Riconoscimento automatico dell'ironia e del sarcasmo @ EVALITA 2018*

Giacomo Fidone
LM in Informatica Umanistica (Tecnologie del Linguaggio)
Università di Pisa
g.fidone1@studenti.unipi.it

Abstract

In questa relazione descrivo lo sviluppo di diversi sistemi per lo svolgimento dei *task* di IronITA 2018, riguardanti il riconoscimento automatico dell'ironia e del sarcasmo in testi in lingua italiana provenienti da *Twitter*.

1 Introduzione

In questa relazione descrivo lo sviluppo di diversi sistemi per lo svolgimento dei due *task* proposti dagli autori di IronITA 2018 (Cignarella *et al.* 2018) nel contesto della competizione EVALITA 2018[1], ovvero il riconoscimento automatico dell'ironia e di una sua forma peculiare – il sarcasmo – in testi in lingua italiana provenienti da *Twitter*.

Si tratta di *task* notoriamente complessi per i sistemi di elaborazione automatica del linguaggio. Le ragioni di questa difficoltà sono da ricondursi anzitutto alla natura stessa dell'ironia, che comporta sia uno scarto tra il significato letterale ed il significato inteso dal locutore; sia una più generale dissimulazione delle reali intenzioni comunicative. A ciò si aggiunge una ricca tassonomia, che comprende non solo l'ironia come figura retorica, ma anche l'umorismo satirico, il sarcasmo, o la più esplicita derisione (Marchetti *et al.* 2007). Inoltre, nel contesto dei *task* di IronITA 2018, non va trascurato il ruolo della lingua usata dagli utenti di Twitter, sia per la presenza di forme sub-standard sia per l'inclusione di elementi idiosincratici quali *hashtag*, menzioni, *emoticon* o *hyper-link*.

Nonostante i limiti evidenti, il riconoscimento automatico dell'ironia può svolgere importanti funzioni di supporto non solo ad un'analisi del *sentiment*, ma anche allo svolgimento di *task* più specifici. Come segnalato dagli autori di IronITA 2018, un'attenzione crescente è oggi rivolta al sarcasmo – un'ironia dai toni caustici e taglienti – per la sua frequente correlazione con la sfera politica e con forme di *hate speech*.

2 Metodo

2.1 Descrizione dei *task*

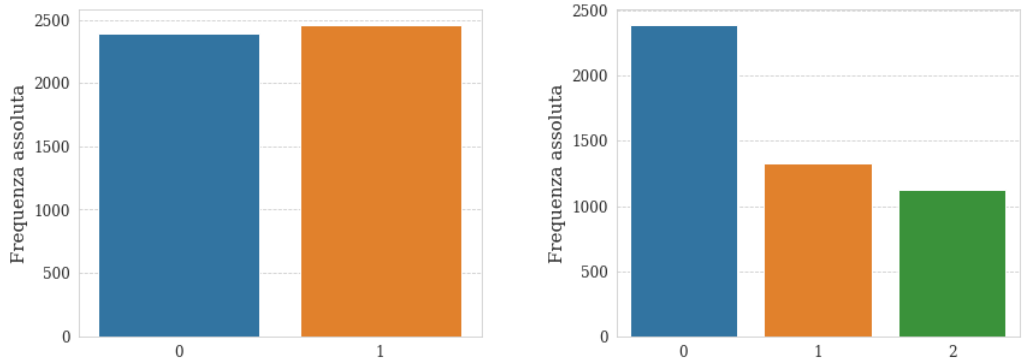
I *task* proposti dagli autori di IronITA 2019 sono i seguenti: *task* A, ovvero un *task* di classificazione binaria in cui al sistema è richiesto di predire se un testo è ironico o non ironico; e *task* B, ovvero un *task* di classificazione multi-classe in cui al sistema è richiesto di predire se un testo è ironico, non ironico o sarcastico. I sistemi sono stati quindi sviluppati per lo svolgimento di entrambi i *task* proposti.

*Progetto per il corso di Linguistica Computazionale II, Università di Pisa, A.A. 2022/23.
[1]https://www.evalita.it/campaigns/evalita-2018/

2.2 Dataset

Per lo sviluppo dei sistemi si opta per l'uso esclusivo dei dati forniti dagli autori di IronITA 2018.[2] Il dataset consiste di un corpus di 4849 testi in lingua italiana (a loro volta prelevati da corpora già esistenti) di cui 3977 (80%) riservati al *training* (TR) e 872 (20%) riservati al *test* (TS). Il corpus è associato a quattro variabili: *twitter_id*, che riporta un identificativo univoco per ciascun testo; *source*, che indica il nome del corpus da cui il testo è stato prelevato; *irony*, che registra la presenza (1) o l'assenza (0) di ironia; e *sarcasm*, che registra la presenza (1) o l'assenza (0) di sarcasmo data la presenza di ironia.

Per il *task* B è stata definita una nuova variabile multi-classe denominata *irony_sarcasm* a partire dall'informazione delle variabili booleane. I valori ammessi di *irony_sarcasm* sono 0 (assenza di ironia), 1 (ironia senza sarcasmo) e 2 (ironia con sarcasmo). La Figura 1 riporta la distribuzione dei testi secondo le due classi target.



(a) *irony* (b) *irony_sarcasm*

Figura 1: Distribuzione dei dati secondo ciascuna classe target.

2.3 Descrizione dei sistemi

Per lo svolgimento di entrambi i *task* ho preso in considerazione due tipologie di sistemi: *Support Vector Machine* (SVM) ed una versione pre-trainata per l'italiano di *Bidirectional Encoder Representations from Transformers* (BERT - Devlin *et al.* 2018)[3].

I SVM sono stati testati rispetto a tre diversi *feature set*:

1. Un *feature set* estratto con il *tool Profiling-UD* (Brunato *et al.* 2020), che codifica informazione riguardante il profilo linguistico del testo;

2. Un *feature set* di *n*-grammi di caratteri, di parole, di parti del discorso (PoS) e/o di lemmi estratti dall'annotazione morfo-sintattica del corpus effettuata con *Universal Dependencies* (UD - De Marneffe *et al.* 2020);

3. Un *feature set* di *Word Embeddings* (WE) 32-dimensionali estratti con CBOW dal corpus *itWaC* (Cimino *et al.* 2018).

Per *n*-grammi e WE il *feature set* per ciascun *task* è stato selezionato con un approccio *wrapper*, ovvero calcolando la media dell'F1-macro su una *5-Fold CV* (TR) di un SVM lineare addestrato su ciascun *feature set* candidato. I *feature set* di *n*-grammi candidati sono stati generati considerando ogni possibile combinazione di *n*-grammi per ciascun valore di n ($1 \leq n \leq 4$). I *feature set* di WE candidati sono stati invece generati aggregando i WE di parole a seconda della loro PoS (UD) e sfruttando due diverse funzioni di aggregazione, ovvero media e concatenazione delle medie calcolate rispetto a ciascuna PoS.

[2] https://live.european-language-grid.eu/catalogue/corpus/7372/download/
[3] https://huggingface.co/dbmdz/bert-base-italian-cased

Più precisamente sono state considerate (per ciascuna funzione di aggregazione) possibili combinazioni di *token* lessicali (NOUN, VERB, ADJ, ADV) con nomi propri (PROPN), punteggiatura (PUNCT) e simboli (SYM).[4] Per la loro valutazione sono stati considerati anche la media e la concatenazione di *token* non lessicali (esclusi PUNCT e SYM) e la media di tutti i WE.

Per entrambi i *task* la selezione degli iper-parametri del SVM è stata modellata su una *randomized search* con *3-fold CV* stratificata – per il problema di classificazione multi-classe è stata usata l'F1-macro come criterio di selezione. Lo spazio degli iper-parametri testato è riportato nella Tabella 1.

Tabella 1: Iper-parametri testati per SVM

Iper-parametro	**Descrizione**	**Valori testati**
C	Coefficiente del termine di errore	Potenze del 10 in $[10^{-4}, 10^{4}]$
γ	Coefficiente del kernel	Potenze del 10 in $[10^{-4}, 10^{4}]$
Kernel	Funzione kernel	Lineare, RBF, Polinomiale
Class Weight	Distribuzione dei costi	Uniforme, Bilanciata

Per l'addestramento di BERT una porzione pari al 10% dei dati di *training* è stata separata come *set* di *validation* (VL) per la valutazione della performance del modello ad ogni epoca ed il controllo del tempo di convergenza. Seguendo le indicazioni degli sviluppatori di *HuggingFace*[5], l'ottimizzazione della *loss* è stata eseguita per 5 epoche usando *AdamW* (Loshchilov e Hutter 2017) con la seguente configurazione: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, *learning rate* $= 2 \times 10^{-5}$, *weight decay* $= 10^{-2}$ e *batch* $= 8$.

2.4 *Tools*

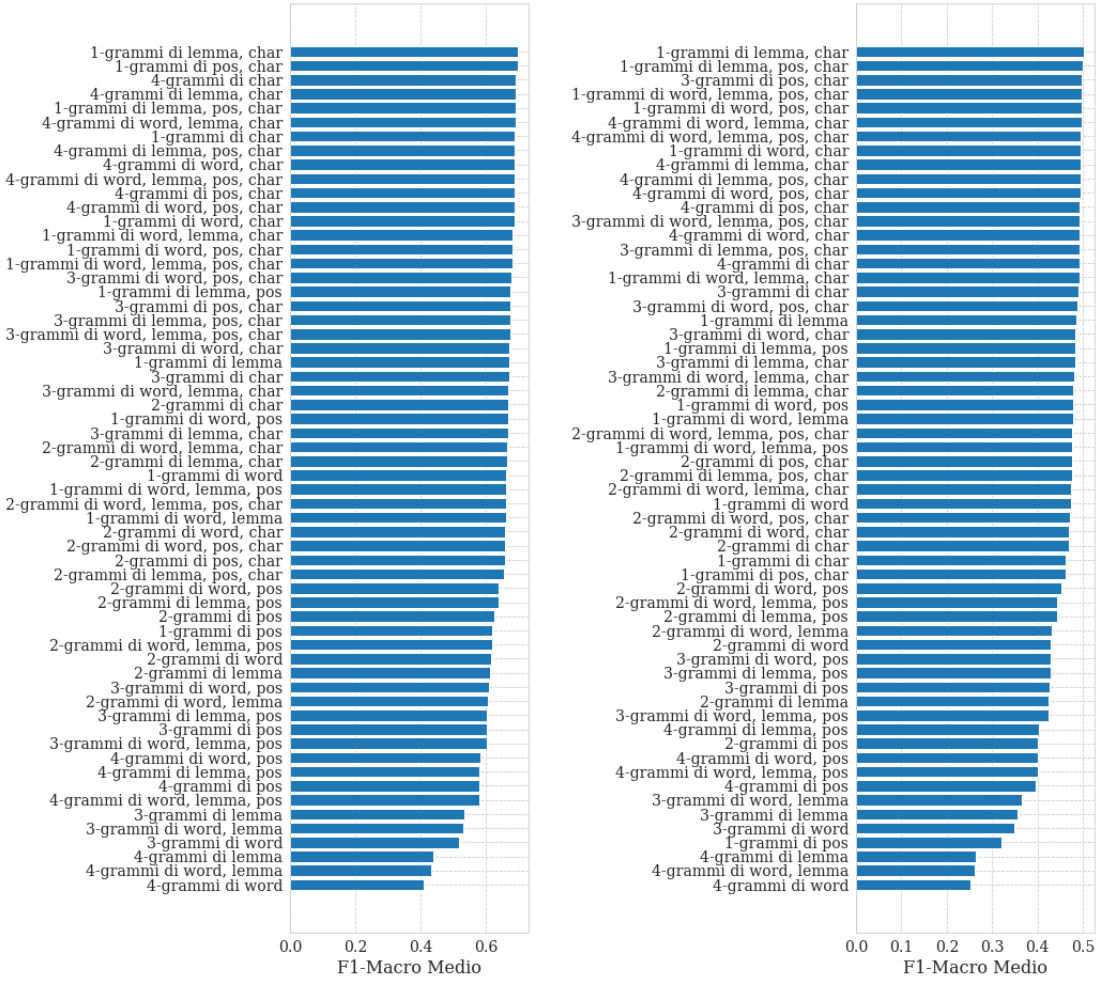
Per lo sviluppo dei SVM ho fatto uso della libreria *Sci-kit Learn*[6], che è stata utilizzata anche a supporto della valutazione di tutti i sistemi. Per lo sviluppo di BERT ho fatto uso della libreria *Transformers*[7]. Per l'estrazione del profilo linguistico e dell'annotazione morfo-sintattica ho fatto uso del già citato *Profiling-UD*[8].

3 Risultati sperimentali

3.1 *Feature selection*

Per entrambi i *task* la maggior parte dei *feature set* di n-grammi candidati risulta associata a prestazioni sufficienti del SVM lineare (Figura 2). Maggiore importanza, tuttavia, può essere attribuita al ruolo degli unigrammi di caratteri – probabilmente per l'uso peculiare dei segni di interpunzione nei testi associati ad ironia o sarcasmo; unito al ruolo degli unigrammi di lemmi – verosimilmente per la connotazione tipicamente ironica o sarcastica di alcune parole piene.

[4] UD etichetta gli *hashtag*, le menzioni e le *emoticon* come simboli con specifici tratti funzionali.
[5] https://huggingface.co/docs/transformers/main_classes/trainer#transformers.TrainingArguments
[6] https://scikit-learn.org/stable/
[7] https://huggingface.co/docs/transformers/index
[8] http://www.italianlp.it/demo/profiling-ud/

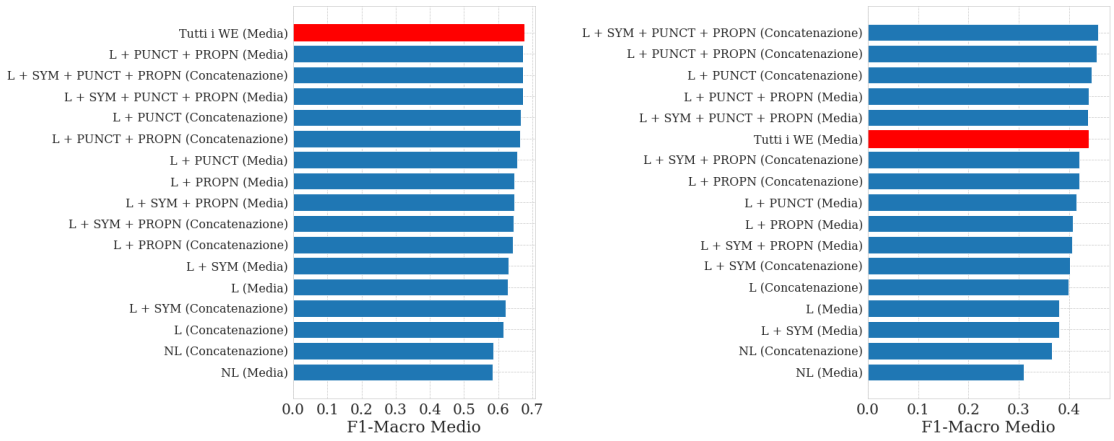


(a) *Task* A (b) *Task* B

Figura 2: Media dell'F1-Macro su *5-Fold CV* di SVM lineare per ciascun *feature set* candidato di *n*-grammi.

Dal momento che ciascun *feature set* candidato è stato generato per un solo valore di n, il *feature set* utilizzato per l'addestramento del SVM è stato selezionato combinando alcuni dei *feature set* candidati più performanti secondo i risultati della *5-Fold CV*. In particolare sono stati selezionati: unigrammi di caratteri, unigrammi di lemmi, quadrigrammi di caratteri e quadrigrammi di lemmi per il *task* A; unigrammi di caratteri, unigrammi di lemmi, trigrammi di caratteri, trigrammi di PoS per il *task* B.

La valutazione dei *feature set* di WE candidati (Figura 3) ha rivelato l'importanza di tutta l'informazione per lo svolgimento del *task A* – compresa quella proveniente da *token* non lessicali, per la quale l'F1-macro supera le prestazioni di un classificatore randomico (F1-macro = 0.5). Diversamente, per lo svolgimento del *task B* alcune specifiche combinazioni di PoS sono correlate a prestazioni più competitive di quelle osservate sul SVM lineare che fa uso di tutta l'informazione. In entrambi i casi, le prestazioni del SVM lineare su *token* lessicali risulta migliorare in combinazione con l'informazione relativa ai nomi propri, alla punteggiatura ed ai simboli.



(a) *Task* A (b) *Task* B

Figura 3: Media dell'F1-macro su *5-Fold CV* di SVM lineare per ciascun *feature set* candidato di WE. «L» e «NL» si riferiscono rispettivamente ai *token* lessicali e ai *token* non lessicali esclusi PUNCT e SYM. In rosso è evidenziata la media di tutti i WE.

Per l'addestramento dei SVM sono stati quindi selezionati: la media di tutti i WE per il *task* A; la concatenazione delle medie di NOUN, VERB, ADJ, ADV, PROPN, PUNCT, SYM per il *task* B.

3.2 *Tuning* degli iper-parametri

Si riportano nella tabella 2 gli iper-parametri selezionati per l'addestramento di ciascun SVM secondo i risultati della *3-fold CV* (TR).

Tabella 2: Iper-parametri selezionati per ciascun SVM

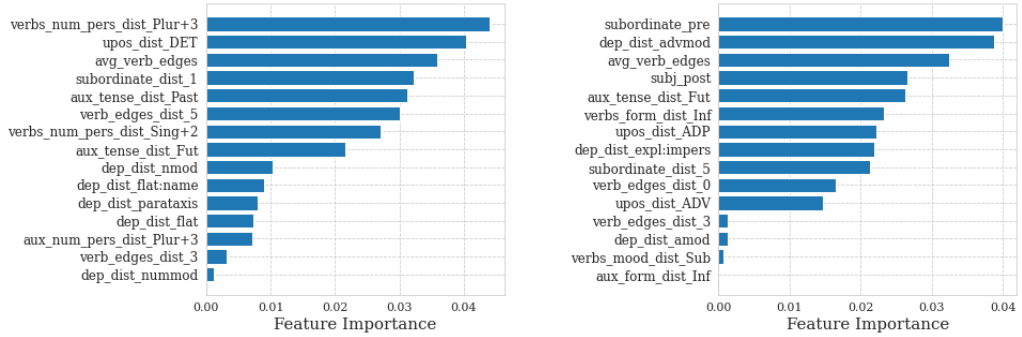
	Modello	**C**	γ	**Kernel**	**Class Weight**
Task A	SVM (Profiling-UD)	1	10^3	Lineare	Bilanciata
	SVM (*n*-grammi)	10^2	10^{-4}	RBF	Uniforme
	SVM (WE)	1	1	RBF	Bilanciata
Task B	SVM (Profiling-UD)	1	10^{-4}	Lineare	Uniforme
	SVM (*n*-grammi)	10^{-2}	10	Lineare	Bilanciata
	SVM (WE)	10	10^{-2}	Polinomiale	Bilanciata

3.3 *Explainability*

Dato il kernel lineare dei SVM addestrati sul *feature set* estratto con *Profiling-UD*, è possibile stimare l'importanza degli attributi con i coefficienti della funzione lineare. In Figura 4 è visualizzata la *feature importance* per ciascun *task* dei primi 15 attributi associati ai coefficienti più grandi.

3.4 Valutazione dei sistemi

La valutazione dei sistemi è stata preliminarmente effettuata su TR per mezzo *5-fold CV*. La Tabella 3 riporta la media e l'incertezza dell'*accuracy* e dell'F1-macro di ciascun SVM sulle 5 iterazioni. Per la valutazione relativa delle metriche si considera una *baseline* che predice la classe più frequente (*baseline-mfc*).



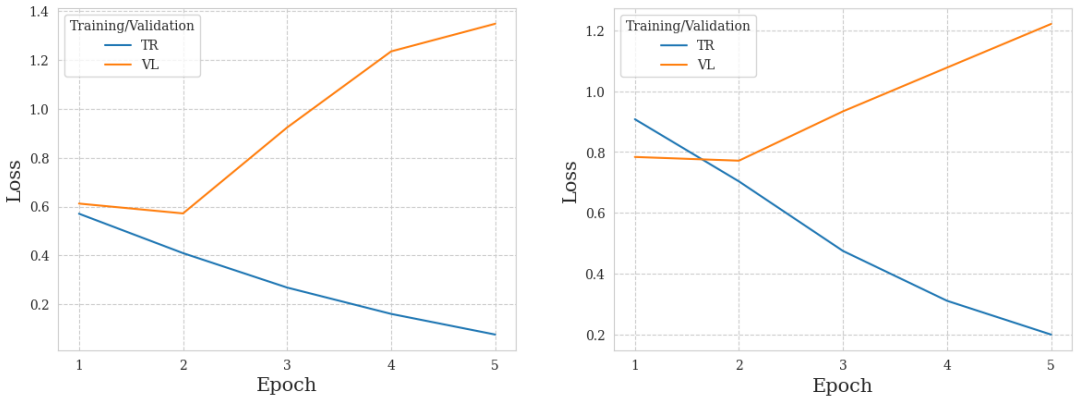
(a) *Task A* (b) *Task B*

Figura 4: Attributi del *feature set* estratto con *Profiling-UD* con maggiore *feature importance*

Tabella 3: Media e deviazione standard di *accuracy* e F1-macro (TS) di SVM su *5-Fold CV*.

	Modello	**F1-macro**	***Accuracy***
Task A	Baseline-mfc	0.34 ± 0.01	0.51 ± 0.01
	SVM (Profiling-UD)	0.66 ± 0.01	0.66 ± 0.01
	SVM (*n*-grams)	$\mathbf{0.74 \pm 0.01}$	$\mathbf{0.74 \pm 0.01}$
	SVM (WE)	0.70 ± 0.01	0.70 ± 0.01
Task B	Baseline-mfc	0.22 ± 0.00	0.49 ± 0.01
	SVM (Profiling-UD)	0.40 ± 0.01	0.52 ± 0.02
	SVM (*n*-grams)	$\mathbf{0.53 \pm 0.01}$	$\mathbf{0.55 \pm 0.01}$
	SVM (WE)	0.50 ± 0.01	0.53 ± 0.01

La Figura 5 riporta invece le curve della *loss* (TR e VL) relative al *fine-tuning* di BERT per ciascun *task*. In entrambi i casi si osserva un aumento della *loss* a partire dalla seconda epoca – dovuto probabilmente alla complessità del modello in rapporto ad un numero relativamente contenuto di dati – che tuttavia non corrisponde ad un concomitante peggioramento delle prestazioni su VL in termini di F1-macro (Tabella 4). Al fine di evitare un possibile *overfitting*, ciascun modello è stato ri-addestrato per sole due epoche.



(a) *Task* A (b) *Task* B

Figura 5: Curve di *loss* su TR e VL per ciascun *task*.

Tabella 4: F1-macro (VL) per ogni epoca

Epoca	**F1-macro**	
	***Task* A**	***Task* B**
1	0.673	0.563
2	0.750	0.589
3	0.746	0.596
4	0.761	0.603
5	0.763	0.597

La valutazione finale dei sistemi è stata effettuata sul TS ufficiale di IronITA 2018 con un semplice protocollo *hold-out*. Le metriche utilizzate per la valutazione sono l'F1 di ciascuna classe e l'F1-macro. E' stata calcolata anche l'*accuracy* dei modelli, la quale tuttavia non è stata considerata nella valutazione del *task* B per la presenza di uno sbilanciamento significativo nella distribuzione delle classi (Figura 1b).

Le Tabelle 5 e 6 riportano i risultati della valutazione dei sistemi rispettivamente per il *task* A ed il *task* B. BERT si rivela il modello più competitivo nello svolgimento di entrambi i *task*, seguito da SVM addestrato su *n*-grammi per il *task* A e da SVM addestrato su WE per il *task* B.

Tabella 5: Risultati (TS) del *task* A

Modello	**F1**			***Accuracy***
	non-ironia	**ironia**	**macro**	
Baseline-random	0.50	0.51	0.51	-
Baseline-mfc	0.67	0	0.33	-
Miglior partecipante	0.71	0.75	0.73	-
SVM (Profiling-UD)	0.65	0.68	0.67	0.67
SVM (*n*-grammi)	0.67	0.74	0.70	0.71
SVM (WE)	0.67	0.56	0.61	0.62
BERT	**0.72**	**0.77**	**0.74**	**0.74**

Tabella 6: Risultati (TS) del *task* B

Modello	**F1**				***Accuracy***
	non-ironia	**ironia**	**sarcasmo**	**macro**	
Baseline-random	0.50	0.27	0.24	0.34	-
Baseline-mfc	0.67	0	0	0.23	-
Miglior partecipante	0.67	0.45	0.45	0.52	-
SVM (Profiling-UD)	0.67	0.38	0.12	0.39	0.51
SVM (*n*-grams)	0.63	0.40	0.42	0.48	0.50
SVM (WE)	**0.70**	0.35	0.44	0.50	**0.56**
BERT	**0.70**	**0.42**	**0.47**	**0.53**	**0.56**

4 Conclusioni

In questa relazione ho presentato quattro diversi sistemi per il riconoscimento dell'ironia e del sarcasmo in testi italiani provenienti da *Twitter*. Come segnalato dalle prestazioni dei sistemi più competitivi, il riconoscimento dell'ironia e del sarcasmo risulta essere un *task* difficile da trattare automaticamente. La ragione è stata in parte suggerita dalla *feature selection* eseguita per *n*-grammi e WE, in cui è difficile isolare una parte dell'informazione che sia significativamente più rilevante alla discriminazione delle classi target. Ed in effetti la comprensione dell'ironia passa non solo per la ricognizione di informazione lessicale, ma

anche di informazione contestuale e pragmatica – e, non di rado, anche per l'accesso a conoscenze condivise non direttamente desumibili dal contesto linguistico. La performance dei sistemi presentati risulta essere comunque in linea con lo SOTA, come indicato dalle prestazioni del miglior sistema presentato nel contesto della competizione.

Riferimenti Bibliografici

Brunato D., Cimino A., Dell'Orletta F., Montemagni S. e Venturi G. (2020). Profiling–UD: a Tool for Linguistic Profiling of Texts. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, 7145–7151.

Cignarella A. T., Frenda S., Basile V., Bosco C., Patti V. e Rosso P. (2018). Overview of the EVALITA 2018 Task on Irony Detection in Italian Tweets (IronITA). In *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 18)*.

Cimino A., De Mattei L. e Dell'Orletta F. (2018). Multi-task Learning in Deep Neural Networks at EVALITA 2018. In *Proceedings of EVALITA '18, Evaluation of NLP and Speech Tools for Italian*, 12-13.

De Marneffe M. C. , De Lhoneux M., Nivre J. e Sebastian Schuster (2020). Proceedings of the Fourth Workshop on Universal Dependencies (2020). Association for Computational Linguistics, Barcelona.

Devlin, J., Chang, M. W., Lee, K., e Toutanova K. (2018). BERT: Pre-Traning of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805

Loshchilov I. e Hutter F. (2017). Decoupled Weight Decay Regularization. *arXiv preprint* arXiv:1711.05101.

Marchetti A., Massaro D. e Valle A. (2007). Non dicevo sul serio. Riflessioni su ironia e psicologia. *Collana di psicologia*, Franco Angeli.