic	0.26	0.72	0.39	367
Obscene	0.52	0.89	0.65	3691
Threat	0.46	0.57	0.51	211
Insult	0.51	0.87	0.65	3427
Identity_hate	0.51	0.97	0.58	712
Micro avg	0.50	0.89	0.64	14498

Tabella 8: Perf. report DistillBERT (dataset aumentato) test set

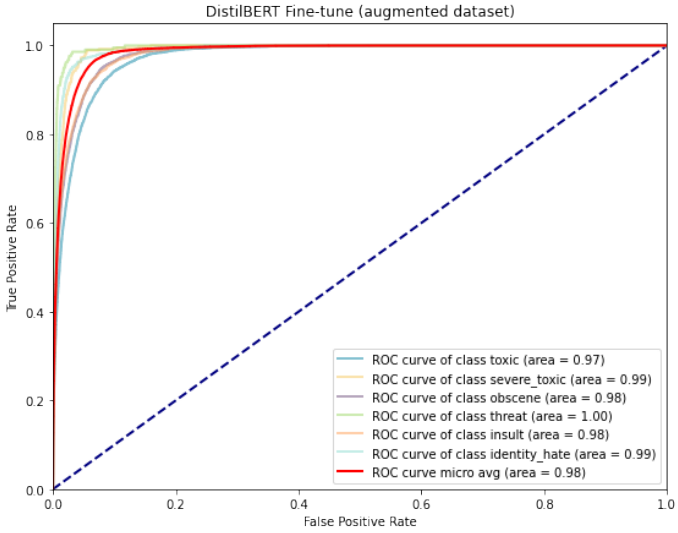

Figura 16: ROC e AUC DistillBERT (dataset aumentato) test set

5 Discussione dei risultati

Per confrontare le performance dei modelli presentati è stata scelta la metrica AUC (metrica utilizzata dalla stessa challenge) e il punteggio F1, essendo queste le metriche più adatte per dataset sbilanciati. L'Accuracy non è stata presa in considerazione in quanto, a causa della natura sbilanciata del dataset, risulta biased. Per la medesima ragione è stata considerata la micro-average delle metriche.

I quattro modelli addestrati sono stati valutati sul dataset di test offerto direttamente dalla challenge.

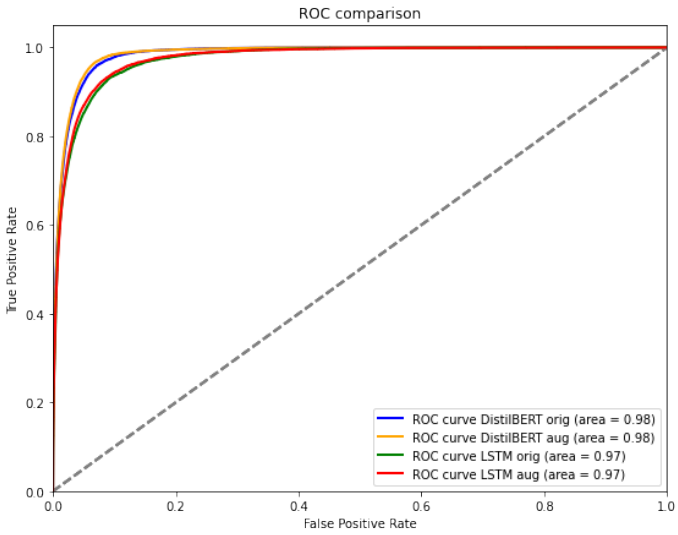

Figura 17: Confronto ROC modelli

	Dataset originale		Dataset aumentato	
	LSTM	**DistilBERT Fine tune**	**LSTM**	**DistilBERT Fine tune**
AUC	0.97	0.98	0.97	0.98
F1	0.64	0.65	0.63	0.64

Tabella 9: Confrondo modelli AUC e F1

Il grafico in Figura 17 confronta la misura AUC di ciascun modello valutato, si denota una leggera differenza di prestazione fra le differenti architetture, DistillBert

ha un punteggio superiore di 0.01 punto rispetto ad LSTM, mentre è presente una differenza trascurabile fra i modelli aventi la stessa architettura ma addestrati su differenti dataset (originale e aumentato).
Per entrambe le architetture l'addestramento sul dataset aumentato ha portato a un calo delle performance in termini di F1 score, in particolare si nota come la data augmentation porti un incremento in termini di recall ma ad un decremento in termini di precision.

È possibile notare la difficoltà dell'architettura basata su LSTM di prevedere le classi con pochi campioni positivi, indipendentemente dal dataset di addestramento utilizzato. Il modello base non riesce ad classificare correttamente alcun esempio delle classi threat e identity_hate *Tabella 1*, leggermente migliore risulta il modello addestrato sul dataset aumentato *Tabella 3*. Con queste classi il modello basato su DistillBERT si comporta notevolmente meglio, come è possibile vedere dalle matrici di confusione *Tabelle 5-7*.
Il modello che quindi ha presentato performance migliori sul test set è il modello basato sul fine-tune di DistilBERT addestrato sul dataset originale (senza data augmentation).

6 Conclusione

In questo paper è stata esposta la pipeline utilizzata per partecipare alla challenge presentata da kaggle *Toxic Comment Classification*, in particolare è stato utilizzato un approccio di deeplearning. Dopo una preliminare analisi del dataset è stata utilizzata la tecninca di data augmentation, per cercare di andare a sopperire alla natura fortemente sbilanciata del dataset. Successivamente, sono state descritte e valutate due differenti architetture di rete neurali appartenenti a periodi differenti: LSTM e DistillBERT. Entrambi i modelli dopo essere stati addestrati con i due differenti dataset (standard e aumentato) hanno dato prova di buone prestazioni, ma quello basato sui transformers, DistillBERT, fine tuned sul dataset non aumentato è riuscito ad ottenere un punteggio leggermente migliore rispetto agli altri e in linea con le prime posizioni dei classificati alla challenge.

Riferimenti bibliografici

[1] Kaggle - Toxic Comment Classification
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[2] EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks
https://arxiv.org/abs/1901.11196

[3] Efficient Estimation of Word Representations in Vector Space
https://arxiv.org/pdf/1301.3781.pdf

[4] Glove: Global Vectors for Word Representation
https://www.researchgate.net/publication/284576917_Glove_Global_Vectors_for_Word_Representation

[5] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
https://arxiv.org/pdf/1810.04805.pdf

[6] DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter
https://arxiv.org/pdf/1910.01108.pdf

[7] WordCloud for Python
https://github.com/amueller/word$_c$loud

[8] LONG SHORT-TERM MEMORY
https://www.researchgate.net/publication/13853244$_L$ong$_S$hort − term$_M$emory

[9] Dropout: A Simple Way to Prevent Neural Networks from Overfitting
http://jmlr.org/papers/v15/srivastava14a.html

[10] Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks
https://arxiv.org/abs/1903.11680