UNIVERSITÀ DELLA CALABRIA

Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica (DIMES)

Laurea Magistrale in Ingegneria Informatica
Indirizzo Artificial Intelligence e Machine Learning

Definizione e valutazione di una metodologia per l'Hashtag Recommendation in Real-Time

Relatori:

Ing. Fabrizio Marozzo
Ing. Riccardo Cantini
Ing. Alessio Orsino

Candidato:

Saverio Azzato
Matr. 207012

Anno Accademico 2020/2021

Indice

1 Big Data e Social Media **4**

1.1 I Big Data 4

1.1.1 Caratteristiche dei Big Data e obiettivi delle analisi 6

1.1.2 I Social Media 8

1.2 Tecnologie e metodologie per Big Data 9

1.2.1 Acquisizione dei dati 11

1.2.2 Estrazione e pulizia dei dati 12

1.2.3 Immagazzinamento dei dati 14

1.2.4 Strumenti e tecnologie di analisi per Big Data 15

1.2.5 Metodologie per i Big Data in Real-Time 19

2 Tecniche per il Language Modeling e Hashtag Recommendation **24**

2.1 Introduzione al Natural Language Processing 25

2.2 Dalle RNN all'Attention 27

2.2.1 Architetture Encoder-Decoder 27

2.2.2 RNN 28

2.2.3 Gated Networks e LSTM 30

2.2.4 Attention e Bidirectional Attention 35

2.2.5 Holistic Attention e Transformers 39

2.3 Esplorazione del contesto e modelli alternativi 45

2.3.1 Hashtag Recommendation 45

2.3.2 Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU) 46

2.3.3 Temporal enhanced Sentence-level Attention Model (tSAM) 48

3 Tecnica proposta - HASHET Realtime **51**

3.1 Sentence Embedding 51

3.1.1 Bidirectional Encoder Representation from Transformer 52
3.1.2 (Google's) Universal Sentence Encoder 57
3.2 Word Embedding: Word2Vec 60
3.3 HASHET 63
3.4 HASHET-Realtime 66
3.4.1 Topologia Apache Storm 67
3.4.2 Funzionamento di HASHET in Real-Time 70
3.4.3 Discussione sulla scalabilità di HASHET-Realtime 71

4 Valutazione sperimentale **73**
4.1 Ispezione del dataset 73
4.2 Confronto delle metodologie per HASHET-Realtime 75
4.2.1 Transfer Learning e fine-tuning sulla Logic Window 76
4.2.2 Fine-tuning di MLP sulla FT Window 77
4.2.3 Addestramento da zero di MLP su Logic Window 78
4.2.4 Addestramento da zero di MLP su Logic Window e fine-tuning dell'intero modello su FT Window 78
4.2.5 Fine-tuning dell'intero modello su FT Window 80
4.2.6 Confronto delle metodologie 81
4.3 Confronto con gli altri modelli 86

Elenco delle figure

1.1 Crescita della circolazione di dati negli ultimi 2 decenni (grafico qualitativo) 5
1.2 Le 5 V dei Big Data 7
1.3 Le fasi dell'analisi di Big Data 10
1.4 Un esempio di risposta delle API di Twitter (JSON corrispondente ad un post) 13
1.5 Modello base illustrativo di MapReduce 18
1.6 Illustrazione esplicativa delle modalità di windowing 21
1.7 Illustrazione di un modello di topologia di Apache Storm 21
1.8 Modello di uno Storm Cluster 22

2.1 Modello della struttura di una RNN 28
2.2 Modello di RNN per evidenziare il caso in cui l'output sia lo stato finale . . . 29
2.3 Modello di architettura Encoder-Decoder basato su RNN 29
2.4 Modello di LSTM 31
2.5 Calcolo del cell state 32
2.6 Calcolo dell'output 33
2.7 Architettura Encoder-Decoder basata su LSTM 33
2.8 Visualizzazione dei problemi delle architetture basate su reti ricorrenti semplici e confronto con il meccanismo di Attention 35
2.9 Architettura Encoder-Decoder basata su RNN e Attention 36
2.10 Calcolo dei pesi, con dettaglio sul modello (in alto a destra) 37
2.11 Esempio esplicativo della tipologia di Attention desiderata 39
2.12 Architettura del Transformer [24] 40
2.13 Illustrazione semplificata del calcolo della Self-Attention (per un singolo input) 41
2.14 Scaled Dot-Product Attention [24] 42
2.15 Multi-Head Attention [24] 43

2.16 Esempio illustrativo della struttura delle due Mappe THFM (a sinistra) e HFM (a destra) [20] 47
2.17 Raffigurazione dell'architettura di tSAM [17] 48

3.1 Estrazione dell'Encoder da un'architettura Encoder-Decoder (a), con esempio nel caso del Transformer (b) 52
3.2 Architettura di BERT 53
3.3 Encoding degli input in BERT [8] 54
3.4 Esempio di fine-tuning di BERT per downstream task differenti [8] 55
3.5 Struttura dell'Universal Sentence Encoder 58
3.6 Task a valle per l'addestramento dell'Universal Sentence Encoder 60
3.7 Struttura di Word2Vec 61
3.8 Relazioni semantiche nello spazio di codifica di W2V 62
3.9 Illustrazione del processo di training del modello HASHET 63
3.10 Illustrazione del processo di Recommendation del modello HASHET 64
3.11 Esempio di Local e Global n-NHE Expansion ($k = 2$ e $n = 1$) [5] 65
3.12 Struttura completa di HASHET-Realtime 67
3.13 Topologia Apache Storm utilizzata in HASHET-Realtime 68
3.14 Procedura di riaddestramento in tempo reale in HASHET 71
3.15 Topologia Storm estesa per coprire la scalabilità 72

4.1 Suddivisione del dataset 73
4.2 Distribuzione dei tweet per numero di hashtag per i 3 dataset (Completo, Statico, Real-Time) 74
4.3 Classifica Top 10 degli hashtag per i 3 dataset (Completo, Statico, Real-Time) 75
4.4 Riaddestramento del modello m0 76
4.5 Riaddestramento del modello m1 77
4.6 Riaddestramento del modello m2 78
4.7 Riaddestramento del modello m3 79
4.8 Riaddestramento del modello m4 80
4.9 Confronto delle diverse metodologie per HASHET sulla misura $F_1 - score$, raggruppate per settimana 81
4.10 Confronto delle diverse metodologie per HASHET sulla misura *Precision*, raggruppate per settimana 81

4.11 Confronto delle diverse metodologie per HASHET sulla misura *Recall*, raggruppate per settimana 82
4.12 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura $F_1 - score$, raggruppate per settimana 82
4.13 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Precision*, raggruppate per settimana 83
4.14 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Recall*, raggruppate per settimana 83
4.15 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura $F_1 - score$ 84
4.16 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Precision* 84
4.17 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Recall* 84
4.18 Confronto tra le classifiche dei top 10 hashtag per il Dataset Statico, per la Logic Window corrispondente all'ultimo riaddestramento e per il 17 ottobre . 85
4.19 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Recall* per $n = 0$ 86
4.20 Confronto tra la metodologia HASHET originale e HASHET-Realtime sulla misura *Recall* per $n = 5$ 86
4.21 Confronto tra HASHET-Realtime e le altre tecniche sulla misura $F_1 - score$, raggruppate per settimana 87
4.22 Confronto tra HASHET-Realtime e le altre tecniche sulla misura *Precision*, raggruppate per settimana 87
4.23 Confronto tra HASHET-Realtime e le altre tecniche sulla misura *Recall*, raggruppate per settimana 87

Introduzione

Lo scopo di questo elaborato è quello di esplorare e valutare un nuovo modello di Hashtag Recommendation per applicazioni Real-Time. Nei Social Media i post possono essere etichettati e arricchiti con l'utilizzo di hashtag, che, a loro volta, permettono la categorizzazione e la ricerca dei post stessi nella piattaforma. Un problema che spesso riscontrano gli utenti è la difficoltà nello scegliere gli hashtag giusti per un post, che risulta spesso in scelte non ottimali, non correlate all'argomento in questione oppure alla publicazione del post senza nessun hashtag. Da ciò nasce la necessità di suggerire gli hashtag più probabili, in base alla pertinenza rispetto all'argomento del testo al quale vengono associati. Per rispondere a questa necessità, si utilizzano i modelli di Hashtag Recommendation (suggerimento di hashtag). La proposta suggerita è quella di presentare un modello che possa effettuare questo task, riadattandosi in tempo reale ai cambiamenti nei topic di discussione che avvengono nella piattaforma.

Il modello in questione è un'estensione di un modello preesistente: HASHET (HAshtag recommendation using Sentence-to-Hashtag Embedding Translation) [5]. L'architettura è composta a sua volta di modelli di Sentence e Word Embedding, utilizzati rispettivamente per generare l'embedding del testo di ciascun post e gli embedding dei suoi hashtag. Per ciascun post viene infine usato un modello di mapping per produrre dei vettori target nello spazio di Embedding degli hashtag dal testo e dei meccanismi di espansione tramite ricerca dei vicini (Nearest Hashtags Expansion) su questo target per suggerire gli hashtag migliori. L'estensione proposta include proprio il riadattamento, in tempo reale, alla variazione dei topic appena discusso. Il modello sarà valutato in scenari Real-Time, in competizione con alcuni dei modelli stato dell'arte più rilevanti nello stesso ambito.

I domini applicativi e gli scenari che interessano i task di Hashtag Recommendation sono svariati, come l'utilizzo discusso per i Social Media, la categorizzazione automatica di e-mail e il suggerimento di tag per contenuti, come note e blog post fino anche all'assegnamento di etichette ad una galleria di film e serie televisive, in base alla descrizione. Per la speri-

mentazione, è stato analizzato un caso di studio generato da dati provenienti da Twitter, più in particolare, concentrandosi sulle elezioni americane del 2020 (sugli hashtag relativi ai diversi candidati). Il modello proposto ha ottenuto risultati ottimali, superando anche le performance dei modelli con i quali è stato messo a confronto. L'estensione in Real-Time si è mostrata efficace nel riadattare il modello alle variazioni di topic, mantenendo risultati elevati anche con l'avanzamento del tempo negli esperimenti.

Il seguente elaborato è organizzato in quattro capitoli. Nel Capitolo 1 verrà esplorato in dettaglio il dominio applicativo di riferimento, offrendo un'introduzione ai Big Data, ai Social Network e, più in particolare, a Twitter. Un'ulteriore sfida aggiunta all'analisi di Big Data è quella di svolgere tali analisi in Real-Time. Questo comporta un maggior livello di attenzione per quanto riguarda l'efficienza e la parallelizzazione dei modelli utilizzati, nonchè sui sistemi di elaborazione scelti per i task. Le problematiche relative alle analisi in tempo reale e le considerazioni fatte in merito saranno discusse sempre nel primo capitolo.

Si andrà inoltre ad esplorare in dettaglio il contesto delle analisi, per comprendere meglio la natura e gli obiettivi dei task di riferimento. Il Capitolo 2 servirà quindi ad introdurre e a spiegare questi contesti, partendo dal campo del Natural Language Processing (NLP), del quale verranno illustrati gli obiettivi e i progressi più significativi. Infine, dopo aver introdotto i contesti di Recommendation System e Real-Time Hashtag Recommendation, questo capitolo si concluderà con una rassegna sintetica dei principali lavori presenti in letteratura, riguardo i task di Real-Time Hastag Recommendation.

Dopo aver concluso l'introduzione ai principali contesti applicativi di questo studio, nel Capitolo 3 verrà presentato il modello proposto. Si inizierà introducendo i modelli di Deep Learning sui quali si basa (BERT, GUSE e Word2Vec), per poi presentare l'architettura completa, spiegandone il funzionamento e la struttura e continuando con l'estensione in Real-Time. Questa seconda parte illustrerà tutte le scelte progettuali effettuate per applicare il modello su uno stream di dati in tempo reale, mostrando quali modifiche sono state eseguite per l'adattamento alla nuova dinamica di esecuzione e quali sistemi e strutture supportino tale funzionamento. Verrà quindi esplorata la differenza del modello HASHET statico e quello in Real-Time, mostrando la struttura completa del framework utilizzato. Il tutto verrà integrato con una spiegazione dei dettagli implementativi e del tuning dei modelli e dei parametri.

Il Capitolo 4 sarà dedicato alla valutazione sperimentale (eseguita sui dati delle elezioni USA del 2020). Verranno qui presentate le prestazioni e i risultati ottenuti dagli esperimenti

eseguiti sul modello proposto, mostrandone le variazioni rispetto ai parametri e alle configurazioni scelte. Sarà qui inoltre presentato un paragone dei risultati ottenuti con HASHET, rispetto ad alcuni dei modelli più di successo nella letteratura.

L'elaborato terminerà con una sezione conclusiva, nella quale saranno discussi riassuntivamente i risultati ottenuti e le considerazioni tratte da essi e dallo sviluppo dell'intero progetto, concentrandosi sugli obiettivi raggiunti e sui risvolti che possono seguire questo elaborato.

Bibliografia

[1] Khalid Adam. «Big Data Analysis and Storage». In: set. 2015.

[2] Sanjeev Arora, Yingyu Liang e Tengyu Ma. «A Simple but Tough-to-Beat Baseline for Sentence Embeddings». In: *ICLR*. 2017.

[3] Dzmitry Bahdanau, Kyunghyun Cho e Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[4] Beyer. «Gartner Says Solving "Big Data" Challenge Involves More Than Just Managing Volumes of Data». In: *gartner.com* (giu. 2011).

[5] Riccardo Cantini et al. «Learning Sentence-to-Hashtags Semantic Mapping for Hashtag Recommendation on Microblogs». In: *ACM Trans. Knowl. Discov. Data* 16.2 (set. 2021). ISSN: 1556-4681. DOI: 10.1145/3466876. URL: https://doi.org/10.1145/3466876.

[6] Daniel Cer et al. «Universal Sentence Encoder». In: *CoRR* abs/1803.11175 (2018). arXiv: 1803.11175. URL: http://arxiv.org/abs/1803.11175.

[7] Kyunghyun Cho et al. «Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation». In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078.

[8] Jacob Devlin et al. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[9] Huadong Guo et al. «Scientific big data and Digital Earth». In: *Chinese Science Bulletin (Chinese Version)* 59 (dic. 2014), p. 1047. DOI: 10.1360/972013-1054.

[10] Sepp Hochreiter e Jürgen Schmidhuber. «Long Short-term Memory». In: *Neural computation* 9 (dic. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[11] Mohit Iyyer et al. «Deep Unordered Composition Rivals Syntactic Methods for Text Classification». In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, lug. 2015, pp. 1681–1691. DOI: 10.3115/v1/P15-1162. URL: https://aclanthology.org/P15-1162.

[12] Mandar Joshi et al. «SpanBERT: Improving Pre-training by Representing and Predicting Spans». In: *CoRR* abs/1907.10529 (2019). arXiv: 1907.10529. URL: http://arxiv.org/abs/1907.10529.

[13] Seyed Mehran Kazemi et al. «Time2Vec: Learning a Vector Representation of Time». In: *CoRR* abs/1907.05321 (2019). arXiv: 1907.05321. URL: http://arxiv.org/abs/1907.05321.

[14] Guillaume Lample e Alexis Conneau. «Cross-lingual Language Model Pretraining». In: *CoRR* abs/1901.07291 (2019). arXiv: 1901.07291. URL: http://arxiv.org/abs/1901.07291.

[15] Yinhan Liu et al. «RoBERTa: A Robustly Optimized BERT Pretraining Approach». In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[16] Minh-Thang Luong, Hieu Pham e Christopher D. Manning. «Effective Approaches to Attention-based Neural Machine Translation». In: *CoRR* abs/1508.04025 (2015). arXiv: 1508.04025. URL: http://arxiv.org/abs/1508.04025.

[17] Jun Ma et al. «Temporal enhanced sentence-level attention model for hashtag recommendation». In: *CAAI Transactions on Intelligence Technology* 3 (apr. 2018). DOI: 10.1049/trit.2018.0012.

[18] Tomas Mikolov, Wen-tau Yih e Geoffrey Zweig. «Linguistic Regularities in Continuous Space Word Representations». In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, giu. 2013, pp. 746–751. URL: https://aclanthology.org/N13-1090.

[19] Tomas Mikolov et al. «Efficient Estimation of Word Representations in Vector Space». In: *Proceedings of Workshop at ICLR* 2013 (gen. 2013).

[20] Eriko Otsuka, Scott A. Wallace e David Chiu. «A hashtag recommendation system for twitter data streams». In: *Computational Social Networks* 3.1 (mag. 2016), p. 3. ISSN: 2197-4314. DOI: 10.1186/s40649-016-0028-9. URL: https://doi.org/10.1186/s40649-016-0028-9.

[21] David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams. «Learning Representations by Back-propagating Errors». In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0. URL: http://www.nature.com/articles/323533a0.

[22] Chris Snijders, Uwe Matzat e Ulf-Dietrich Reips. «"Big Data" : Big Gaps of Knowledge in the Field of Internet Science». In: *International Journal of Internet Science* 7 (gen. 2012), pp. 1–5.

[23] Ilya Sutskever, Oriol Vinyals e Quoc V. Le. «Sequence to Sequence Learning with Neural Networks». In: *CoRR* abs/1409.3215 (2014). arXiv: 1409.3215. URL: http://arxiv.org/abs/1409.3215.

[24] Ashish Vaswani et al. «Attention Is All You Need». In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[25] Yonghui Wu et al. «Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation». In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144. URL: http://arxiv.org/abs/1609.08144.

[26] Zhilin Yang et al. «XLNet: Generalized Autoregressive Pretraining for Language Understanding». In: *CoRR* abs/1906.08237 (2019). arXiv: 1906.08237. URL: http://arxiv.org/abs/1906.08237.