Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Ingegneria Informatica

Metodi di identificazione per le Fake News

Laureando
Angelica Zanato

Relatore
Prof. Nicola Ferro

Anno Accademico 2018/2019

Indice

Introduzione

Il fenomeno delle Fake News è stato recentemente molto studiato in quanto visto come una grande minaccia per la società. Uno dei più noti episodi riguarda i mesi critici prima delle elezioni presidenziali avvenute negli Stati Uniti nel 2016. Come riportato da [83], uno degli articoli falsi discussi maggiormente durante le elezioni ha ottenuto più di otto milioni di condivisioni, reazioni e commenti su Facebook, mentre la notizia veritiera più discussa ottenne poco più di sette milioni nonostante fosse stata pubblicata da 19 maggiori siti di news. In figura 1 si riportano le 10 notizie false più influenti e condivise nel 2016 su Facebook e, come si può vedere, 6 di esse riguardano le elezioni. Tali notizie potrebbero aver influenzato i risultati presidenziali, anche se non se ne ha ancora avuto la conferma.

S/N	Fake News Headline	Category
1	Obama Signs Executive Order Banning The Pledge of Allegiance in Schools Nationwide	Politics
2	Woman arrested for defecating on boss' desk after winning the lottery	Crime
3	Pope Francis Shocks World, Endorses Donald Trump for President, Releases Statement	Politics
4	Trump Offering Free One-Way Tickets to Africa & Mexico for Those Who Wanna Leave America	Politics
5	Cinnamon Roll Can Explodes Inside Man's Butt During Shoplifting Incident	Crime
6	Florida Man dies in meth-lab explosion after lighting farts on fire	Crime
7	FBI Agent Suspected in Hillary email Leaks Found Dead in Apparent Murder-Suicide	Politics
8	RAGE AGAINST THE MACHINE To Reunite And Release Anti Donald Trump Album	Politics
9	Police Find 19 White Female Bodies In Freezers With “Black Lives Matter” Carved Into Skin	Crime
10	ISIS Leader calls for American Muslim Voters to Support Hillary Clinton	Politics

Figura 1: Fake News più influenti e condivise nel 2016 su Facebook [2]

Per dimostrare quanto le Fake News possano portare a conseguenze anche gravi, si riporta un esempio nel seguente messaggio in figura 2. Il tweet proviene dall’account ufficiale di Associated Press, la prima agenzia di stampa internazionale, e riporta “Ultim’ora: Due esplosioni alla Casa Bianca e Barack Obama è ferito”. Naturalmente la notizia causò preoccupazione e, inoltre, portò ad un crollo instantaneo della Borsa di New York, provocando una perdita di 136 miliardi di dollari agli investitori in circa due minuti [2].

La presenza di notizie non vere, però, non è una novità e trova le sue origini nel diciasettesimo secolo, sotto il nome di *propaganda*. Nel 1925, ossia negli anni in cui la comunicazione tra offici giornalistici iniziò tramite cavo, l’autenticità delle informazioni ricevute divenne una preoccupazione. Gli editori dovevano trovare il modo di verificare i fatti, in modo da ridurre

Figura 2: Tweet della notizia secondo la quale l'ex presidente americano Barack Obama era rimasto ferito durante delle esplosioni, messaggio proveniente da Associated Press il cui account era sotto attacco hacker [2].

il più possibile la probabilità di proporre notizie non veritiere [43]. In seguito, durante gli anni della Guerra Fredda, il fenomeno venne denominato *disinformazione* [19].

Nonostante abbiano una storia così lunga, le Fake News hanno ricevuto maggiore attenzione recentemente per via dei social network come Facebook e Twitter. Questi ultimi, infatti, hanno un ruolo importante sul modo in cui le persone comunicano e su come le notizie vengono consumate, grazie al facile accesso che gli utenti hanno tali piattaforme [81]. Se, da un lato, i social network facilitano la diffusione delle notizie, dall'altro causano grandi problemi, in quando le notizie spesso non risultano verificate ed è dimostrato che le persone non sono in grado di distinguere tra vero e falso.

Grazie all'esplosione dei social media e al più semplice divulgarsi delle notizie, nel 2016, dopo le elezioni presidenziali negli Stati Uniti, il termine "Fake News" ha visto un incremento di utilizzo del 365%, diventando parola dell'anno. Gli autori di [19] ritengono che tale aumento è prova dell'interesse che le persone posseditrici di social network hanno nei confronti delle notizie false. Secondo [84], la quantità di persone che si affidano ai social media per ottenere notizie risaliva nel 2017 al 67% nella popolazione americana. Queste piattaforme generano l'effetto *echo chamber* (o camera dell'eco) per cui le informazioni vengono spesso amplificate e rinforzate, grazie alla natura stessa dei social network che invitano gli utenti a interagire con la notizia tramite commenti e condivisioni.

Fattori psicologici e sociali ricoprono un ruolo fondamentale nella diffusione delle Fake News; è stato dimostrato che la nostra abilità nel riconoscere le notizie vere da quelle false è di poco superiore al caso: l'accuratezza si aggira intorno al 55%-58% [83]. Inoltre, gli individui tendono a fidarsi delle notizie (sia vere che false) dopo esserne stati esposti ripetutamente per via dell'*effetto di validità* o se i fatti presentati confermano una conoscen-

za pre-esistente, principio chiamato *conferma del bias*. Un ulteriore motivo che a volte spinge a credere alle Fake News è il cosidetto *effetto bandwagon*, secondo la quale le persone tendono a credere ciò a cui gli altri credono [84].

Dato la crescente diffusione delle Fake News, esistono alcuni siti che si occupano di identificare il valore di verità delle notizie e, secondo [52], tali organizzazioni si sono quasi triplicate nel 2017 rispetto al 2014. È importante notare, però, che sia questo genere di organizzazione svolge la funzione di fact checking manualmente e, dunque, non è adaguata a verificare l'enorme quantità di notizie che ogni giorno vengono condivise sui social media. Per questo, si sono svolti molti studi per distiguere le Fake News dalle notizie vere in maniera automatica, in modo da evitare la propagazione di quelle false e da ridurne gli effetti negativi.

La seguente tesi è organizzata come segue: nel capitolo 1 si forniranno delle definizioni per le Fake News accompagnate da una breve descrizione delle tipologie esistenti e delle intenzioni per cui l'autore ha scritto una data notizia; nel capitolo 2 si riassumeranno brevemente delle tecniche provenienti dai campi di Natural Language Processing, Reperimento dell'Informazione, Machine Learning e Deep Learning allo scopo di una migliore comprensione delle metodologie di identificazione presentate nei capitoli 4, 5 e 6, mentre nel capitolo 3 si presenterà il problema dei dataset e si descriveranno alcune delle organizzazioni che si occupano del campo. Infine, nel capitolo 7 viene effettuato un confronto dei risultati dei vari studi tramite una metrica denominata *Normalized Winning Number* e al capitolo 8 verranno proposte delle possibili direzioni per la ricerca nell'ambito delle Fake News.

Capitolo 1

Definizioni e Tipologie di Fake News

Nel seguente capitolo verranno definite le varie tipologie di Fake News esistenti e le caratteristiche che le contraddistinguono, a partire da una generale definizione del termine in sè.

1.1 Definizione di Fake News

L'utilizzo del termine *Fake News* si è evoluto con il tempo e, come già notato, ha ricevuto maggiore attenzione dopo le elezioni presidenziali degli Stati Uniti del 2016. Una definizione generica del termine è la seguente:

Definizione 1.1. Per Fake News si intende l'informazione presentata come notizia, ma di fatto scorretta e creata in maniera da indurre l'utente a credere che sia vera [3, 17, 61].

Questa definizione, però, è vista dagli autori di [59] come restrittiva in quando non permette di includere ogni tipo di informazione e intenzione con cui la Fake News è stata scritta o condivisa. Per questo, viene proposta una seconda definizione:

Definizione 1.2. Per Fake News si intende un articolo o un messaggio pubblicato e propagato dai media, contenente informazioni false indipendentemente dai mezzi e dalle motivazioni [59].

La precedente permette di catturare diversi tipi di Fake News esistenti e di includere le varie intenzioni, in quanto non tutte le notizie false vengono propagate con obiettivi ostili. Le motivazioni, infatti, possono essere tra le più varie: per profitto (dovuto alle visualizzazioni da parte di altri utenti), per influenzare e manipolare la pubblica opinione, per creare disordini, per passione o per divertimento.

1.2 Tipologie di Fake News

Come visibile in figura 1.1 le Fake News sono di diverse tipologie e hanno diverse motivazioni. Per questo motivo, si presentano ora le varie forme individuabili sul web. La lista è stata ricavata da [81] e non è da considerarsi completa rispetto alla varietà di notizie presenti nel Web.

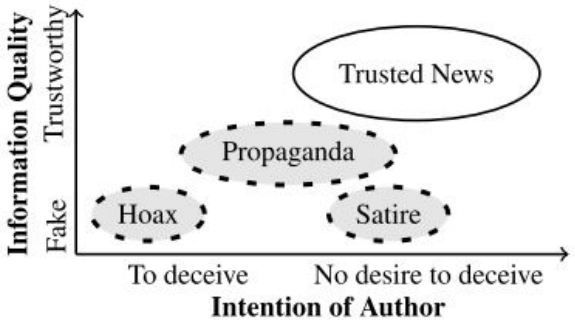


Figura 1.1: Alcuni tipi di notizia classificati in base alle intenzioni dell'autore e alla qualità delle informazioni [52].

- **Notizie Fabbricate**: Si tratta di storie completamente false e sconnesse dai fatti reali. Questo tipo di notizia non è nuova e la sua creazione risale alla nascita del giornalismo;

- **Propaganda**: Un particolare tipo di storia fabbricata con l'obiettivo di nuocere gli interessi di un particolare partito e solitamente tratta di argomenti politici. Anche questo genere non è nuovo, veniva spesso usato durante la Seconda Guerra Modiale e la Guerra Fredda. La propaganda è detta notizia falsa consequenziale in quanto può cambiare il corso della storia;

- **Teorie di Cospirazione**: Storie che cercano di spiegare situazioni o eventi invocando una cospirazione senza alcuna prova. Solitamente trattano di attività illegali avvenute e collegate in qualche modo al governo o a individui importanti. Generalmente, in questo genere di storia ci sono riferimenti a fatti senza fonte o si basano semplicemente sull'evidenza;

- **Bufale**: Si tratta di storie che riportano fatti falsi o non del tutto corretti presentandoli come reali. Uno dei temi più popolari di questo genere di notizia riguarda le presunte morti di varie celebrità e persone famose;

- **Notizie di Parte**: Solitamente di carattere politico, riporta notizie a favore di un particolare partito o persona;
- **Dicerie**: Si tratta semplicemente di storie la cui veracità è ambigua o mai confermata. Questo genere di notizia è piuttosto comune sui social network e per questo è stata molto studiata;
- **Clickbait**: Si intende dei titoli di articoli o delle immagini il cui semplice scopo è indurre l'utente a cliccarci. Anche questa tecnica non è nuova e trova la sua creazione nell'era dei quotidiani, con il nome di giornalismo giallo. Grazie al Web, però, si è diffuso notevolmente e il suo obiettivo è ottenere quante più visualizzazioni possibili, in modo da avere maggiore profitto o maggiore popolarità. Non si tratta, però, di un metodo dannoso in quanto, generalmente, se l'utente legge o guarda l'intero contenuto, si renderà conto facilmente che il titolo o l'immagine era ingannevole;
- **Satire**: Storie che contengono molta ironia e umorismo. Nel recente passato, hanno avuto una considerevole attenzione sul Web ed esistono molti siti pubblicano esclusivamente questo tipo di notizia (ad esempio, TheOnion). Generalmente, i siti satirici si rivelano come tali in una delle loro pagine, ma gli utenti che trovano la notizia condivisa sui social network spesso non lo vedranno. Per questo anche le notizie satiriche diventano reali agli occhi dell'utente se quest'ultimo non svolge ulteriori ricerche.

Le precedenti tipologie indicate non sono solitamente indipendenti l'una dall'altra; ad esempio, è possibile che una bufala abbia anche un titolo clickbait, in modo da attrarre ancora più persone ed indurle a credere a ciò che è scritto. Allo stesso modo la propaganda può essere spesso considerata anche una storia di parte [81].

1.3 Motivazioni

Come precedentemente affermato, non tutte le Fake News sono create e condivise con cattive intenzioni. Per questo motivo, si presenta ora una breve lista delle possibili motivazioni dietro a tali notizie, così come descritte da [81].

- **Cattive Intenzioni**: si riferisce a quelle Fake News che mirano a ferire gli altri e a danneggiare l'immagine pubblica di persone, organizzazioni o entità;
- **Influenza**: il creatore della notizia cerca di influenzare le decisioni delle persone o di manipolare l'opinione pubblica su un determinato argomento. Questa categoria può essere ulteriormente suddivisa in:

1. Ottenere maggiori follower, ossia avere più potere;
2. Cambiare l'opinione del pubblico, disseminando informazioni false. Questa seconda categoria è preoccupante in ambiente politico poiché gli individui cercano di accrescere l'immagine di una certa persona o di rovinare la reputazione del partito opposto condividendo informazioni non vere specialmente durante le elezioni, come è avvenuto nel 2016 per le elezioni presidenziali americane;

- **Seminare Discordia**: In particolari periodi di tempo, le persone o le organizzazioni cercano di creare confusione o discordia nel pubblico a loro vantaggio;

- **Profitto**: Una delle principali ragioni per cui vengono create le Fake News riguarda il profitto. Infatti, più una notizia viene condivisa, più genererà guadagno grazie alle pubblicità presenti. Di solito, questo genere di intenzione è accompagnato dall'utilizzo di clickbait per attrarre più utenti possibili;

- **Passione**: Molti utenti sono appassionati a qualcosa e questo influenza la loro capacità di giudizio per quell'argomento. Siccome tali persone sono spesso accecate dalle proprie ideologie, sarà più probabile che credano e condividano notizie sull'argomento che confermano il loro punto di vista;

- **Divertimento**: Online esistono degli individui chiamati "troll" che condividono e diffondono informazioni non vere per semplice divertimento.

Similmente alla sezione 1.2, le motivazioni non sono indipendenti le une dalle altre. Ad esempio, è possibile che una persona condivida informazioni false per ottenere influenza politica, ma allo stesso tempo anche perché è appasionato dell'ambito.

Capitolo 2

Concetti di Base

Nel seguente capitolo, si forniranno alcune nozioni di base per le tecniche più utilizzate nel campo delle Fake News, provenienti dagli ambienti di Reperimento dell'Informazione (IR), Natural Language Processing (NLP), Machine Learning e Deep Learning, allo scopo di assicurare una maggiore comprensione delle tecniche che verranno presentate in seguito.

2.1 Basi di Reperimento dell'Informazione

Per Reperimento dell'Informazione (IR) (in inglese, Information Retrieval) si intende l'insieme di tecniche utilizzate per gestire la rappresentazione, la memorizzazione, l'organizzazione e l'accesso ad oggetti contenenti informazioni, ad esempio documenti, pagine web e oggetti multimediali allo scopo di soddisfare il *bisogno informativo dell'utente* [10]. Dunque, l'obiettivo dell'IR è fornire i documenti e le informazioni utili a rispondere alle esigenze dell'utente dopo che quest'ultimo ha espresso una richiesta, detta *query*. Lo schema in figura 2.1 riassume brevemente il processo del reperimento.

2.1.1 Indicizzazione

Una delle fasi essenziali del reperimento dell'informazione è denominato *indicizzazione*, ossia il processo che, dopo aver esaminato i documenti, produce una lista di termini indice presenti nei documenti stessi e ogni termine sarà collegato al documento di provenienza. L'indicizzazione si sviluppa su più step che solitamente vengono usati anche nell'analisi delle Fake News. Tali passi sono:

1. Analisi lessicale: il documento viene scansionato completamente e vengono generati dei token rappresentati i potenziali descrittori del documento;

2. Rimozione delle stop word: una volta individuati i token, alcuni non hanno valore descrittivo in quanto si tratta di parole funzionali (o

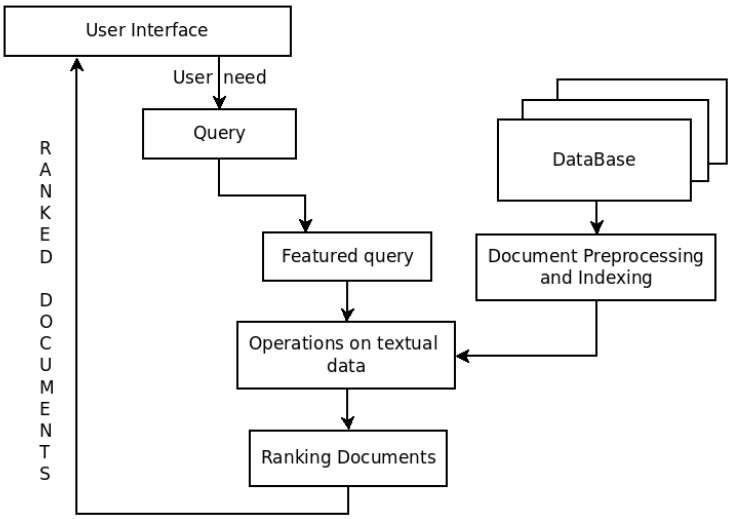


Figura 2.1: Rappresentazione del processo di Reperimento dell'Informazione.

stopword), ossia parole che di scarso contenuto informativo se usate da sole. Questo genere di token viene generalmente rimosso grazie all'utilizzo di stop list, liste contenenti le stopword;

3. Stemming: si occupa di ridurre le parole rimanenti alle proprie radici, in maniera da diminuire le forme differenti di una parola. Lo strumento che si occupa di questa fase è detto stemmer e il più noto è il Porter Stemmer [48];

4. Composizione termini: talvolta, risulta utile considerare due o più parole insieme, perciò è possibile comporre i token.

Per comprendere quali sono i termini più importanti in un documento o in una query, si applica una pesatura dei termini di cui la più caratteristica è $TF * IDF$, dove per TF si intende la frequenza di un termine nel documento/query e per IDF si intende l'inverse document frequency calcolata come segue:

$$IDF_t = \log\left(\frac{N}{df_t}\right)$$

dove N è il numero di documenti nella collezione e df_t è la frequenza del temine t nella collezione. Più raro è il termine nella collezione, più il suo IDF sarà alto.

2.1.2 Modelli

La rappresentazione delle query, dei documenti e la somiglianza tra i due sono generalmente descritti da un modello, ossia un insieme di costrutti

ideati e formalizzati a tale scopo. In IR, esistono vari tipi di modelli e i principali sono i seguenti:

- Modello booleano: la formulazione delle query avviene secondo la logica booleana utilizzando come operandi i descrittori dei documenti. In tal modo si recuperano solo i documenti che rendono vera la query e non viene fornito un ordinamento;
- Modello vettoriale: i documenti sono rappresentati sotto forma di vettori ed ogni elemento rappresenta un termine del documento. La similarità in questo caso può essere valutata in vari modi tra cui:
  - Prodotto interno tra vettori;
  - Cosine Similarity: rappresenta la distanza dei vettori in base all'angolo compreso tra loro: minore l'angolo, maggiore la similarità. Questa misura di similarità è spesso utilizzata nelle tecniche di identificazione di Fake News;
- Modello probabilistico: crea una stima della probabilità che un documento sia rilevante alla query. Esistono varie formule per valutare tale probabilità e la più famosa è la BM25 [55].

2.1.3 Word Embedding

Un ultimo importante concetto spesso utilizzato in IR riguarda i *word embedding*, ossia un insieme di tecniche di modellazione in cui parole o frasi di un vocabolario vengono mappate in vettori di numeri reali mantenendone le proprietà. Gli embedding sono utili poiché fornisco una rappresentazione più compatta e solitamente sono migliori a catturare sinonimi, ma risultano più difficili da spiegare. I word embedding più conosciuti ed utilizzati sono:

- *Word2vec*: è composto da una rete neurale (si veda sezione [**?**]) a due strati progettata per elaborare il linguaggio naturale. L'algoritmo richiede in ingresso una collezione di documenti e restituisce un insieme di vettori che rappresentano la distribuzione semantica delle parole nel testo. Per ogni parola contenuta nella collezione, in modo univoco, viene costruito un vettore in modo da rappresentarla come un punto nello spazio multidimensionale creato. In questo spazio le parole saranno più vicine se riconosciute come semanticamente più simili[1];
- *GloVe*: è costruito da un algoritmo di unsupervised learning e mappa le parole in uno spazio dove le distanze tra le parole è rappresentativa della loro similarità. La differenza principale da Word2vec è che GloVe considera le statistiche di tutta la collezione e mantiene la co-occorrenza delle parole nell'embedding;

[1] https://it.wikipedia.org/wiki/Word2vec

- *fasttext*: si tratta di una tecnica ideata dal Facebook Research Team e, al contrario di Word2vec e GloVe, assume che le parole siano composte di n-grammi, ossia da sottosequenze di n elementi. Grazie a questa assunzione, è in grado di trovare la rappresentazione di parole rare dato che anch'esse condividono n-grammi con altri termini.

2.1.4 Misure delle performance

Esistono principalmente due modi per valutare un sistema di Reperimento dell'Informazione:

1. **Efficienza**: come in altri ambienti informatici, è importante valutare le performance in termini di tempo e spazio;

2. **Efficacia**: siccome lo scopo è rispondere ai bisogni dell'utente, è necessario misurare la rilevanza che i risultati hanno rispetto alla query dell'utente. Alcune delle metriche più utilizzate sono:

   - **Precisione**: misura la proporzione di documenti pertinenti tra quelli recuperati. Detto A l'insieme dei documenti rilevanti e B l'insieme di quelli recuperati, la precisione è definita come:

   $$P = \frac{|A \cap B|}{|B|}$$

   - **Recall**: misura la porzione di documenti rilevanti che sono stati effettivamente recuperati. La recall è definita da:

   $$R = \frac{|A \cap B|}{|A|}$$

   - **F-measure**: si tratta della media armonica di precisione e recall, in modo da ottenere solo un valore per valutare le performance del sistema:

   $$F = 2 \cdot \frac{P \cdot R}{P + R}$$

   A volte, la F-measure viene denominata anche F1-score o F-score.

Le precedenti metriche sono utilizzate nella rilevanza binaria di un articolo ed esistono altre metriche per valutare un sistema di reperimento, qui non riportate poiché non rilevanti ai fini della tesi.

2.2 Basi di Machine Learning

Il Machine Learning, conosciuto anche come apprendimento automatico, è un ramo dell'intelligenza artificiale che comprende diversi ambiti: statistica

computazionale, riconoscimento di pattern, reti neurali artificiali, filtraggio adattivo, teoria dei sistemi dinamici, elaborazione delle immagini, data mining, algoritmi adattivi, ecc [5, 54].

Un algoritmo di Machine Learning è un algoritmo in grado di imparare dai dati. Imparare, in questo caso, significa che "un computer è in grado apprendere dall'esperienza E con riferimento ad alcune classi di task T e con misurazione della performance P, se le sue performance nel task T, come misurato da P, migliorano l'esperienza E" [38].

Nel seguito, saranno presentate alcune descrizioni intuitive di vari tipi di task, di misurazioni di performance e di esperienze che possono essere usate nella costruzione di algoritmi di Machine Learning così come definite dal libro [18]. Inoltre, verranno introdotti alcune delle tecniche più conosciute ed utilizzate.

2.2.1 Il Task, T

Il processo di apprendimento non è il task (o compito) in sè, ma imparare significa ottenere l'abilità di eseguire un determinato task. Ad esempio, se vogliamo che un robot sia in grado di camminare, allora camminare sarà il compito.

Esistono vari tipi di compiti:

- *Classificazione*: si chiede al programma di specificare a quale delle k categorie un dato input appartiene. Un esempio di questo tipo di task è descritto dalla identificazione di oggetti in un'immagine, il cui input è un'immagine e l'output è un codice numerico che identifica l'oggetto desiderato;

- *Classificazione con input mancanti*: la classificazione diventa più complessa se al programma non è garantito che ogni misura di input sia fornita; se alcuni input sono mancanti, invece di dare un'unica funzione di classificazione, l'algoritmo deve imparare un insieme di funzioni;

- *Regressione*: si chiede al programma di predirre un valore numerico dato un certo vettore di input, ossia si vuole una funzione f: $\mathbb{R}^n \to \mathbb{R}$. È simile alla classificazione, ma il formato di output è diverso. Un esempio è la predizione dei futuri prezzi dei titoli finanziari;

- *Trascrizione*: si domanda al sistema di osservare una rappresentazione relativamente non strutturata per un certo tipo di dati e di trascriverli in forma testuale. Un esempio è il riconoscimento ottico dei caratteri in cui viene mostrata al programma una fotografia contenente del testo e gli si chiede di trascriverla come sequenza di caratteri. Un altro importante esempio è il riconoscimento vocale;

- *Traduzione automatica*: dato un input di simboli in una data lingua, vogliamo che il programma lo converta in una sequenza di simboli in un'altra lingua. Solitamente è applicato ai linguaggi naturali;

- *Output strutturato*: contiene ogni task dove l'output non è più un singolo valore, ma un vettore o un'altra struttura contenente valori multipli. Un esempio è il parsing, ossia la mappatura di un linguaggio naturale in un albero che ne descriva la struttura grammaticale e ne etichetta i nodi come verbi, nomi, avverbi o altro;

- *Rilevazione di anomalie*: il programma vaglia su un insieme di eventi o oggetti e ne etichetta alcuni come inusuali o atipici. Un'applicazione è quella di scoprire un frode sulle carte di credito, in quanto gli acquisti fatti da un ladro avvengono con distribuzione di probabilità diversa dal proprietario della carta;

- *Sintesi e campionamento*: si chiede all'algoritmo di generare nuovi esempi simili a quelli dei dati di addestramento. Sintesi e campionamento possono essere utili per applicazioni multimediali; ad esempio, nei videogiochi è possibile generare automaticamente le texture di oggetti di grandi dimensioni o di paesaggi, invece di dover manualmente assegnare ogni pixel;

- *Denoising*: viene dato in input all'algoritmo un esempio corrotto $\widetilde{\boldsymbol{x}}$ ottenuto da un processo di corruzione sconosciuto e viene richiesto di ricostruire la versione originale dell'input $\boldsymbol{x}$ o, più in generale, di predirre la probabilità condizionata $p(\boldsymbol{x}|\widetilde{\boldsymbol{x}})$;

- *Stima di densità*: si domanda all'algoritmo di imparare una funzione p_{model} interpretabile come una densità di probabilità (o come distribuzione di massa se si tratta di una variabile aleatoria discreta). Per poterlo fare, l'algoritmo deve imparare la struttura dei dati che ha visto e deve sapere dove gli esempi si raggruppano e dove è raro che si trovino.

Naturalmente esistono molti altri tipi di compiti ed alcuni di quelli elencati possono essere specifici di altre aree di ricerca, come il Natural Language Processing (sezione 2.3).

2.2.2 La misurazione delle performance, P

Per poter valutare le abilità degli algoritmi di Machine Learning, è necessaria una misura delle sue prestazioni che spesso è specifica al compito che si vuole portare a termine.

Per task come la classificazione, solitamente si misura la precisione del modello; un'informazione equivalente si può ricavare dal tasso di errore, ossia

la proporzione di quanti esempi vengono mappati su un output sbagliato dal modello. Invece, per le stime di densità si usano altri metodi, il cui più comune è comunicare la media della log-probability che il modello assegna ad alcuni esempi [18].

Solitamente si è interessati a sapere quanto le prestazioni dell'algoritmo sono buone su dati che non mai visti prima. Per valutare queste performance, si usa un **test set** di dati separato da quelli che vengono usati per addestrare il sistema.

2.2.3 L'Esperienza, *E*

Gli algoritmi di Machine Learning possono essere categorizzati in *apprendimento supervisionato* e *non supervisionato* in base a che tipo di esperienza a cui esposti durante il processo di allenamento [18].

- *Apprendimento supervisionato*: gli algoritimi di questa categoria ricevono in input dei dati a cui è associata un'etichetta rappresentante l'output desiderato;

- *Apprendimento non supervisionato*: gli algoritmi di questo tipo si occupano di analizzare la struttura dei dati forniti in input allo scopo di impararne le caratteristiche senza la presenza di etichette.

A metà tra queste due categorie, vi è l'apprendimento semi-supervisionato in cui il dataset fornito per la fase di allenamento è incompleto, ossia alcuni dei dati presenti non sono etichettati con l'output desiderato.

Infine, esiste l'apprendimento per rinforzo in cui l'obiettivo è utilizzare le osservazioni ottenute interagendo con l'ambiente per fare delle azioni che massimizzerano il segnale di feedback, detto premio o segnale di rinforzo, o mininizzeranno il task. Gli algoritmi di questo tipo si dicono *agenti* e sono in continuo apprendimento dell'ambiente [14]. La figura 2.2 schematizza brevemente il processo dell'apprendimento per rinforzo, mentre la figura 2.3 organizza le varie categorie di Machine Learning.

2.2.4 Perceptron e Multilayer Perceptron

Il perceptron è un algortimo ricorsivo proposto da Frank Rosenbaltt e si tratta di un neurone artificiale che, composto con altri, può risolvere problemi complessi [1]. Il suo funzionamento è piuttosto semplice: può accettare dei dati di input e, in base ai pesi assegnati a questi input, viene generato un output pari a -1 o a 1 (figura 2.4), ed è perciò un algoritmo di classificazione binaria. Più nei dettagli, il perceptron accetta in input dei dati di training $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$, dove $\boldsymbol{x}_k$ è il vettore rappresentante il k-esimo elemento e y_k è la relativa etichetta; in seguito, un vettore di pesi $\boldsymbol{w}^{(1)}$ viene inizializzato a zero. All'iterazione t, il perceptron individua un

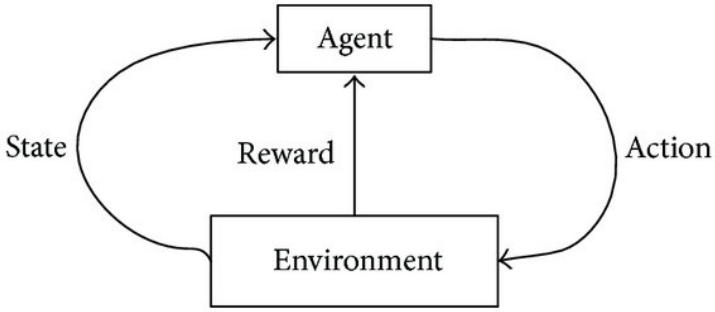


Figura 2.2: Schema del processo dell'apprendimento per rinforzo [14].

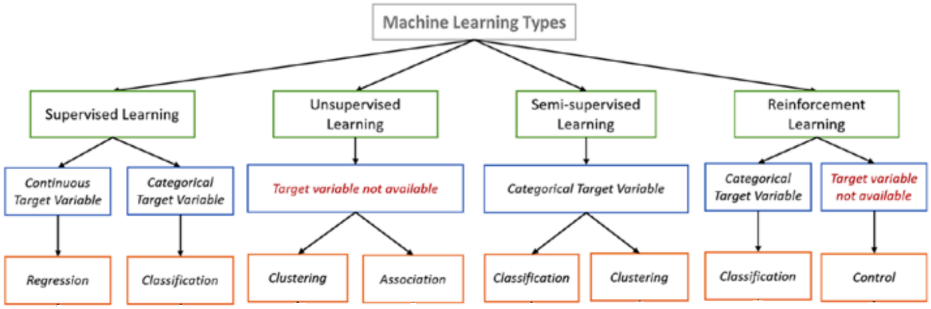


Figura 2.3: Suddivisione in categorie delle tecniche di Machine Learning [14].

elemento i etichettato erratamente dai pesi $\boldsymbol{w}^{(t)}$, ossia un elemento per cui $\text{sign}(\langle \boldsymbol{w}^{(t)}, \boldsymbol{x}_i \rangle) \neq y_i$. A questo punto, l'algoritmo aggiorna il vettore dei pesi come $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + y_i \boldsymbol{x}_i$. Il perceptron si ferma quando tutti gli elementi sono classificati correttamente ed è dimostrabile che ciò avviene in tempo determinato[58].

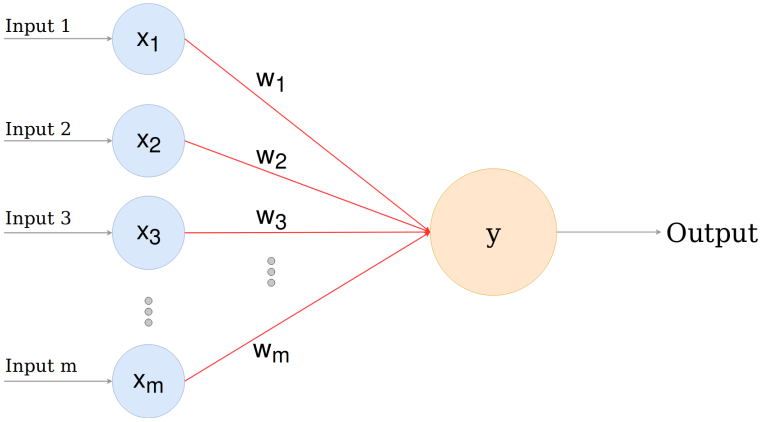


Figura 2.4: Semplice architettura descrittiva dell'algoritmo del perceptron [4].

Il multilayer perceptron (MLP) è una modifica al perceptron lineare ed è formato da più perceptron [1]. Un MLP è formato di strati multipli di nodi, ognuno completamente connesso al successivo come si può vedere in figura 2.5. Ogni perceptron invia segnali multipli e per ogni segnale si usano pesi diversi. Ogni livello può avere un grande numero di perceptron e ci possono essere multipli livelli. A volte ci si riferisce al MLP come rete neurale, ma le reti neurali sono una forma evoluta dei multilayer perceptron. Rispetto al perceptron classico, il MLP è in grado di portare a termine anche classificazioni multiclasse e task di regressione.

2.2.5 Modelli

In Machine Learning esistono vari modelli sviluppati per risolvere i task presentati in sottosezione 2.2.1, molti dei quali sono diffusi e utilizzati. Nel seguito si riassumono brevemente alcune di queste tecniche allo scopo di agevolare la comprensione delle metodologie atte all'identificazione delle Fake News.

- **Regressione Lineare**: la regressione lineare si occupa di individuare una relazione funzionale lineare presente nei dati. La relazione è data dalla migliore retta che interpola i dati ed è rappresentata dall'equazione lineare $Y = a \cdot X + b$, dove a e b sono i coefficienti, X è la variabile indipendente e Y è la variabile dipendente. Ve ne sono di due tipi:

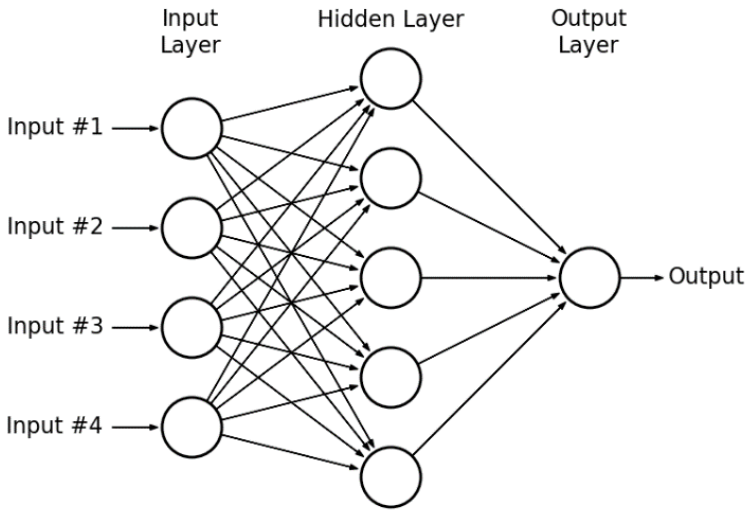


Figura 2.5: Esempio di multilayer perceptron [36].

regressione lineare semplice se si ha una sola variabile indipendente e regressione lineare multipla se si hanno più variabili indipendenti [53];

- **Regressione Logistica**: detto anche modello logit, si tratta di un modello di regressione non lineare il cui obiettivo è stabilire la probabilità con cui un'osservazione può generare uno o l'altro valore della variabile dipendente [65]. Tale valore viene utilizzato per un task di classificazione binaria;

- **Support Vector Machines**: solitamente abbreviate a SVM, si tratta di un insieme di metodi di apprendimento supervisionato utilizzati per i task di classificazione e regressione. Dato un set di esempi di training, si vuole determinare l'iperpiano o gli iperpiani che dividono tali esempi nelle categorie disponibili. Generalmente le SVM si utilizzano per modelli di classificazione lineare, ma possono anche essere usate efficientemente per classificazione non lineare con l'impiego di ciò che è conosciuto come *kernel trick* il cui scopo è mappare i suoi input a spazi di feature di maggiori dimensioni (vedi figura 2.6). Il kernel più comunemente utilizzato è il Radias Basis Function Kernel, usualmente abbreviato RBF;

- **Decision Tree Learning**: gli alberi di decisione sono usati in Machine Learning come modello predittivo per il valore target a partire dalle osservazioni su un oggetto. Se l'insieme dei valori di target è discreto, si parla di alberi di classificazione, in cui le foglie rappresentano le classi e i rami rappresentano le congiunzioni tra feature che portano a quelle classi (esempio in figura 2.7). Se la variabile target ha valori

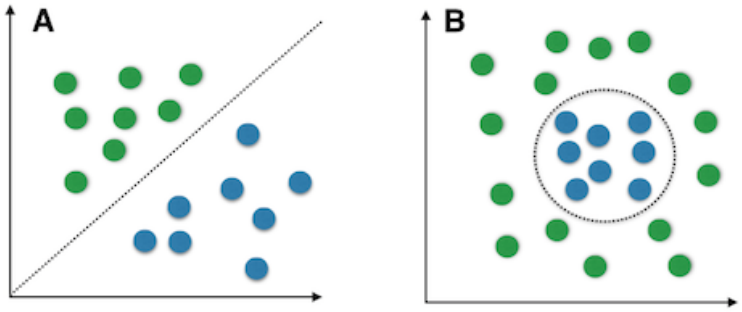


Figura 2.6: Esempio di utilizzo di SVM per uno spazio bidimensionale; a sinistra uno spazio linearmente divisibile, a destra uno spazio non linearmente separabile, risolvibile con l'utilizzo del kernel trick.

continui, allora si ha un albero di regressione. Nonostante siano facili da comprendere, gli alberi di decisione sono inclini all'overfitting, ossia il modello tenderà ad adattarsi molto bene agli esempi di training, ma le sue performance con esempi mai visti saranno basse;

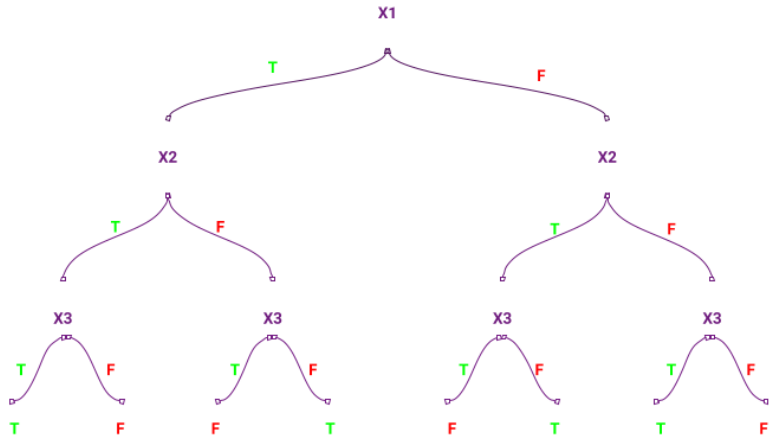


Figura 2.7: Esempio di albero di decisione per l'operazione XOR con tre operandi [21].

- **Random Forest**: le random forest sono un metodo di ensemble learning, ossia un metodo di learning che utilizza più algoritmi di apprendimento per ottenere migliori performance per la classificazione, la regressione e altri task che operano costruendo una moltitudine di alberi di decisione durante la fase di training. L'output di una Random Forest è la norma delle classi (o delle predizioni) degli alberi che

la compongono. La random forest è stata studiata per correggere la tendenza all'overfitting presentato dal Decision Tree Learning [13];

- **Naive Bayes**: il classificatore bayesiano si basa sul teorema di Bayes e richiede la conoscenza delle probabilità a priori e condizionali relative al problema, quantità che in generale non sono note, ma sono tipicamente stimabili. Se è possibile ottenere delle stime affidabili delle probabilità coinvolte nel teorema, il classificatore bayesiano risulta generalmente affidabile e potenzialmente compatto. In particolare, il classificatore è detto Naive se ha alla base l'ipotesi di indipendenza tra le feature;

- **k-Nearest Neighbor**: spesso abbreviato a kNN, si tratta di un algoritmo utilizzabile sia per task di classificazione che per la regressione [53]. Un oggetto è classificato in base alla maggioranza dei voti dei suoi k vicini, dove k è un intero positivo tipicamente non molto grande. La distanza dai vicini viene misurata solitamente con la distanza euclidea, distanza di Manhattan o distanza di Hamming. In figura 2.8 un semplice esempio di k-Nearest Neighbor;

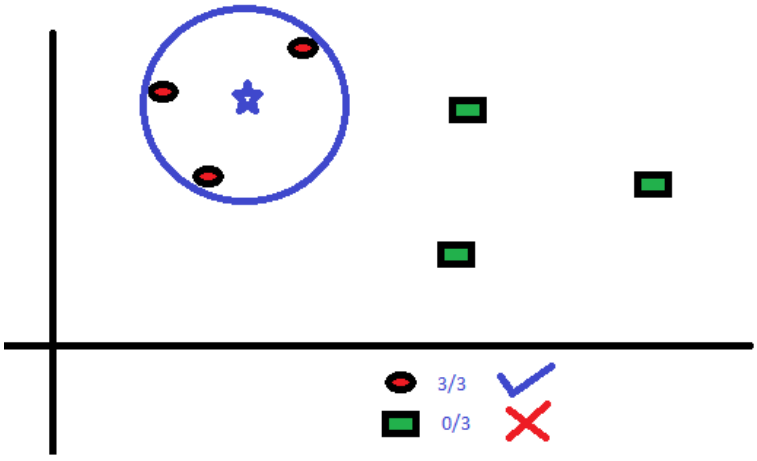


Figura 2.8: Esempio di utilizzo di kNN con k=3 [64]. Il punto indicato da una stella si trova più vicino a tre punti rossi rispetto ai rettangoli verdi. Per questo motivo, la stella sarà classificata sotto la stessa etichetta dei punti rossi.

- **k-Means**: si tratta di un algoritmo non supervisionato per risolvere un problema di clustering, cioè un problema di raggruppamento di elementi omogenei in un insieme di dati [53]. La procedura comincia con la creazione di k cluster e i dati di input vengono suddivisi in tali cluster casualmente o secondo delle regole euristice. Quindi, si calcolano i centroidi per ogni cluster e si ridividono i dati assegnando i punti

rimanenti in base alla loro distanza dai centroidi. Il processo viene ripetuto fino a convergenza, ossia finché il centroidi non cambiano. In figura 2.9 un esempio di come l'algoritmo di k-Means funziona;

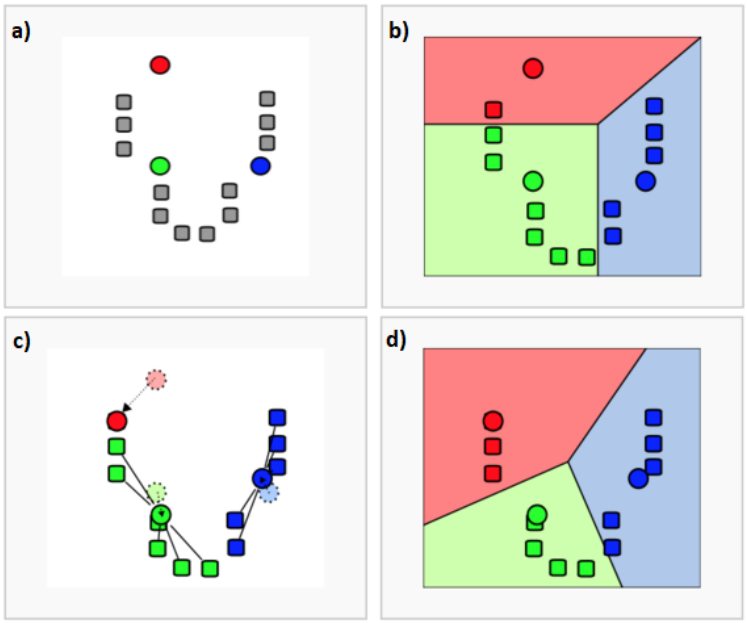


Figura 2.9: Esempio di utilizzo di k-Means [31] con k=3. a) k centri vengono scelti casualmente tra i dati di input; b) k cluster vengono generati in base alla distanza dal centroide più vicino; c) vengono calcolati i nuovi centroidi per ogni cluster; d) si generano i nuovi cluster e si procede iterativamente.

- **AdaBoost**: è un meta-algoritmo che utilizza in combinazione altre tecniche di apprendimento per migliorare le performance. L'output degli altri algoritmi (detti *weak learners*) viene combinato in una somma pesata rappresentante l'output finale del classificatore potenziato come visualizzato in figura 2.10. AdaBoost è adattivo nel senso che i weak learner vengono ottimizzati in maniera da favorire gli esempi classificati erratamente dagli altri classificatori. Anche se gli algoritmi di apprendimento utilizzati sono deboli, il risultato finale del modello converge ad un classificatore forte;

- **Bagging**: anche conosciuto con il nome *bootstrap aggregating*, si tratta di un meta-algoritmo il cui obiettivo è migliorare la stabilità e l'accuratezza degli algoritmi d'apprendimento per la classificazione e per la regressione. Dato un insieme di input di dimensione n, bagging crea m nuovi set di dimensione n' campionando i dati iniziali. Alcuni dei valori presenti nel set iniziale potrebbero essere ripetuti in più di uno dei nuovi set creati. In seguito, i nuovi set sono utilizzati per allenare

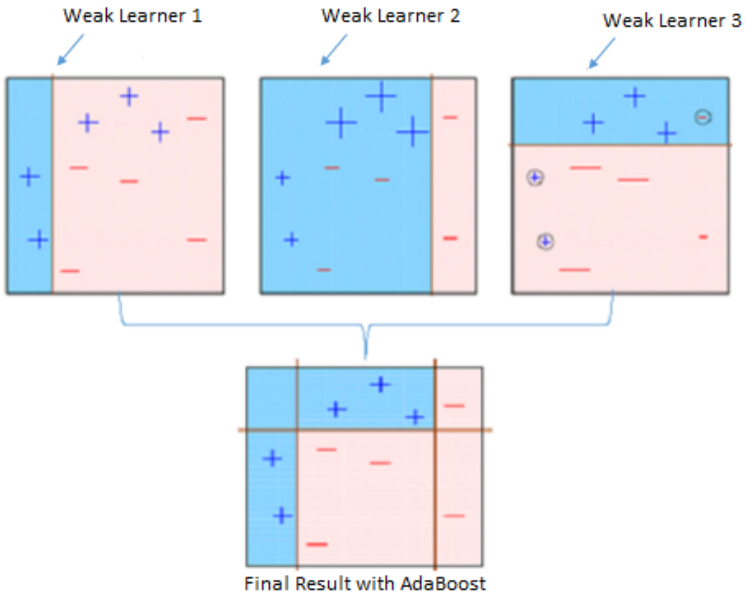


Figura 2.10: Schema rappresentate il funzionamento del meta-algoritmo AdaBoost [44].

dei modelli di Machine Learning e, infine, i risultati saranno combinati per generare il risultato finale. La combinazione è svolta tramite media se si sta lavorando ad un task di regressione e tramine un sistema di voto se si ha una classificazione. In figura 2.11 una rappresentazione di come il meta-algoritmo di Bagging lavora.

2.3 Basi di Natural Language Processing

Per Natural Language Processing (NLP si intende il processo di trattamento automatico mediante un calcolatore elettronico delle informazioni scritte o parlate in una lingua naturale tramite l'utilizzo di tecniche linguistiche, informatiche e di intelligenza artificiale. Gli studi in questo particolare campo sono iniziati generalmente negli anni cinquanta e le sfide principali di solito riguardano il riconoscimento vocale, la comprensione e la generazione di linguaggio naturale.

Inizialmente, i sistemi di NLP erano basati su delle regole programmate manualmente [74], ma con l'avvenimento della "rivoluzione statistica" [25] avvenuta tra gli anni ottanta e gli anni novanta, la ricerca per il Natural Language Processing si è concentrata maggiormente sull'utilizzo di modelli statistici e di Machine Learning, in quanto questi ultimi erano in grado di imparare automaticamente le regole tramite l'analisi di grandi collezioni

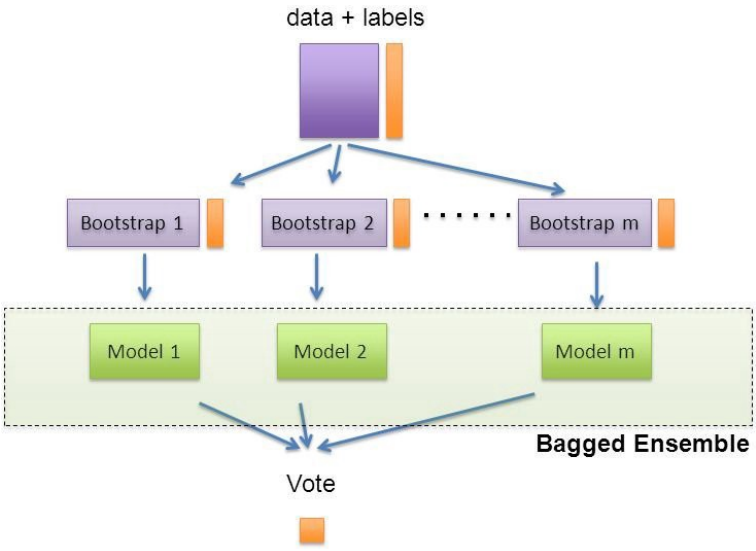


Figura 2.11: Schema rappresentate il funzionamento del meta-algoritmo Bagging [45].

di esempi. I modelli basati sul Machine Learning hanno alcuni vantaggi rispetto alle regole progettate manualmente:

- L'apprendimento automatico è in grado di concentrasi sui casi più comuni, mentre con le regole scritte a mano non è sempre ovvio dove gli sforzi debbano essere diretti;
- I modelli prodotti dal Machine Leaning sono generalmente robusti a input insoliti o scorretti, cosa estremamente difficile e dispendiosa in termini di tempo con i sistemi creati manualmente;
- Mentre i sistemi manuali migliorano solo modificando le regole, per i modelli basati sul Machine Learning è semplicemente necessario fornire più dati di input.

2.3.1 Categorie

Nonostante i task del NLP siano connessi tra loro, in genere vengono divisi in quattro categorie principali, a loro volta suddivise in base all'obiettivo da raggiungere (figura 2.12):

1. **Sintassi**: per sintassi si intende il posizionamento delle parole in una frase in maniera che abbiano senso grammaticalmente e, in NLP, l'analisi sintattica è usata per determinare come il linguaggio naturale si allinea alle regole grammaticali [15]. Alcune tecniche sintattiche sono le seguenti:

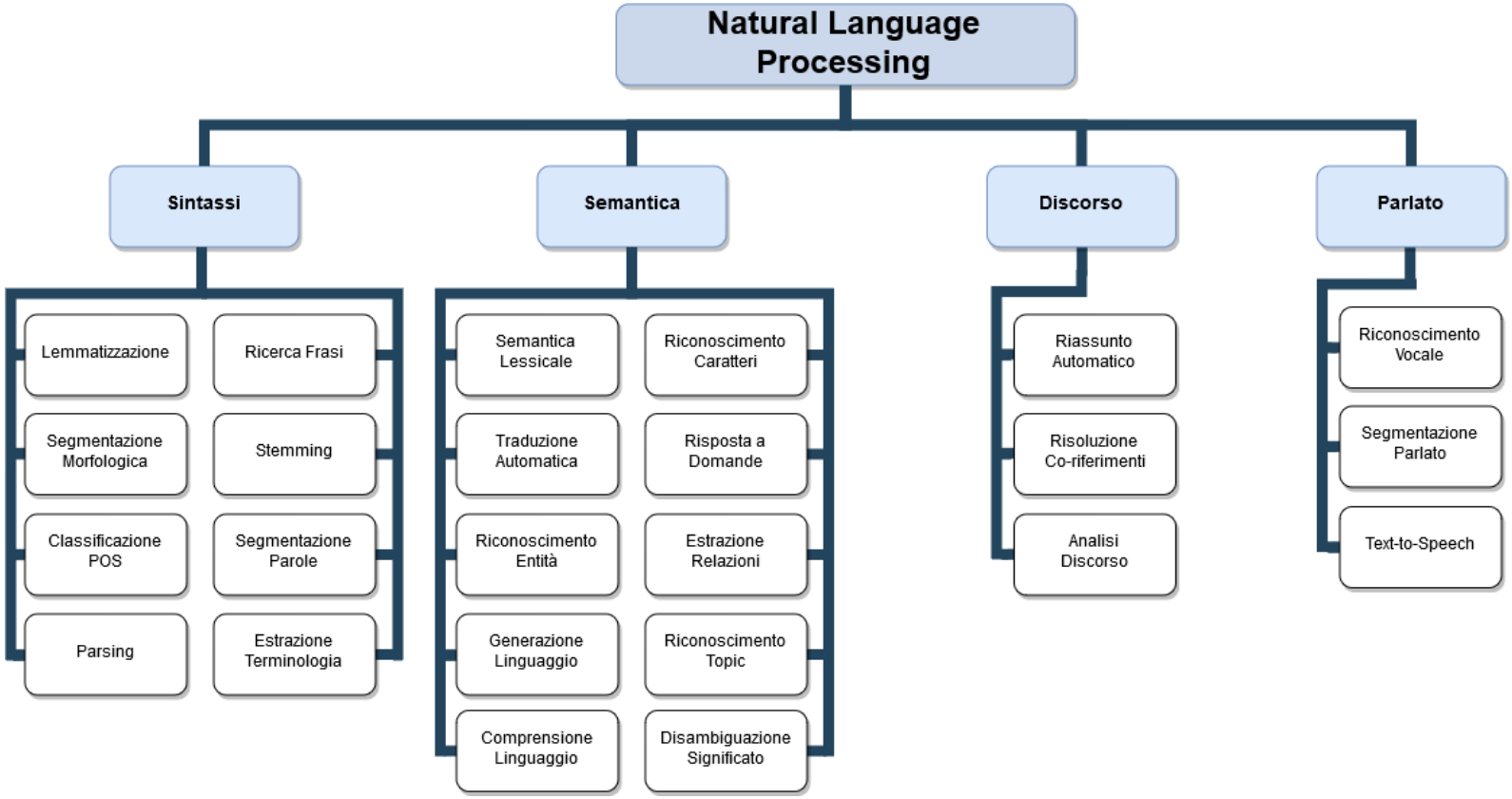


Figura 2.12: Suddivisione dei task studiati nel campo del Natural Language Processing.

- *Lemmatizzazione*: rimuove le flessioni nelle parole e restituisce la forma base, detta lemma;
- *Segmentazione Morfologica*: divide le parole in morfemi e ne identifica la classe;
- *Etichettare parti del discorso*: data una frase, si individuano le parti del discorso (POS) di ogni parola, ossia la funzione che quella parola ha nella frase;
- *Parsing*: data una frase, se ne determina l'albero sintattico;
- *Ricerca frasi*: dato un pezzo di testo, si individuano i limiti delle frasi, solitamente indicati da punti o altra punteggiatura, simboli usati anche per altri scopi;
- *Stemming*: come definito in sezione 2.1, si tratta del processo di riduzione delle parole alle loro radici;
- *Segmentazione delle Parole*: si occupa di separare un flusso di testo in parole singole;
- *Estrazione della Teminologia*: estrae automaticamente da una data collezione di testi i temini più rilevanti.

2. **Semantica**: per semantica si intende il significato contenuto in un testo e si tratta di uno degli aspetti più difficili del NLP [15]. Alcune delle tecniche semantiche sono:

   - *Semantica Lessicale*: individua il significato computazionale delle parole nel contesto;
   - *Traduzione Automatica*: si tratta di uno dei problemi più difficili e si occupa di tradurre un testo da una lingua ad un'altra;
   - *Riconoscimento di Entità con Nome*: dato un flusso di testo, si determina quali oggetti del testo mappare come nomi propri e che tipo di nome si tratta (se è nome di persona, di luogo, di organizzazione, ecc.);
   - *Generazione di linguaggio naturale*: converte l'informazione presa dai database dei computer in linguaggio umano;
   - *Comprensione di linguaggio naturale*: converte pezzi di testo in rappresentazioni più come strutture logice di primo ordine, più facilmente manipolabili dai programmi;
   - *Riconoscimento Caratteri*: data un'immagine contenente testo, determina il testo corrispondente;
   - *Risposta a domande*: data una domanda in linguaggio naturale, si occupa di determinare una risposta;
   - *Estrazione delle relazioni*: dato un pezzo di testo, si vogliono determinare le relazioni presenti tra le entità;

- *Segmentazione e Riconoscimento del topic*: dato un flusso di testo, si occupa di separarlo in segmenti riguardanti un topic e di individuare un topic;
- *Disambiguazione del significato*: siccome molte parole hanno più significato, ha l'obiettivo di determinare il significato più appropriato in base al contesto.

3. **Discorso**: detto in inglese *Discourse*, si suddivide in:
   - *Riassunto Automatico*: produce un riassunto leggibile di un pezzo di testo;
   - *Risoluzione di co-riferimenti*: data una frase o un testo, determina quali parole si riferiscono allo stesso oggetto. Include anche l'identificazione di relazioni ponte;
   - *Analisi del Discorso*: include una serie di task, tra cui l'identificazione della struttura discorsiva del testo.
4. **Parlato**: detto in inglese *Speech*, si occupa di riconoscimento del testo parlato e si suddivide in:
   - *Riconoscimento Vocale*: data una clip di una persona che parla, ne determina la rappresentazione testuale. Si tratta di un task difficile in quanto nel parlato non ci sono molte pause tra le parole e, in molte lingue, lettere successive vengono unite per aiutare la pronuncia, complicando la trascrizione;
   - *Segmentazione del Parlato*: data una clip di una persona che parla, si occupa di separarla in parole;
   - *Text-to-Speech*: task opposto al riconoscimento vocale, trasforma il testo in una rappresentazione parlata.

2.4 Basi di Deep Learning

Il Deep Learning, tradotto letteralmente come apprendimento profondo, è un ramo di Machine Learning e Intelligenza Artificiale e fa riferimento a quegli algoritmi, detti *reti neurali* la cui struttura e funzionalità sono inspirati al cervello umano. Queste reti neurali sono organizzate in diversi livelli (si veda figura 2.13) ed ogni livello si occupa di calcolare i valori del successivo affinché l'informazione venga elaborata in maniera più completa.

Secondo [6], il Deep Learning può essere definito come un sistema che sfrutta una classe di algoritmi di Machine Learning con le seguenti caratteristiche:

1. utilizza i livelli a cascata per svolgere compiti di estrazione di caratteristiche e trasformazione. L'output di un livello rappresenta l'input del livello successivo;

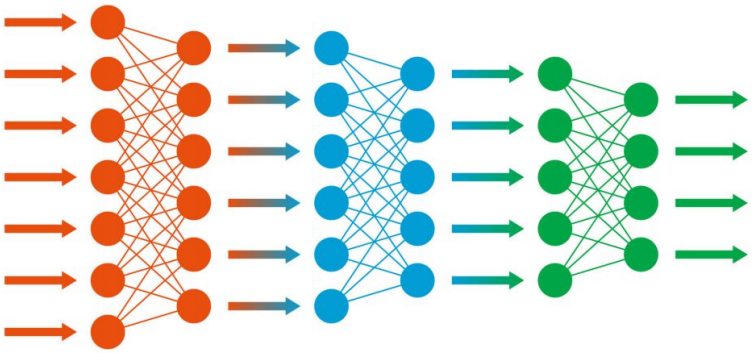


Figura 2.13: Esempio di rete neurale divisa in livelli [6].

2. è basato sull'apprendimento non supervisionato e le caratteristiche di un livello più alto sono derivate da quello più basso per rappresentare una gerarchia;

3. fa parte di una classe di algoritmi d'apprendimento della rappresentazione dei dati;

4. apprende diversi livelli di rappresentazione corrispondenti a livelli diversi di astrazione.

Il Deep Learning, dunque, punta a simulare il funzionamento del cervello umano tramite delle reti neurali artificiali le quali si presentano come un sistema "adattativo", in grado di modificare la sua struttura di nodi ed interconnessioni basandosi sia su dati esterni che su informazioni interne. I livelli delle reti neurali sono formati da neuroni, ossia da una struttura di base che accetta degli input e restituisce un output. Matematicamente, un neurone in Deep Leaning è semplicemente una funzione matematica [42], detta funzione di attivazione. I livelli interni di una rete normale sono spesso denominati livelli nascosti e una rete con più di un livello nascosto viene comunemente denominata profonda.

Le architetture di Deep Learning sono per esempio state applicate nella Computer Vision, nel riconoscimento automatico della lingua parlata, nell'elaborazione del linguaggio naturale, nel riconoscimento audio e nella bioinformatica [6].

2.4.1 Architetture

In letteratura si possono trovare molte architetture sviluppate nel campo del Deep Learning e nel seguito si elencheranno quelle più utilizzate nell'ambito di identificazione delle Fake News.

- **Reti Neurali Convoluzionali**: solitamente abbreviate a CNN si tratta di una variazione delle reti neurali utilizzata molto in Compu-

ter Vision. Generalmente, i livelli nascosti di una CNN sono convoluzionali, di pooling, livelli completamente connessi e di normalizzazione, come visualizzato in figura 2.14. I livelli convoluzionali utilizzano dei neuroni dove la funzione di attivazione è una convoluzione, mentre i livelli di pooling si occupano di ridurre la dimensione dei dati combinando gli output del livello precedente [42];

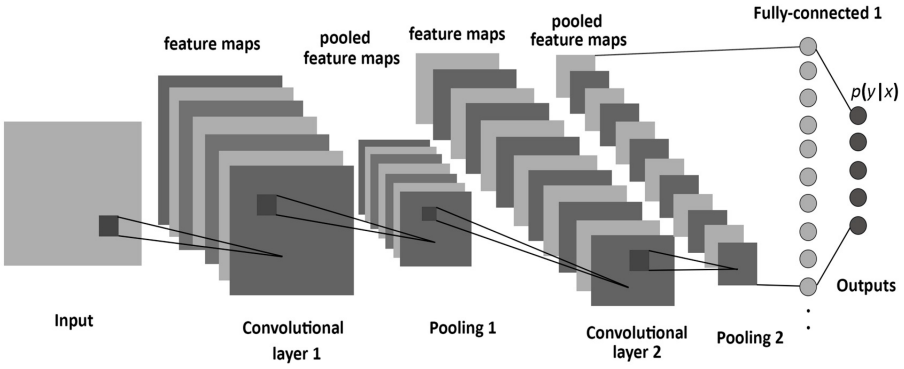


Figura 2.14: Esempio di rete neurale convoluzionale [42].

- **Reti Neurali Ricorrenti**: si tratta di una classe di reti neurali dove le connessioni tra i nodi formano un grafo diretto lungo una sequenza temporale [9]. Le Reti Neurali Ricorrenti (o RNN) possono usare il loro stato interno, detto memoria, per processare le sequenze di input, cosa che le rende utili a task come riconoscimento di scrittura manuale o riconoscimento vocale. Il termine RNN viene usato per riferirsi a due classi di reti con strutture simili, una con impulso finito e una con impulso infinito. Entrambe hanno un comportamento temporale dinamico. La rete a impulso finito è un grafo diretto aciclico che può essere srotolata e rimpiazzata da altre reti neurali, mentre quelle ad impulso infinito sono grafi diretti ciclici non srotolabili [37]. In figura 2.15 un esempio di come una rete neurale ricorrente è strutturata;

- **Long Short-Term Memory**: abbreviate a LSTM, si tratta di un tipo particolare di RNN, in grado di "ricordare" informazioni per periodi più lunghi di tempo per evitare il problema della dipendenza a lungo termine. Le RNN classiche sono in grado di tenere traccia delle dipendenze a lungo termine, ma per via della back-propagation utilizzata, i gradienti potrebbero sparire (ossia, tendono a zero) o esplodere (ossia, tendere all'infinito) per via delle computazioni svolte. Questo problema viene parzialmente risolto dall'architettura della LSTM poiché queste ultime permettono ai gradienti di fluire senza cambiare;

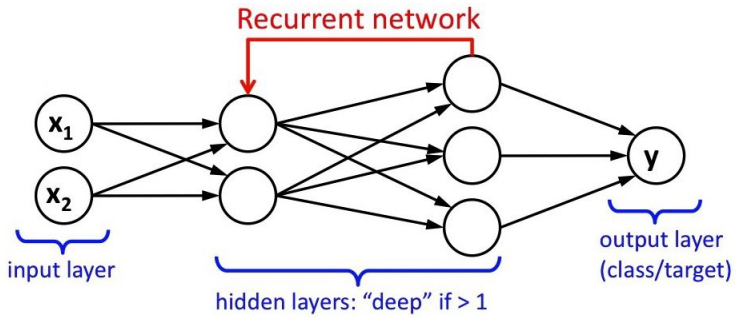


Figura 2.15: Esempio di rete neurale ricorrente [9].

2.4.2 Meccasmo di Attenzione

Un altro importante concetto spesso utilizzato in Deep Learning come in altri campi è il cosiddetto *meccanismo di attenzione*. Come suggerisce il nome, il suo scopo in Deep Learning è basato sul concetto di direzionare la concentrazione e di prestare maggiore attenzione a certi fattori nel processo di elaborazione dei dati. Un semplice esempio è presentato in figura 2.16: si vuole tradurre la frase "How was your day" dall'inglese al francese "Comment se passe ta journée"; il meccanisco di attenzione della rete farà in modo che ogni parola della sequenza di output sia mappata alle parole rilevanti della sequenza di input ed assegnerà maggior peso a tali parole, aumentando la precisione del output.

Il meccanismo di attenzione rappresenta una componente dell'architettura della rete neurale e ha il compito di quantificare l'**interdipendenza**, solitamente distinta nelle seguenti due categorie:

- Interdipendenza tra input e output, detta Attenzione Generale;
- Interdipendenza tra gli elementi di input, detta Self-Attention o semplicemente Attenzione.

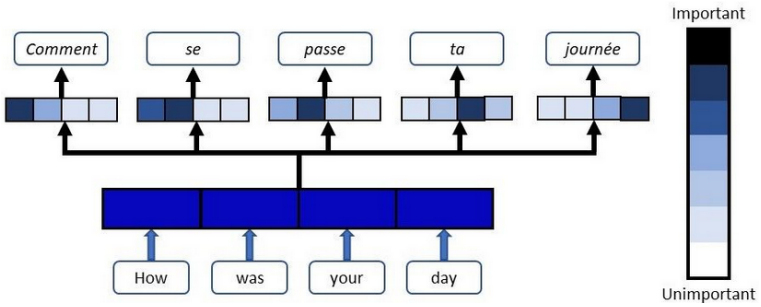


Figura 2.16: Meccanismo di attenzione: vengono assegnati dei pesi alle parole di input ad ogni passo della traduzione [33].

Capitolo 3

Dataset e Organizzazioni

In questo terzo capitolo viene introdotto uno dei problemi principali che la ricerca deve affrontare nello studio delle Fake News, la mancanza di dataset. Inoltre, in sezione 3.2 e 3.3 si presentano alcune delle organizzazioni più conosciute per il fact checking manuale e alcune delle challenge nate allo scopo di migliorare l'identificazione delle notizie false.

3.1 Dataset

Nell'ambito delle Fake News, per dataset si intende una collezione di articoli annotati dal loro grado di verità e forniti di informazioni riguardanti gli articoli stessi, come la sorgente, l'autore e il titolo. Sfortunatamente, il collezionamento di questo genere di dataset è uno dei maggiori problemi da affrontare in quanto non esiste un dataset consolidato per l'identificazione. I motivi alla base del problema sono vari ed includono:

- Diversità tra le Fake News: come presentato precedentemente in sezione 1.2, esistono molte sottocategorie di notizie false e, spesso, i dataset esistenti si concentrano solo su una o alcune di queste categorie;
- Si tratta di un ambito di ricerca piuttosto recente, perciò i dataset sviluppati con l'obiettivo di identificazione sono limitati;
- La quantità di notizie false rappresenta solo una frazione dei contenuti prodotti online ogni giorno [7];
- Molti social network hanno adottato delle politiche più rigorose in merito all'analisi dei dati prodotti dagli utenti, spesso ostacolando il collezionamento [7];

Secondo [19], un ulteriore problema riguardante i dataset è la piccola quantità di argomenti (veri o falsi) compresi nella collezione il che rende i dataset stessi *biased* e i modelli allenati con questi dataset potrebbero non

generalizzare bene. Per questo motivo, gli autori presentano alcune regole da seguire per creare un dataset *unbiased*, ossia un dataset che contiene un numero bilanciato di notizie vere e false:

- Ogni notizia falsa dev'essere verificata da esperti;
- Le Fake News devono provenire da sorgenti diverse;
- Le notizie vere devono essere pubblicate da organizzazioni giornalistiche credibili;
- Gli articoli ottenuti devono provenire da categorie diverse per rendere la collezione varia.

Come si può vedere in tabella 3.1, esistono alcune collezioni utilizzabili per il task dell'identificazione. Per via dei diversi approcci che la ricerca si pone nello studiare le Fake News, tali dataset variano molto tra di loro; ad esempio, alcuni si concentrano solo su affermazioni politiche mentre altri mantengono il loro campo più libero. Inoltre, non tutte le collezioni forniscono le stesse etichette, si differenziano per il metodo di collezionamento e per le informazioni annotate riguardanti l'articolo (ad esempio, l'autore, la sorgente, i commenti degli utenti, ecc.) [59]. Infine, la maggior parte dei dataset in tabella sono biased e contengono troppe pochi articoli per metodi che intendono sfruttare il Deep Learning.

Un passo verso la risoluzione di questo problema è stato fatto dal **Fake News Corpus**[2]; tale dataset è composto da milioni di articoli, la gran parte ricavati dai 1001 siti provenienti da http://www.opensources.co/ ed alcuni presi da NYTimes e WebHose English News Articles per mantenere bilanciate le classi. Ogni articolo è fornito di informazioni essenziali, tra cui di partcolare interesse è la tipologia di notizia; le tipologie rappresentate sono: Fake News, satire, notizie di parte, teorie di conspirazione, junk science, clickbait, hate news, notizie politiche, inaffidabili ed affidabili. Al momento, il corpus è comprensivo di oltre 9 milioni di articoli, ma presenta delle limitazioni. Prima di tutto, le notizie non sono state filtrate manualmente e, perciò, alcune etichette potrebbero non rivelarsi corrette. Inoltre, l'autore ha espresso che una volta finalizzato il dataset, non verrà mantenuto aggiornato, perciò è probabile che la collezione diventi obsoleta nel giro di breve tempo data la velocità con cui vengono create nuove notizie.

La mancanza di dataset contenenti notizie categorizzate manualmente rappresenta dunque una sfida e, secondo [72], questo problema rappresenta il bottleneck per l'avanzamento nelle tecniche di identificazione automatica.

[2]https://github.com/several27/FakeNewsCorpus

Tabella 3.1: Alcuni dei dataset disponibili pubblicamente [59].

Dataset	**Applicazione**	**Etichette**	**Tipo di Contenuti**
LIAR [72]	Identificazione Fake News	pants-fire, false, barely true, half-true, mostly true, true	affermazioni politiche
FakevsSatire [17]	Identificazione Fake News	fake news, satire	notizie politiche
NewsFN	Identificazione Fake News	fake, real	articoli di giornale
BuzzfeedPolitical [23]	Identificazione Fake News	fake, real	notizie politiche
Political-1 [23]	Identificazione Fake News	fake, real, satire	notizie politiche
NewsFN-2014	Identificazione Fake News	true, mostly true, half true, mostly false, false	affermazioni verificate
NewsTrustData [39]	Valutazione credibilità	punteggio qualitativo	articoli di giornale
Twitter-Weibo [34]	Classificazione dicerie	rumor, non rumor	affermazioni verificate
Twitter15 [35]	Classificazione dicerie	rumor (false, true, remains unverified), non rumor	affermazioni verificare
Twitter16 [35]	Classificazione dicerie	rumor (false, true, remains unverified), non rumor	affermazioni verificate
FacebookHoax [66]	Identificazione bufale	hoax, non hoax	cospirazioni, articoli scientifici
PHEME-R [85]	Analisi dicerie	rumor only (flase, true,remains unverified)	storie degne di nota
PHEME [30]	Classificazione dicerie	rumor (false, true, remains unverified), non rumor	storie degne di nota
BuzzfeedNews [23]	Identificazione Fake News	mostly true, mixture, mostly false, no factual content	notizie politiche
KaggleEmergent	Classificazione dicerie	rumor (false, true, remains unverified)	affermazioni verificate
KaggleFN	Identificazione Fake News	only fake	articoli di giornale
Cred-1 [46]	Estrazione fatti	false, true	affermazioni verificate
Cred-2 [46]	Estrazione fatti	only false	bufale wikipedia
FEVER [69]	Estrazione fatti	supported, refuted, not enough info	affermazioni fabbricate
FNC-1	Identificazione posizione	agrees, disagrees, discussed, unrelated	articoli di giornale
FakeNewsCorpus	Classificazione Fake News	fake, satire, bias, conspiracy, junksci, hate, clickbait, unreliable, reliable, political	articoli reperiti sul web

3.2 Organizzazioni

Come citato brevemente nell'introduzione, esistono delle organizzazioni il cui principale obiettivo è verificare le notizie pubblicate online e identificarne un valore di verità, ossia svolgono il ruolo di fact-checker. Le più importanti e conosciute organizzazioni sono le seguenti:

- **PolitiFact**: Fondato nel 2007 come progetto non-profit, riportano con l'aiuto di giornalisti e media associati l'accuratezza delle dichiarazioni provenienti da figure politiche. Le loro valutazioni sono di diversi tipi: Vero, Quasi Vero, Mezzo Vero, Quasi Falso, Falso e *Pants on Fire*, letteralmente "Pantaloni Infuocati", a significare che la dichiarazione è talmente falsa da essere ridicola. Essendo consapevoli che verificare ogni affermazione politica è un compito impossibile, si affidano ai loro giornalisti ed ai loro lettori per determinare quali fatti siano più meritevoli di essere controllati;

- **Snopes**: Fondato nel 1994, si occupa di investigare leggende metropolitane, bufale, Fake News e folklore riguardanti gli Stati Uniti. Si tratta del più vecchio e più grande sito di fact-checking online e viene riconosciuto da lettori e giornalisti come uno strumento valido per verificare le notizie. I loro valori di verità sono sei: Vero, Quasi Vero, Misto, Quasi Falso e Falso Il loro metodo di selezione di quali articoli verificare è basato unicamente sulle richieste degli utenti: più un fatto è richiesto, più probabilmente sarà verificato.

In figura 3.1 e 3.2 si trovano degli esempi di come i fatti verificati vengono visualizzati sui siti delle organizzazioni descritte. Per quanto riguarda PolitiFact viene riportato prima di tutto il nome della persona a cui l'affermazione è attribuita e il valore di veracità è riportato su una scala, fornita di un ulteriore nota atta a spiegare brevemente il valore assegnato. Cliccando sull'affermazione è possibile poi ottenere maggiori informazioni.

Snopes, invece, riporta semplicemente la notizia e per vederne il valore di verità è necessario entrare nella pagina. Anche in questo caso, la pagina della notizia presenta una descrizione più approfondita sulla notizia stessa e i motivi per cui è stato assegnato un determinato valore di verità.

Tali organizzazioni sono aumentate molto negli ultimi anni e, nonostante si siano presentate qui solo alcune di quelle americane per via del loro grande utilizzo da parte dei ricercatori, ne esistono in tutto il mondo. Sfortunatamente, la mole di notizie che ogni giorno viene pubblicata in rete è enorme, così tanto da rendere impossible la verifica manuale da parte di esperti. Per questo, la ricerca si è concentrata nel ricercare metodi automatici per rilevare le notizie false e si sono instituite anche delle challenge allo scopo di incentivare la ricerca, come vedremo nella prossima sezione.

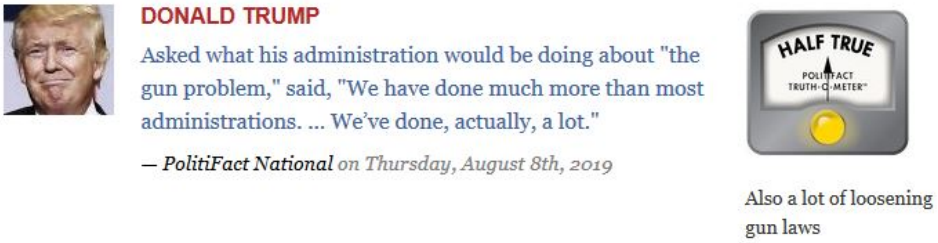


Figura 3.1: Esempio tratto da PolitiFact.

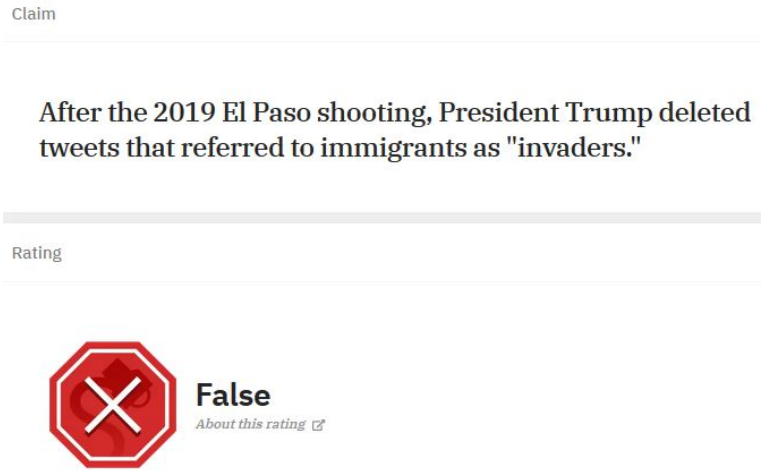


Figura 3.2: Esempio tratto da Snopes.

3.3 Challenge

Così come in altri ambiti della ricerca, anche nel campo delle Fake News sono nate negli ultimi anni alcune challenge atte ad inspirare e stimolare i ricercatori a trovare delle soluzioni, offrendo un premio ai migliori team.

Una delle più famose challenge è la Fake News Challenge (FNC)[3], organizzata alla fine del 2016 e conclusa a Giugno del 2017. Come riportato dal sito ufficiale, l'obiettivo della challenge era esplorare come le tecnologie di intelligenza artificiale possano essere sfruttate per combattere il problema delle Fake News. Secondo i creatori della FNC, il processo di identificazione può essere diviso in fasi e la prima è rappresentata dalla *Stance Detection*. La Stance Detection si occupa di stimare la stance (o la prospettiva) che il titolo di un certo articolo ha con il suo corpo; vengono identificate quattro diverse etichette: agree, disagree, discuss o unrelated, assegnate in base al rapporto che il corpo dell'articolo ha con il titolo. Gli autori della challenge

[3]http://www.fakenewschallenge.org/

proponevano una baseline basata su GradientBoosting e offrivano un premio in denaro per i migliori tre team tra i partecipanti.

Una seconda challenge è denominata CheckThat! Lab ed è stata riproposta più volte nel corso degli ultimi anni. Secondo gli autori, il processo di fact-checking comprende tre step, come mostrato dalla figura 3.3, e la challenge comprende due di questi step, dando per scontato la fase di normalizzazione intermedia. La descrizione dei task viene fornita in [40] come segue:

- **Task 1**: lo scopo di questa fase era individuare quali affermazioni sono utili da verificare e si chiedeva ai partecipanti di creare un sistema simile a factcheck.org per determinare delle strategie di selezione. Dunque, data una trascrizione di un dibattito politico, era richiesto individuare quali frasi priorizzare. Si trattava di un task di ranking, in cui l'output doveva essere una lista ordinata delle frasi da verificare.
- **Task 2**: prendendo in input le affermazioni individuate dal task 1 normalizzate, ossia editate per non essere ambigue o con riferimenti non risolti, era necessario verificarne il grado di veracità. Si trattava, dunque, di un task di classificazione.

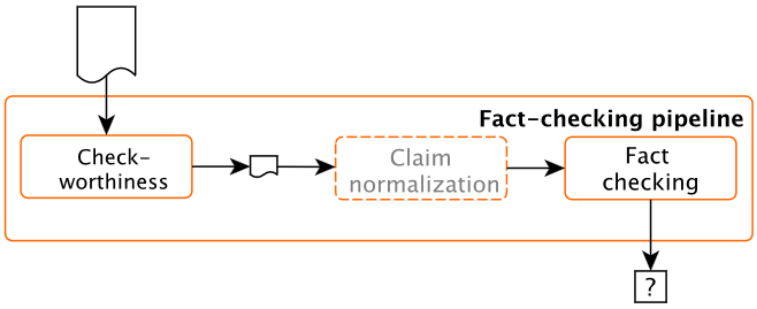


Figura 3.3: Pipeline generale del processo di fact-checking secondo la challenge CheckThat! Lab [40]. Innanzitutto, il documento di input viene analizzato per identificare frasi che meritano di essere verificate; in seguito, le affermazioni individuate vengono estratte, normalizzate ed, infine, verificate.

La presenza di queste challenge nel campo delle Fake News dimostrano ancora una volta l'interesse della ricerca nell'identificazione automatica della veracità e alcuni dei metodi che verranno riportati nei capitoli successivi provengono da team che ne hanno preso parte. Prima di passare alla descrizione delle tecniche sviluppate per l'identificazione negli ultimi anni, verranno ora presentate alcune basi di Reperimento dell'Informazione, Natural Language Processing, Machine Learning e Deep Learning.

Capitolo 4

Metodi basati sul Contenuto

Questo capitolo si occuperà di presentare alcune delle tecniche presenti in letteratura riguardo all'identificazione di Fake News basandosi sul contenuto della notizia stessa. Per contenuto si intende lo studio del testo del documento in modo da ricavare differenze tra articoli veri e falsi e, dunque, generare delle metriche per quantificare tali differenze. I metodi sviluppati in questo campo si avvalgono di competenze proveniente da diversi settori, a partire dal Natural Language Processing (NLP) fino al Deep Learning, ed utilizzano feature estratte manualmente o automaticamente dal testo per determinare la veracità delle notizie.

4.1 Stance Detection

La **Stance Detention** è il primo step individuato dalla Fake News Challenge (FNC) per identificare la veracità di una data notizia, in cui si cerca di capire cosa le varie organizzazioni propaganti articoli dicono su un particolare argomento. Il sito ufficiale della FNC[4] provvede la seguente definizione del problema:

Definizione 4.1. Per Stance Detection si intende il problema che:

- riceve in *input* un titolo e un corpo di un articolo, provenienti dalla stessa notizia o da articoli diversi;
- produce in *output* una classificazione del corpo rispetto al titolo in una delle seguenti categorie:
    1. **Agrees**: il testo è d'accordo con il titolo;
    2. **Disagrees**: il testo non è d'accordo con il titolo;
    3. **Discusses**: il testo parla dello stesso argomento del titolo, ma non prende una posizione;

[4] http://www.fakenewschallenge.org

4. **Unrelated**: il testo riguarda un argomento diverso al titolo.

Il metodo baseline proposto dalla FNC stessa utilizzava delle feature estratte manualmente e un classificatore GradientBoosting, con un'accuratezza del 79,20%.

Secondo [82], un metodo basato sulla classificazione genera delle forti assunzioni sul fatto che ci sia una chiara distinzione sulle posizioni possibili e, dunque, propongono una tecnica ranking-based in modo da identificare la stance relativamente e non in maniera assoluta. Dato il testo e il titolo, utilizzano un multilayer perceptron a due layer nascosti per produrre un valore per ogni possibile posizione. Sia il testo che il titolo sono rappresentati con feature TF-IDF e viene considerata anche la cosine similarity tra testo e titolo, creando dunque il vettore di input al multilayer perceptron $\mathbf{v}$ concatenando le feature. Lo scopo della loss function è di massimizzare la differenza tra i valori associati alle varie stance disponibili.

In seguito, per poter svolgere la classificazione, si assegna l'etichetta corrispondente alla posizione che ha ricevuto un valore maggiore. Utilizzando il dropout e la regolarizzazione L2, [82] ottiene un'accuratezza di classificazione del 86,66%, dimostrandosi migliore della baseline.

Un altro metodo è presentato in [8] in cui la stance detenction viene divisa in due step: il primo per determinare se il titolo è relativo o meno al corpo e il secondo per assegnare la posizione.
Per la prima parte, l'input è lemmatizzato con il CoreNLP Lemmatiser e si ricercano gli n-grammi corrispondenti tra titolo e corpo. Il totale di n-grammi corrispondenti viene moltiplicato per la lunghezza e l'IDF dell'n-grammo corrispondente e poi diviso per il numero totale di n-grammi. Sperimentalmente, si è determinata una soglia di 0,096 e se il totale supera tale valore, la coppia titolo-corpo sarà considerata rilevante. Una definizione formale è la seguente:

$$sc = \frac{\sum_{i=1}^{len(\mathbf{h})} TF^{h_i} \cdot IDF^{h_i}}{len(\mathbf{h}) + len(\mathbf{a})} \tag{4.1}$$

dove $\mathbf{h}$ e $\mathbf{a}$ sono vettori di tutti i possibili n-grammi lemmatizzati del titolo e dell'articolo, $n \in [1, 6]$, h_i e a_i sono l'ennesimo elemento dei vettori $\mathbf{h}$ e $\mathbf{a}$, $len(\cdot)$ una funzione che calcola la lunghezza di una stringa, $TF^{h_i} = [(TF^{h_i}_{\mathbf{h}} + TF^{h_i}_{\mathbf{a}}) \cdot len(h_i)]$ in cui $TF^{x_i}_{\mathbf{x}}$ è la frequenza del termine x_i nel vettore $\mathbf{x}$. IDF^{h_i} è naturalmente l'inverse document frequency del temine h_i.

Per classificare le varie stance, [8] utilizza una classificazione basata su Regressione Logistica allenata solo con i titoli e senza lemmatizzazione o rimozione delle stop world. La scelta della migliore classe avviene in base alla distanza tra quella con miglior punteggio e la seconda migliore: se tale distanza è $\geq 0,7$, la classe migliore sarà scelta, altrimenti tre ulteriori classificatori, allenati con titoli e corpi, vengono usati per assegnare la classe.

Tabella 4.1

Tecnica	**Accuratezza**
Baseline	75,20%
Ranking-based [82]	86,66%
Classificatori combinati [8]	**89,59%**

L'accuratezza totale di questa tecnica risulta essere maggiore del precedente e della baseline con 89,59%.

In tabella 4.1 si riassumono le accuratezze della baseline e dei metodi introdotti per quanto riguarda la Stance Detection, evidenziandone il migliore.

Naturalmente, determinare la posizione che il corpo dell'articolo ha rispetto al titolo è un utile indicatore per determinare se la notizia è falsa e meno, specialmente se ci si trova ad analizzare una notizia clickbait, ma non è una metrica sufficiente.

4.2 Metodi basati su particolari parti del testo

In questa sezione si parlerà di alcune tecniche che concentrano i loro studi su alcune parti della notizia per assegnare il valore di verità.

Il metodo presentato in [70] utilizza la Grounded Theory di Glaser e Strauss per identificare manualmente dei pattern presenti nei documenti di Fake News, in modo da poterli poi raggruppare in categorie e creare la base per una nuova teoria. Ad esempio, se tutte le notizie false iniziassero con la frase "Fidati, non sto mentendo", eventualmente si raggrupperebbero abbastanza dati per formare l'ipotesi che tutte le Fake News iniziano in quel modo. Il lavoro si concentra sull'utilizzare un classificatore che usa l'attribuzione delle citazioni come indicatore. Il processo inizia analizzando i documenti della collezione, contando e dividendo in token i paragrafi. Inoltre, viene controllato se nel paragrafo è presente una citazione tramite un estrattore appositamente realizzato per determinare se la citazione è attribuita o meno; detto C il contenuto in analisi, si definisce una distanza assoluta d entro la quale l'attribuzione avviene nel testo compreso tra virgolette. Per ogni citazione attribuita correttamente, vale che:

$$\exists(Source, Cue)\text{per } C \text{ tale che} \begin{cases} (Source, Cue) \leq x^i + len(C) + 2d \\ (Source, Cue) \geq x^i \end{cases} \tag{4.2}$$

dove per *Source* si intende la porzione di testo che include l'attribuzione della citazione, *Cue* è il verbo o frase che collega la *Source* alla citazione e x^i

è l'inizio della possibile attribuzione prima della citazione (vedi figura 4.1). Se la citazione è attribuita, si ottiene un +1 sul risultato finale, altrimenti -1. Una volta sommati gli individuali valori, se la somma è maggiore di 0, il documento sarà etichettato come vero, altrimenti sarà falso.

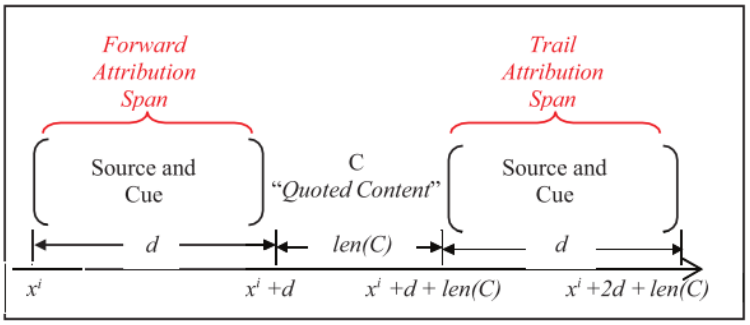


Figura 4.1: Dimostrazione grafica della possibile distanza della coppia $(Source, Cue)$ dal contenuto della citazione C [70].

Gli autori di [70] ammettono che, nonostante il sistema sia in grado di catturare con accuratezza del 96% le citazioni in un testo, sbagliando solo per citazioni complesse o malformate, non ha performance accettabili quando utilizzato per l'identificazione. Infatti, il sistema dimostra di avere problemi a gestire citazioni multiple vicine tra loro e l'accuratezza del classificatore è del solo 69,4%. Analizzando le etichette assegnate scorrettamente, si è trovato che gli errori avvenivano nel caso in cui il documento non contenesse citazioni o se le citazioni erano attribuite ad altri documenti falsi.

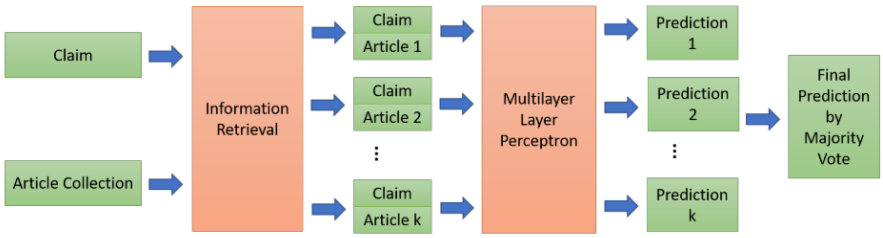
Basarsi, dunque, su delle citazioni può essere utile, ma non porta a dei risultati soddisfacenti. Woloszyn e Wolfgang in [75] propongono di concentrarsi solo sul titolo dell'articolo, estratto dall'URL della notizia. La scelta è giustificata in quanto permetterebbe alla loro tecnica, detta **DistrustRank**, di essere integrata nei browser senza feature addizionali; inoltre, si è osservato empiricamente che le pagine di Fake News sono simili tra loro. Il problema dell'identificazione delle Fake News, dunque, si riduce semplicemente a trovare siti internet i cui titoli non differiscono molto da quei siti che condividono notizie false. DistrustRank consiste nel costruire un grafo in cui i vertici sono i siti web e gli archi definiscono la similarità tra i due siti coinvolti tramite Cosine Similarity calcolata sulla rappresentazione TF-IDF dei siti stessi. Se questa similarità è maggiore ad una certa soglia, i siti saranno considerati simili e un arco sarà inserito nel grafo. Infine viene calcolata la centralità usando PageRank; il ranking ottenuto viene poi trasformato in una classificazione considerando i primi k elementi come positivi e il resto negativi. In tal modo, si ha una precisione dell'80%.

DistrustRank, però, considera i siti come falsi e non si basa sul contenuto della notizia; a volte, però, anche i siti ufficiali potrebbero condividere

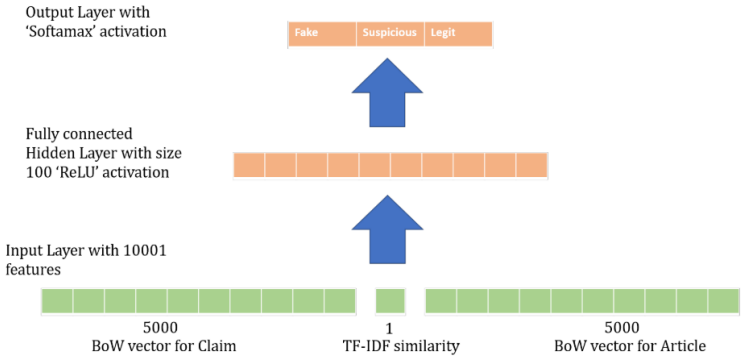
notizie non veritiere ed è dunque importante concentrarsi sul testo e sulle affermazioni riportate.

4.3 Classificazione Binaria

Souvick e Chirag in [16] propongono l'utilizzo di due moduli, uno per verificare la veracità basandosi su un knowledge base esistente con tecniche di Reperimento dell'Informazione, e un altro basato su tecniche di Deep Learning come Long Short Term Memory (LSTM) per catturare le caratteristiche stilistiche con cui la Fake News si distiguono dalle notizie reali. L'architettura utilizzata è riportata in figura 4.2 e, combinando i due moduli, si ha un'accuratezza dell'82,4%.



(a) Primo modulo per la veracità.



(b) Secondo modulo per lo stile di scrittura.

Figura 4.2: Rappresentazione dei componenti di [16].

[19] si propone di studiare quali siano le migliori caratteristiche linguistiche da estrarre dal contenuto delle notizie ed eseguono dei test sulle migliori feature per determinare il miglior algoritmo di Machine Learning per fornire una classificazione. Il processo seguito è indicato in figura 4.3. La valutazione è stata svolta secondo il *test statistico di Friedman* (FST) e viene utilizzato per determinare le migliori feature linguistiche, il miglior tipo di combinazione tra feature e possibili embeddings (ad esempio, Word2vec), e

il miglior algoritmo per la classificazione. Alcune delle migliori feature comprendono: il numero di sillabe, il numero di parole e frasi, il numero di frasi corte e lunghe, la complessità delle frasi, la percentuale di negazioni e articoli, la lunghezza media delle parole e altro. Per quanto riguarda gli algoritmi di Machine Learning testati si riportano in tabella 4.2 la media dei risultati ottenuti per ogni dataset testato; si nota che SVM ottiene un'accuratezza del 95%, dimostrando che è possibile ottenere risultati migliori utilizzando un insieme di feature ottimali.

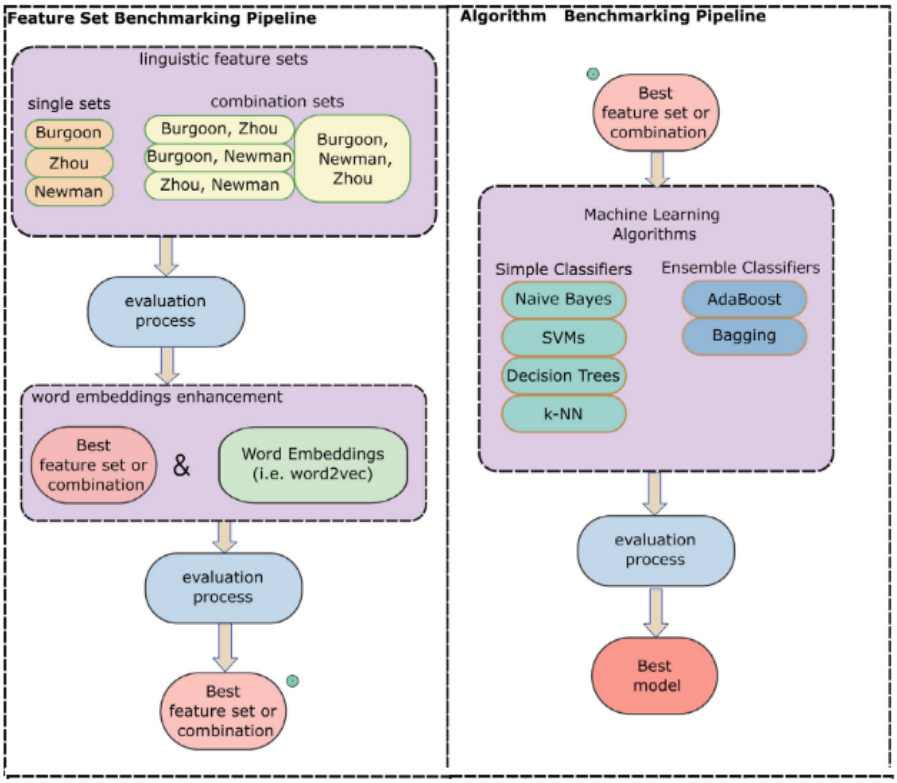


Figura 4.3: Procedimento dello studio in [19].

L'approccio presentato in [2] concentra i suoi sforzi sui post pubblicati e condivisi su Twitter e comprende due componenti; la prima si occupa di identificare automaticamente tramite tecniche di Deep Learning, quali LSTM e Reti Neurali Convoluzionali (CNN), delle feature sui post, mentre la seconda ha il compito di determinare e classificare quali siano le Fake News. Oltre ad utilizzare la fonte testuale del post, vengono prese in considerazione anche le immagini collegate quando disponibili. Svolgendo degli esperimenti, gli autori si sono resi conto che il semplice modelo LSTM ha una accuratezza del 82,29%, migliore del modello ibrido proposto con accuratezza del 80,38% nel dataset PHEME. Nonostante entrami i risultati siano migliori dello stato

Tabella 4.2: Confronto algoritmi di [19] con l'utilizzo di insieme di feature ottimali.

Algoritmo	**Accuratezza**
k-NN	0,921
Decision Tree	0,858
Naive Bayes	0,881
SVM	0,950
AdaBoost	0,949
Bagging	0,944

dell'arte per il dataset utilizzato, Ajao, Bohowmik e Zargari fanno notare che la combinazione LSTM-CNN ha un risultato minore per via del dataset stesso in quanto contiene troppo pochi elementi per allenare efficientemente la rete neurale.

4.3.1 Ricerche Web

In [26], gli autori si affidano ad una ricerca web per capire se l'affermazione è vera o meno; in base a tecniche di IR, partendo dall'affermazione da verificare, si ricava una query composta da nomi, verbi e aggettivi. Dai migliori risultati della ricerca, vengono ricavati gli snippet e le pagine web e si calcolano le seguenti similarità tra affermazione-snippet o affermazione-sito web: Cosine Similarity con TF-IDF, cosine similarity con embedding generati da GloVe e containment, un metodo per calcolare l'intersezione tra due insiemi. La classificazione viene generata da una combinazione di una Rete Neurale (NN) composta da cinque sottoreti LSTM e una Support Vector Machine (SVM) con kernel RBF. Grazie ad un esperimento svolto su delle affermazioni provenienti da Snopes, si è determinata un'accuratezza del 80%; inoltre, gli autori hanno verificato se il risultato è influenzato dal motore di ricerca utilizzato, svolgendo test con Google e Bing, non trovando molta differenza tra i due e ottenendo risultati lievemente superiori usando entrambi i motori.

Un altro approccio che sfrutta ricerche sul Web relative all'affermazione da verificare è presentato in [47] con **DeClarE** (fig.4.4). In questo caso, si considera il contesto dell'affermazione grazie a degli embedding e gli articoli reperiti nel web sono catturati con una LSTM bidirezionale. Vi è inoltre un meccanismo di attenzione che si concentra su parti degli articoli reperiti rilevanti in base all'affermazione in questione. L'importanza di ogni termine di un articolo è calcolata in base alla rappresentazione totale dell'affermazione. Una volta ottenute la rappresentazione dell'articolo e la sua rilevanza rispet-

to all'affermazione, vengono unite tramite una media pesata sull'attenzione precedentemente calcolata. Gli ultimi layer della rete neurale combinano i risultati precedenti prima di ricorrere a un classificatore per generare la credibilità. Gli esperimenti dimostrano che si ottiene un'accuratezza del 78,32% sul dataset di Snopes e del 69,62% su PolitiFact. La differenza rispetto ad altri metodi di confronto che utilizzano LSTM e ottengono risultati lievemente migliori su Snopes è che DeClarE non necessita di feature generate manualmente e può generalizzare bene in altri domini.

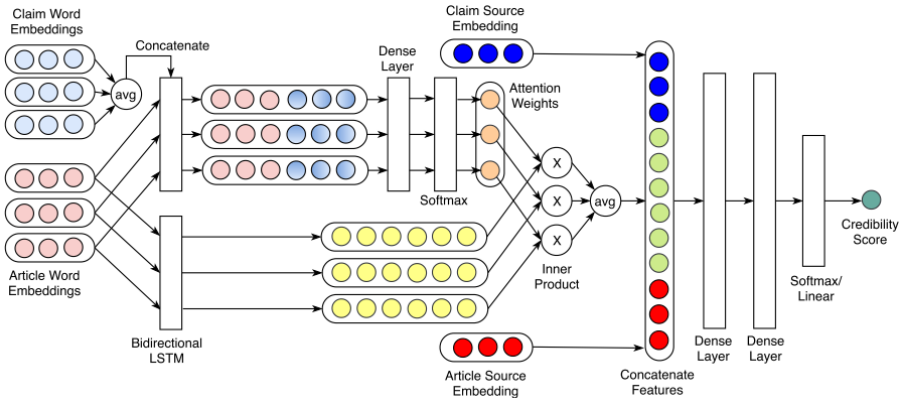


Figura 4.4: Architettura di [47].

4.3.2 Metodi per l'*Early Detection*

Un altro aspetto importante nello studio dell'identificazione delle Fake News è l'*early detection*, ossia l'individuazione delle notizie false quanto prima possibile per evitare tutte le conseguenze negative che queste portano e per mitigare il loro effetto sul pubblico. In [83], gli autori utilizzano un modello basato sull'*Undeutsch hypothesis*, la quale sostiene che le notizie false differiscono da quelle vere per stile e qualità, e si propone risolvere il problema dell'early detection utilizzando solamente il contenuto stesso della notizia. Per catturare lo stile di scrittura, la notizia viene studiata su quattro livelli:

- **Lessico**: a questo livello, si considera la frequenza delle parole tramine il modello *Bag-of-Word* (BOW) standardizzato in modo da catturare le frequenze relative invece di quelle assolute così da tener conto della lunghezza del testo;

- **Sintassi**: per investigare la sintassi, viene usata sia la fequenza di determinati tag (quali nomi, verbi, ecc.) e la frequenza delle produzioni, come le *rewrite rules*. Queste ultime possono essere ottenute

sulla base degli alberi sintattici dellla Grammatica libera da contesto probabilistica;

- **Semantica**: in questa fase, si studiano alcuni attributi della psicologia del linguaggio, ad esempio i sentimenti espressi nel testo. Per farlo, si utilizzano dizionari contenenti espressioni comuni per comprendere se il titolo è clickbait o no, si studia la leggibilità tramite varie metrice sviluppate in educazione, si cerca di determinare se l'articolo è di qualità o meno in base a quante parole significative sono presenti e si valuta l'informalità del contenuto;
- **Discorsivo**: a livello discorsivo, si investigano le relazioni retoriche tra frasi tramite il parser RST.

Le feature identificate ai vari livelli sono poi utilizzate come input in vari framework provenienti dal Machine Learning e testate su fatti collezionati da PolitiFact e BuzzFeed, risultando in un'accuratezza del 89,2 % su PolitiFact e del 87,9% su BuzzFeed. Lo studio conferma, inoltre, che le Fake News sono diverse a livello di scrittura, dimostrando che spesso i titoli hanno alto sensazionalismo, le parole usate sono generalmente corte ed, in proporzione, il titolo contiene più caratteri del testo stesso.

4.4 Classificazione Multipla

Come precedentemente detto, siti quali Snopes e PolitiFact forniscono un valore di verità alle notizie su una scala, comprendendo dei valori intermedi, e gran parte dei fatti verificati risultano essere quasi falsi o quasi veri. Alla luce di questo fatto, [52] si propone di investigare l'identificazione della verità delle notizie su una scala di 6 valori: Vero, Quasi Vero, Mezzo Vero, Quasi Falso, Falso, Pants-on-Fire, valori ricavati da PolitiFact. Dopo aver collezionato ventimila articoli sia da siti ufficiali che da siti noti per pubblicazioni false, le sequenze di parole con embedding GloVe sono utilizzate come input di un modello LSTM. La tabella 4.3 mostra i risultati ottenuti calcolando la macro averaged F1-score e con l'utilizzo di 2 (Vero, Falso) o 6 etichette.

Tabella 4.3

Modello	**2 Classi**	**6 Classi**
Baseline	0.39	0.06
LSTM	0.56	0.20

Anche [27] si concentra nella classificazione multipla, integrando l'utilizzo di varie sorgenti in un sistema denominato **MMFD** (Multi-source Multi-

class Fake News Detection). Il problema viene definito dagli autori come segue:

Definizione 4.2. Dato un dataset $\mathcal{X}$ multi-sorgente contenente n notizie e le corrispondenti etichette Y multi-classe con m livelli di verità, si vuole imparare un modello $\mathcal{M}$ che mappi $\mathcal{X}$ in Y in modo da predirre automaticamente il grado di verità di notizie non ancora verificate.

L'identificazione vista in questi termini genera tre sfide:

1. Le feature devono essere estratte automaticamente in quanto quelle generate a mano non sono molto efficienti;
2. Le diverse sorgenti devono essere combinate in maniera che la soluzione finale sia interpretabile;
3. I gradi di verità, utili per offrire una migliore comprensione, spesso non sono chiaramente separati.

Come altri lavori precedentemente presentati, [27] ricorre a CNN e LSTM, due tecniche di Deep Leaning, per estrarre feature dalle varie sorgenti testuali. La rete neurale convoluzionale si occupa di estrarre pattern locali da un testo, mentre Long Short Term Memory punta a catturare le dipendenze temporali presenti.

Per quanto riguarda la seconda sfida, un metodo ingenuo di combinare le sorgenti è quello di concatenare i vettori di feature estratti precedentemente. Questa tecnica, però, non risulta appropriata in quanto non tutte le sorgenti offrono la stessa potenza nell'identificazione ed è dunque necessario dare maggiore importanza alle sorgenti che risultano più significative al fine del task. Per questo motivo, gli autori suggeriscono di utilizzare un meccanismo di attenzione in modo da rendere la combinazione più interpretabile.

Infine, una funzione multi-classe discriminativa (MFD) viene proposta per risolvere il problema relativo ai gradi di verità; lo scopo è quello di raggruppare gli esempi simili insieme e tenere lontani quelli diversi, ossia si vogliono creare dei cluster di notizie simili. Si calcolano quindi i centri delle varie classi nello spazio delle feature e, se alcuni centri sono troppo vicini tra loro, gli articoli appartenenti a questi gruppi risulteranno meno discriminativi. Inoltre, le notizie devono trovarsi vicino al centro della classe di appartenenza. MFD, dunque, include due termini: uno che spinge i centri ai margini per ridurre le possibili sovrapposizioni e uno che attrae gli articoli verso il rispettivo centro.

Negli esperimenti viene usato il dataset LIAR presentato in [72] e MMFD viene comparato con alcuni metodi:

- SVM: baseline basata sull'utilizzo di Support Vector Machine sulle feature estratte dalle sorgenti;

- Random Forest: baseline simile alla precedente, ma con Random Forest usato come classificatore;
- Rete Neurale: baseline che utilizza una rete neurale completamente connessa ad un livello come classificatore;
- LIAR [72]: per testare il dataset LIAR, gli autori di [72] utilizzano una tecnica basata su CNN e LSTM bidirezionale;
- Random: la classe di appartenenza viene scelta casualmente;
- Maggioranza: a tutti gli articoli viene assegnata la classe con più articoli, ossia la classe *Quasi Vero*.

La tabella 4.4 mostra come il metodo proposto in [27] ha migliori performance in termini di accuratezza rispetto alle tecniche di confronto considerate.

Tabella 4.4: Confronto presentato in [27].

Metodo	**Accuratezza**
Random	17,4%
Maggioranza	20,8%
SVM	29,98%
Random Forest	27,01%
Rete Neurale	29,12%
LIAR [72][5]	27,04%
MMFD	38,81%

4.5 Metodi con Visualizzazione

Così come l'identificazione rappresenta una delle principali sfide proposte dalle Fake News, alcuni studiosi vogliono sottolineare l'importanza di fornire ad eventuali utenti una spiegazione sul risultato ottenuto dal sistema, così da permettere agli utenti stessi di leggere in maniera più critica articoli successivi. Per questo, [77] implementa un sistema spiegabile chiamato **XFake**, composto di tre diversi framework e di una fase di visualizzazione finale con cui l'utente si interfaccia (figura 4.5).

I framework utilizzati sono i seguenti:

[5] L'accuratezza relativa al metodo presentato il [72] non tiene conto dei verdetti compresi nel dataset utilizzato provenienti da PolitiFact.

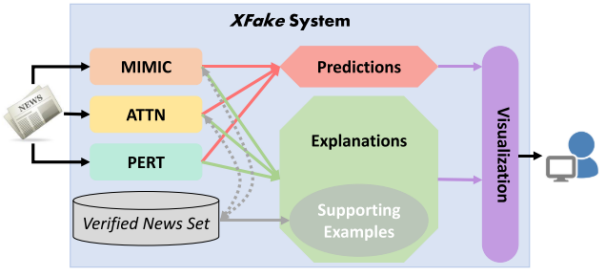


Figura 4.5: Architettura di [77].

- **MIMIC**: si occupa di analizzare gli attributi delle notizie con due rete neurali, denominate insegnante e studente. L'idea è di imitare le reti neurali con metodi di tree ensemble, così da mantenere le performance di una NN e la spiegabilità del tree ensemble. L'input alla rete è l'embedding GloVe ricavato dal contenuto della notizia in tutti i suoi aspetti (comprendendo l'affermazione in esame, chi l'ha detto e per chi, la materia e il contesto);

- **ATTN**: ha il compito di analizzare l'affermazione in questione dal punto di vista semantico con l'utilizzo di embedding Word2vec, CNN e meccanismi di attenzione per catturare le relazioni globali tra parole diverse;

- **PERT**: analizza l'affermazione dal punto di vista linguistico con le seguenti otto feature: percentuale di aggettivi, di sostantivi, di nomi propri, sentiment score, lunghezza del testo normalizzata e se contiene punti di domanda e/o punti esclamativi. In seguito, vi è un classificatore XGBoost genera una predizione.

La predizione finale di XFake indica la probabilità che la notizia sia falsa in base ai risultati combinati dei tre framework. La spiegazione su tale probabilità utilizza alcuni esempi generati da MIMIC e ATTN e l'output è visualizzato tramite D3 Javascript tramite istogrammi e heatmat delle parole o frasi importanti. Inoltre, è possibile interagire con gli ensemble tree generati. Nelle figure 4.6 e 4.7, un esempio di visualizzazione generato dal sistema XFake.

Un approccio diverso è intrapreso in [20]: invece di svolgere esplicitamente il compito di identificare se la notizia è falsa o meno, gli autori vogliono offrire un servizio il cui compito è presentare dagli articoli simili già verificati. Il sistema coinvolge i seguenti sei passi:

1. Creazione corpus di articoli: delle affermazioni sono ricavate da Schema.org ClaimReview e, una volta rimosse le istanze duplicate, si sono ottenuti 5350 fatti verificati;

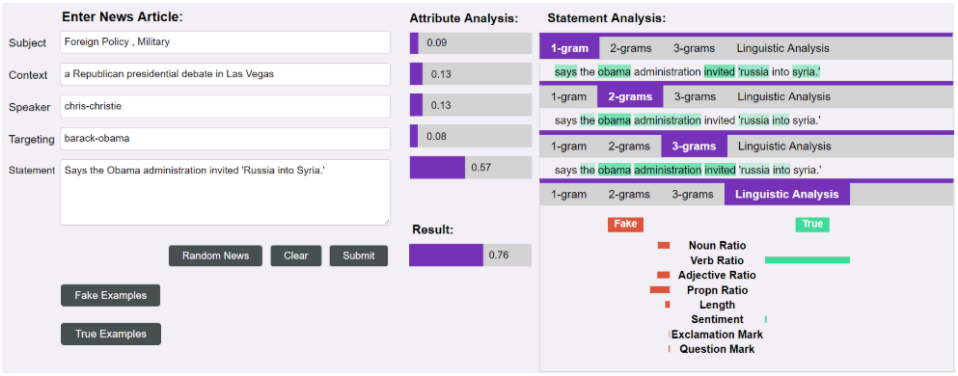


Figura 4.6: Predizione e spiegazione di XFake [77].

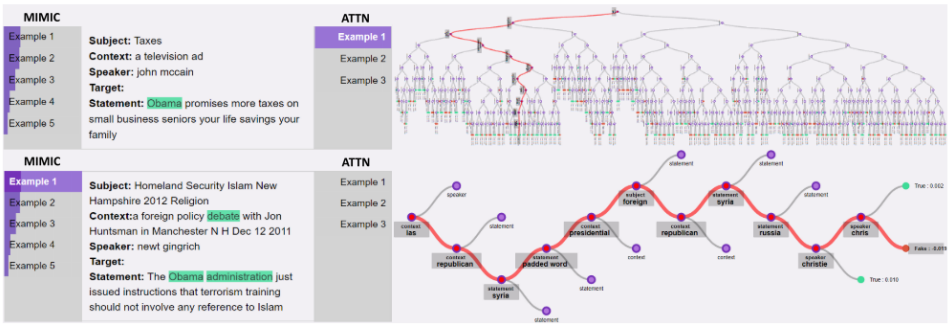


Figura 4.7: Esempi di supporto ed ensemble tree di XFake.

2. Identificazione argomenti comuni nelle Fake News: si nota che le notizie false tendono a concentrarsi su particolari temi, ad esempio la pericolosità dei vaccini e il cambiamento climatico e si sono identificati cinque temi principali. Per ognuno di questi argomenti è poi costruito un classificatore;

3. Ranking dei fatti verificati in base alla rilevanza rispetto un particolare articolo: la rilevanza di un fatto all'articolo è prima di tutto data da quei fatti che sono specificatamente menzionati nell'articolo e da quei fatti che sono sullo stesso argomento dell'articolo. Gli autori affermano che se l'utente può vedere che la notizia che sta leggendo è parte di una lunga serie di storie, allora sarà in una posizione migliore per giudicare la veracità;

4. Mappatura da un particolare articolo ad altri simili e verificati;

5. Valutazione della rilevanza dei risultati: questo passo è svolto tramite tecniche di Reperimento dell'Informazione: modello vettoriale, valo-

ri TF-IDF e similitudine valutata secondo il coseno dell'angolo tra i vettori degli articoli considerati.

Questo servizio è accessibile come estensione browser e, se l'utente vuole usufruirne, è sufficiente cliccare sull'icona "F" dell'estensione. Il risultato è presentato in maniera piuttosto semplice, come si può vedere in fig.4.8

Figura 4.8: Visualizzazione del servizio presentato in [20]. In alto, l'affermazione che l'utente ha deciso di verificare riguardante vaccini e in seguito alcuni articoli relativi e verificati da Snopes. Infine, un articolo simile proveniente dallo stesso sito dell'affermazione in questione.

In tabella 4.5 si trovano tutti i metodi presentati nel capitolo, riportandone le principali caratteristiche e le performance. Per i metodi sprovvisti di nomi, si è utilizzato il nome degli autori o un nome rappresentativo della tecnica.

Nonostante esistano molti metodi in letteratura che utilizzano semplicemente il contenuto della notizia per dedurne la veracità, il task di identificazione rimane una sfida principalmente perché le Fake News sono costruite e si evolvono in modo da sembrare vere agli utenti e per evitare l'identificazione. Per questo, sebbene l'utilizzo del contenuto sia stato studiato approfonditamente, spesso è difficile determinare la veracità basandosi solo sul testo.

Tabella 4.5

Metodo	**Caratteristiche**	**Acc.**
Stance Detection Ranking [82]	Stance Detection con stance relativa, rappresentazione TF-IDF titolo e corpo, cosine similarity, classificazione tramite multilayer perceptron	86,66%
Classificatori Combinati [8]	Stance Detection con stance assoluta, titolo e corpo lemmatizzati, classificazione con quattro classificatori allenati su feature diverse	89,59%
Grounded Truth [70]	Tokenizzazione paragrafi, verifica attribuzione delle citazioni, classificazione binaria basata sul punteggio ottenuto dalle citazioni attribuite e non attribuite presenti	69,4%
DistrustRank [75]	Similitudine tra siti calcolata tramite rappresentazione TF-IDF dei siti stessi, utilizzo solo dell'URL dell'articolo, classificazione binaria determinata in base al ranking fornito da PageRank	80%
Souvick-Chirag [16]	Studio feature stilistiche tramite LSTM bidirezionale, classificazione binaria e ternaria tramite tecniche di IR e perceptron	82,4%
Web Search [26]	Ricerche web con query generati dall'articolo da verificare, cosine similarity sulla rappresentazione dell'articolo e i risultati delle ricerche, classificazione binaria tramine 5 reti LSTM e una SVM	80%
DeClarE [47]	Verifica affermazioni, embedding dell'affermazione per considerarne il contesto, meccanismo di attenzione per concentrarsi su parti rilevanti di articoli reperiti dal web, classficazione binaria	78,2%
Best Combination [19]	Composizione feature di studi precedenti, test di Friedman per determinare migliore combinazione di feature e miglior classificatore per classificazione binaria	95%
Ajao-Bhowmik-Zargari [2]	Feature estratte automaticamente tramite LSTM e CNN, classificazione binaria basata sia sul testo che sulle immagini	80,38%
Undeutsch Hypothesis [83]	Early detection basato su feature lessicali, sintattiche, semantiche e discorsive, classificazione binaria tramite tecniche di Machine Learning	89,2%
Rashkin [52]	Embedding GloVe, utilizzo del dizionario LIWC, classificazione multiclasse tramite LSTM	n/a
MMFD [27]	Estrazione automatica delle feature con CNN e LSTM, multiple sorgenti dell'articolo considerate, classificazione multiclasse basata su una funzione MFD per determinare e raggruppare le notizie nei gradi considerati	38,81%
XFake [77]	Classificazione basata su un punteggio determinato da 3 moduli consideranti contenuto, semantica e caratteristiche linguistiche, visualizzazione tramite varie tecniche con l'obiettivo di spiegare all'utente il risultato sulla veracità	n/a
Related Fact Check [20]	Browser add-on con l'obiettivo di aiutare l'utente a determinare indipendentemente se l'articolo è vero o meno, visualizzazione articoli correlati con relativa valutazione di veracità	n/a

Capitolo 5

Metodi basati sulla Propagazione o sul Feedback

Sui social media sia le notizie vere che le Fake News si propagano tramite condivisioni da parte degli utenti e, dunque, si vengono a creare diffusioni a cascata o ad albero, il cui il post iniziale rappresenta la radice (figura 5.1). Lo studio di questa propagazione può portare all'identificazione delle notizie false in quanto si è verificato che queste ultime si propagano in maniera diversa rispetto alle notizie ufficiali.

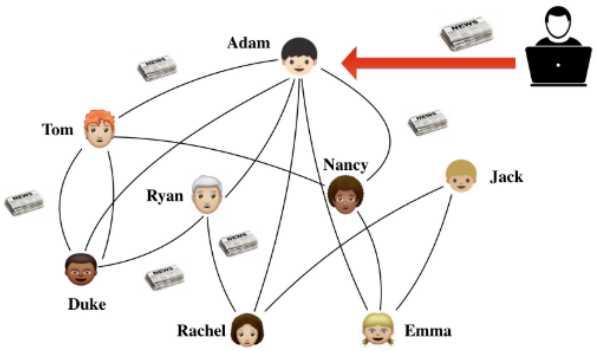


Figura 5.1: Esempio di propagazione sui social network [80].

Un'altra risorsa molto importante per l'identificazione di Fake News è il feedback fornito dagli utenti. Quando un post viene pubblicato su un social media, generalmente gli utenti che lo ricevono, offriranno nei commenti una loro opinione riguardo a quanto letto. Inoltre, Facebook e altri social network hanno recentemente aggiunto un sistema di flagging: se un utente ritiene che la notizia sia falsa, è possibile segnalarla con un flag. Questo genere di interazione con la notizia viene spesso chiamato Feedback ed esi-

stono, in letteratura, degli studi sull'utilizzo di tali indicatori per individuare le Fake News.

5.1 Metodi basati sulla Propagazione

Un possibile rimedio per combattere la propagazione delle Fake News è quello di propagare parallelamente la notizia reale sulla rete [22, 41], ma la tecnica non è efficace in quando è impossibile determinare la notizia vera da diffondere prima che degli individui selezionati I conoscano la Fake News da combattere. Per questo motivo, [63] propone un metodo diverso per determinare gli individui I in modo che la Fake News proveniente dalla sorgente S raggiunga con alta probabilità I e molti altri nodi non ancora influenzati sono, in seguito, raggiungibili da I. La tecnica proposta prende il nome di **FActCheck** e si basa sugli insiemi PRR (Pruned Reverse Reachable). Si individuano gli insiemi di tutti i nodi che collegano almeno una sorgente in S a un vertice scelto casualmente; una volta generati θ molto grande di questi PRR, si utilizza un algoritmo greedy per selezionare il set migliore.

Wu e Liu in [76] propongono **TraceMiner**, il quale utilizza le tracce dei messaggi come input per determinare la categoria. Si definisce un grafo $G = (U, E)$ con $u_i \in U$ nodo rappresentate un utente e E insieme di vertici, e sia M l'insieme di messaggi scambiati. Ogni messaggio m_i ha il suo insieme di condivisori $\{(u_1^{m_i}, t_1^{m_i}), ..., (u_n^{m_i}, t_n^{m_i})\}$ dove $t_k^{m_i}$ è l'istante in cui il messaggio è stato condiviso dall'utente u_k. Inoltre, sia Y l'insieme delle etichette.
Data la sequenza di condivisori del messaggio m_i e il grafo G, è possibile ottenere la topologia, solitamente ad albero, ma tale metodo è difficile da gestire in quanto ci sono n^{n-2} diversi alberi possibili. Per questo, gli autori ricorrono a convertire la sequenza di condivisori in una sequenza temporale. In tal modo, però, si potrebbero perdere le dipendenze tra utenti presenti nel grafo tramite archi e [76] propone di utilizzare una rete neurale ricorrente (RNN) con LSTM per catturare le dipendenze. La loss function del modello è la sequente:

$$\sum_{i=1}^{|X_{tr}|} |Y_{tr} = 0| y_i log(\hat{y}_i) + |Y_{tr} = 1|(1 - y_i) log((1 - \hat{y}_i)) \tag{5.1}$$

dove $\hat{y}_i$ è l'etichetta predetta dal modello, y è l'etichetta reale, X_{tr} sono le sequenze temporali di training e $|Y_{tr} = 0|$ ($|Y_{tr} = 1|$) è il numero di istanze negative (positive) di training. Oltre alla classificazione, per alleviare la sparsità dei dati, gli autori ricorrono a metodi di embedding per catturare le feature degli utenti nei social media.
Il dataset utilizzato negli esperimenti è composto da 3600 tweet collezionati dagli autori e verificati su Snopes per ottenerne la veracità; TraceMiner viene confrontato con due algoritmi, uno basato su SVM e uno su XGBoost, e viene usata F-measure come metrica di valutazione. In tabella 5.1 i risultati

Tabella 5.1: F1-measure dei diversi metodi confrontati in [76].

Dati di Training	10%	30%	50%	70%	90%
SVM	0,5825	0,6122	0,6658	0,7224	0,7581
XGBoost	0,6558	0,7002	0,7288	0,7984	0,8226
TraceMiner	0,7867	0,8344	0,8547	0,8988	0,9124

ottenuti e si nota che le performance sono buone anche con l'utilizzo del 10% dei dati di training, dimostrando la potenziale utilità della tecnica in ambiente di *early detection*.

5.1.1 Metodi per l'*Early Detection*

Essendo, come già notato, la velocità un fattore molto importante quando si tratta di diffusione di notizie false, gli autori di [73] propongono una tecnica denominata **QuickStop**. Un concetto di base introdotto qui è quello che una sola condivisione rappresenta un segnale debole e dunque non sufficiente per una classificazione accurata, ma accumulando segnali deboli è possibile ottenere migliori risultati al prezzo di maggiore diffusione. Vi è un fattore di trade-off tra quantità di segnali deboli accumulati e propagazione della notizia. L'obiettivo è, di conseguenza, determinare una stopping policy, ossia quando fermare il collezionamento di informazioni per determinare la veracità prima che la misinformazione sia troppo diffusa.

Come per [76], la rete viene rappresentata con un grafo $G = (U, E)$ dove U è l'insieme degli utenti ed E l'insieme degli archi rappresentanti connessioni tra utenti. Gli archi sono diretti: se $(u, v) \in E$ significa che v è un *follower* di u. Un utente u decide di ri-condividere l'informazione ricevuta in base a tre fattori: il tipo di informazione, le sue caratteristiche $\mathbf{U}(u)$ e l'insieme di suoi vicini che hanno condiviso l'informazione prima di u. Per modellare le ricondivisioni, si utilizza un modello basato sugli archi: è più probabile che le Fake News si diffondano su un arco tra due utenti maligni rispetto a uno tra due utenti onesti.

Si assume che le osservazioni sequenziali formino una catena di Markov. La stopping condition cercata dipende da due costi:

1. Error cost: ossia la probabilità che l'etichetta assegnata sia sbagliata. Detti c_I il costo di un errore di tipo I (detto anche falso positivo, notizia vera segnalata come falsa) e c_{II} il costo dell'errore di tipo II (detto anche falso negativo, notizia falsa dichiarata vera), il costo dell'errore è:

$$c_e(\delta_T) = c_I Pr(\hat{y}_T = 1|H_0)(1 - \pi_0) + c_{II} Pr(\hat{y} = 0|H_1)\pi_0 \qquad (5.2)$$

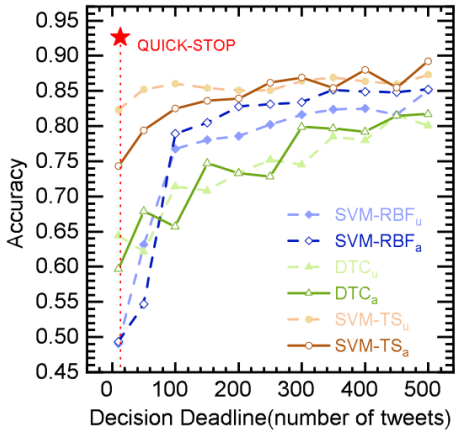


Figura 5.2: Confronto accuratezza QuickStop con altri metodi presenti in letteratura [73].

dove H_0 (H_1) è l'ipotesi che l'informazione sia vera (falsa), $\hat{y}_T$ è l'etichetta predetta trascorso il tempo T (stopping time) e π_0 è la probabilità a priori di H_1.

2. Propagation Cost: dovuto al fatto che più l'informazione si diffonde, più le persone la condivideranno. Individuare la misinformazione il prima possibile è necessario per limitarne i danni e il costo di propagazione al tempo T è definito come:

$$E[cT\mathbb{I}_{H_1}] \tag{5.3}$$

dove c è il costo associato ad ogni slot temporale se l'informazione propagata è falsa e $\mathbb{I}_{H_1}$ è la funzione scalino pari a 1 quando H_1 è vera e zero altrimenti.

Dunque, la funzione obiettivo sarà:

$$\inf_{T,\delta_T} c_e(\delta_T) + E[cT\mathbb{I}_{H_1}] \tag{5.4}$$

Gli esperimenti sono svolti sul dataset Weibo e i risultati sono riportati in figura 5.2. QuickStop risulta avere un'accuratezza del 93% con solo 15 osservazioni in media, migliore di altri algoritmi che richiedono 500 osservazioni per accuratezza di meno del 90%.

Un secondo metodo sviluppato per l'early detection è presentato in [32] e considera i cammini di propagazione come serie temporali multivariate. Siccome si tratta di individuare le Fake News il prima possibile, si considera solo un cammino parziale entro il tempo T. La classificazione è poi basata sulle rappresentazioni del cammino di propagazione ricavate da RNN e

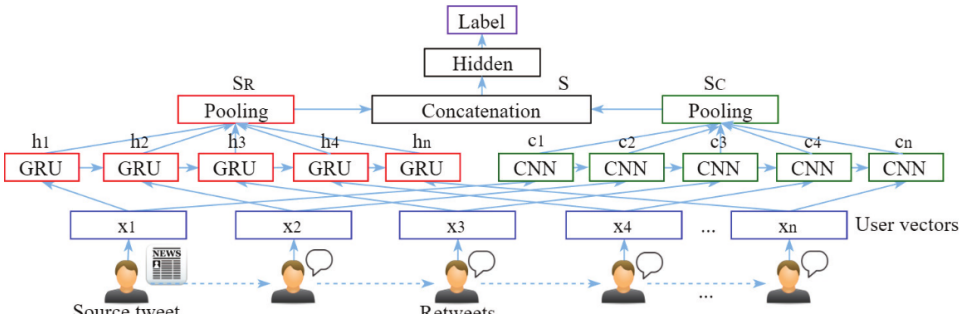
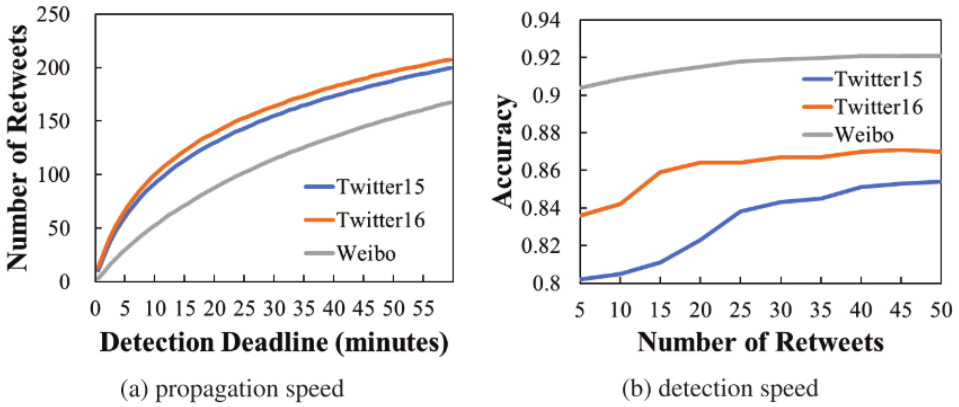


Figura 5.3: Architettura [32].



(a) propagation speed (b) detection speed

Figura 5.4: Identificazione in base alla velocità di propagazione e al numero di ricondivisioni [32].

CNN con un'architettura illustrata in figura 5.3; le rappresentazioni sono dunque concatenate e una rete neurale feedforward multi-livello si occupa dell'assegnamento delle etitchette.

Gli esperimenti vengono svolti sui dataset Weibo, Twitter15 e Twitter16 e l'accuratezza risulta essere rispettivamente del 92% e del 85% dopo solo cinque minuti dall'inizio della diffusione (figura 5.4).

5.2 Metodi basati sul Feedback

Jiang e Wilson in [24] si propongono di studiare i commenti degli utenti dal punto di vista linguistico al fine di predirre la veracità di un determinato post. Per poter sviluppare tale tecnica, prima di tutto gli autori hanno costruito un dataset di 5303 fatti verificati e i loro relativi commenti per poi usare la collezione per creare un nuovo dizionario detto **ComLex**. ComLex è stato generato con combinazioni di embedding di parole e clustering non supervisionato. I cluster finali sono stati etichettati a mano.

Per quanto riguarda l'identificazione, si è usato il dataset collezionato, esaminando solo i commenti pubblicati prima che il fatto venisse verificato. Si sono verificate le seguenti affermazioni grazie all'utilizzo di ComLex:

- Gli utenti tendono a commentare con parole relative a misinformazione su notizie poi determinate come false;
- Gli utenti usano più emoji su notizie in seguito scoperte false;
- La discussione di argomenti concreti e l'oggettività decresce all'aumentare della misinformazione;
- Gli utenti esprimono più opinioni personali su post veri.

A forte delle precedenti scoperte, [24] applica tecniche di Machine Learning (SVM, regressione lineare, NN) fornendo in input i vettori rappresentativi dei commenti degli utenti. Nonostante ComLex dimostri avere performance migliori in termini di Spearman ρ, LRAP (Label Ranking Average Precision) e LRL (Label Ranking Loss) rispetto ad altri dizionari quali LIWC e EmoLex, l'errore è comunque alto, provando che i commenti degli utenti sono un segnale debole per l'identificazione.

5.2.1 Metodi che utilizzano le *flag*

Molti social media hanno aggiunto al loro sistema un sistema di flagging per permettere ai loro utenti di segnalare quei post che, a loro avviso, non riportano notizie veritiere. Grazie a questo sistema, i ricercatori hanno determinato alcuni metodi per individuare quali notizie sono meritevoli di essere verificate.

Una prima tecnica che si prefissa di utilizzare il sistema di flagging è **CURB**, presentata in [29]: se un post ottiene abbastanza flag, viene inviato a degli esperti per la verifica. Il problema, dunque, riguarda determinare quante flag sono necessarie prima di intervenire. Si utilizzano due marked temporal point process, uno per gli eventi esogeni (ossia la pubblicazione spontanea dell'utente di una storia) e uno per gli eventi endogeni corrispondenti alla ricondivisione e/o all'attivazione di una flag. Per determinare quali notizie mandare in verifica agli esperti, viene risolto un problema di ottimizzazione statistica del controllo per SDE con salti. Gli esperimenti svolti sui dataset di Twitter e Weibo dimostrano come CURB sia in grado di limitare la diffusione delle Fake News (figura 5.5), fermando il processo di propagazione prima che la notizia diventi viral.

Anche [71] si propone di selezionare k notizie in base al numero di flag ricevute ed inviarle ad esperti per la verifica. Le differenze dal metodo precedente sono le seguenti:

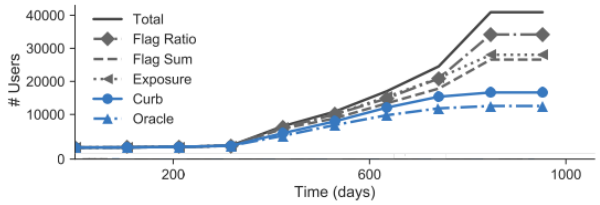


Figura 5.5: Performance di CURB [29].

1. gli utenti non sono considerati uguali nella loro capacità di riconoscere articoli falsi e l'accuratezza dell'attività di flagging è imparata durante il processo;
2. il tempo è discreto, mentre [29] considerava il tempo continuo;
3. la tecnica proposta è indipendente dalla propagazione sulla rete della notizia.

La rete è rappresentata come grafo $G = (U, E)$ e l'esecuzione è divisa in epoche denotate $t = 1, 2, ..., T$, con un'epoca equivalente a un determinato lasso di tempo (ad esempio, un giorno). Un insieme X^t di notizie nuove viene generato all'inizio di ogni epoca e l'etichetta $\hat{y}(x)$ può essere determinata solo dagli esperti. È necessario determinare l'insieme S^t sottoinsieme delle notizie attive A^t da inviare per la verifica riducendo il più possibile il numero di utenti che entra in contatto con la notizia; l'utilità della verifica è data da:

$$Util(T, alg) = \sum_{t=1}^{T} \mathbb{E}\left[\sum_{s \in S^t} \mathbb{I}_{(y^*(s)=f)} val^t(s)\right] \tag{5.5}$$

dove alg è l'algoritmo utilizzato per determinare S^t, $val^t(s)$ è la differenza tra il numero di utenti che eventualmente vedranno la notizia e il numero di utenti che ha visto la notizia fino al tempo t. Gli autori di [71] propongono un algoritmo denominato DETECTIVE basato sul sampling di Thompson: ad ogni invocazione, l'algoritmo campiona i parametri degli utenti dalle distribuzioni posteriori correnti degli utenti ed invoca l'algoritmo TOPX per la selezione degli articoli da inviare agli esperti. Tale algoritmo converge teoricamente per $T \to \infty$, ma si è osservato nella pratica che ha performance competitive rispetto ad altri metodi presenti in letteratura.

Sia affidarsi alla propagazione che al feedback, però, porta a risultati generalmente inferiori a quelli forniti da tecniche basate solo sul contenuto della notizia. Gli autori di [29] fanno inoltre notare che un metodo basato sul flagging diventa naturalmente meno efficacie in base all'accuratezza degli utenti nel segnalare gli articoli. Inoltre, come riporta [50], quando si tratta di *early detection*, non è possibile fare affidamento al feedback fornito dagli

utenti in quanto i commenti non sono ancora disponibili nel momento della pubblicazione della notizia.

La tabella 5.2 mostra brevemente i metodi presentati nel capitolo, riportandone le principali caratteristiche. Per i metodi sprovvisti di nomi, si è utilizzato il nome degli autori o un nome rappresentativo della tecnica.

Tabella 5.2

Metodo	**Caratteristiche**
FActCheck [63]	Utenti della rete selezionati tramite PRR per propagare le notizie vere e combattere le false, rete rappresentata da un grafo con nodi gli utenti e archi le interazioni
TraceMiner [76]	Rete rappresentata come grafo, propagazione descritta da una sequenza di condivisori, classificatore binario basato su LSTM, possibile utilizzo per early detection
QuickStop [73]	Rete rappresentata come grafo, propagazione descritta come osservazioni sequenziali modellati secondo catene di Markov, stopping condition per evitare che la notizia diventi viral, utilizzabile per early detection (performance massime dopo 15 osservazioni)
Liu-Wu [32]	Utilizzo della propagazione con cammini visti come serie temporali multivariate, classificazione binaria tramite CNN e RNN
ComLex [24]	Studio dei commenti degli utenti e creazione di un dizionario di essi, varie tecniche di Machine Learning testate come classificatore binario
CURB [29]	Individua fatti da verificare tramite il sistema di flagging, processi temporali per rappresentare gli eventi di pubblicazione, condivisione e flagging degli utenti, tempo considerato continuo, verifica svolta da esperti
Crowd Signals [71]	Individua fatti da verificare tramite il sistema di flagging, rete rappresentata come grafo, tempo considerato discreto, nella pratica ha performance competitive ad altre tecniche

Capitolo 6

Metodi basati sul Contenuto e sul Feedback

Si è visto finora che basarsi solamente su contenuto, feedback o propagazione non porta a risultati ottimi in fase di identificazione. Per questo motivo, la ricerca nel campo delle Fake News ha cercato una soluzione nella combinazione di feature provenienti dal contenuto della notizia e feature provenienti dai profili degli utenti che le condividono o i loro commenti.

6.1 Metodologie

J. Kim, D. Kim e A. Oh in [28] supportano l'ipotesi cche il maggior fattore che permette la disseminazione di notizie è l'*omogeneità* tra gli utenti e, perciò, propongono **HBTP** (Homogeneity-Based Trasmissive Process), un modello Baesiano non parametrico in grado di catturare l'interazione tra contenuto, argomenti trattati dalle notizie vere e false, e gli interessi degli utenti. L'interesse dell'utente è determinato tramite dei processi Dirichlet a due livelli, mentre il modello non-parametrico è usato per i contenuti; combinando i due modelli con un processo Gaussiano Baesiano, viene determinato un indice di omogeneità che dimostra avere delle relazioni interessanti con la veracità dell'articolo (figura 6.1).
Verificata tale dipendenza, gli autori hanno costruito un modello supervisionato per HBTP per svolgere la classificazione ottenendo un'accuratezza del 78%.

In [60] viene proposto **FakeNewsTracker** i cui obiettivi sono i seguenti:

1. *Collezionare Fake News*: essendo importante la creazione di un dataset, gli autori propongono di collezionare notizie vere e false giornalmente usando siti come PolitiFact come risorsa e la API di Twitter per ricavare le risposte degli utenti alle notizie;

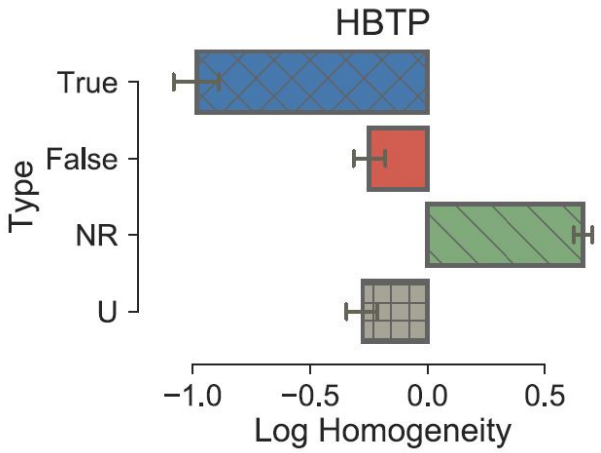


Figura 6.1: Media dell'indice di omogeneità per storie con diverse etichette [28].

Tabella 6.1: Confronto sull'accuratezza del metodo proposto da [60].

	SAF/S	**SAF/A**	**SAF**
PolitiFact	63,3%	62%	67%
BuzzFeed	62,3%	57,1%	74,2%

2. *Identificare le Fake News*: l'identificazione avviene tramite un modello SAF (Social Article Fusion). La rappresentazione degli articoli avviene tramite autoencoder, mentre per l'interazione con gli utenti si catturano le dipendenze temporali con una LSTM. In seguito, SAF unisce la rappresentazione del contenuto con quella del Feedback degli utenti ed utilizza tali feature per svolgere la classificazione binaria;

3. *Visualizzazione*: per individuare le caratteristiche delle Fake News e le differenze tra utenti, gli autori propongono di varie tecniche di visualizzazione, tra cui world cloud che evidenziano le parole più usate per notizie false e vere, e una geo-localizzazione per mostrare in quali aree le Fake News solitamente si propagano.

Gli esperimenti sono svolti sul dataset collezionato e si dividono in SAF/S, il quale considera solo il feedback, SAF/A che si avvale solo del contenuto, e SAF, il modello completo. Si può vedere in tabella 6.1 che la combinazione di contenuto e feedback porta a risultati migliori in termini di accuratezza del modello.

Shu, Wang e Liu in [62] dimostrano che anche gli editori delle notizie giocano un ruolo importante e, perciò, propongono di studiare le relazioni

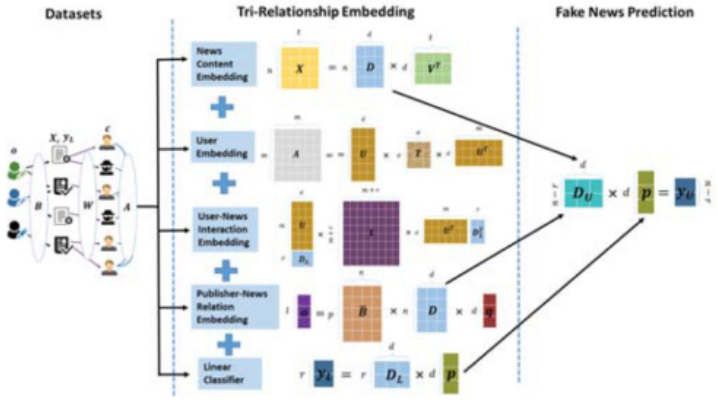
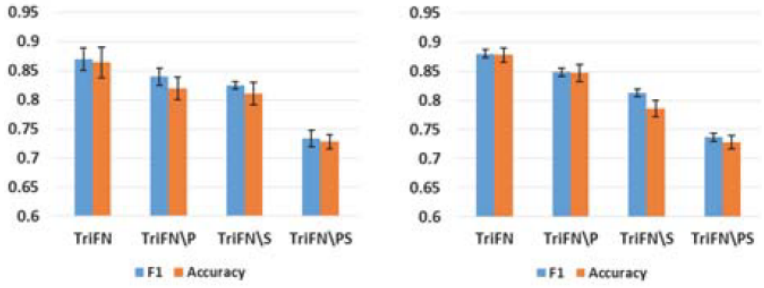


Figura 6.2: Architettura TriFN con cinque componenti: embedding per il contenuto, per gli utenti, per l'interazione tra utenti e articoli, per l'interazione tra editori e articoli, e classificatore lineare per gli articoli [62].



(a) BuzzFeed (b) PolitiFact

Figura 6.3: Impatto dei vari moduli sul risultato; TriFN\P esculde gli editori, TriFN\S esclude gli utenti, TriFN\PS esclude utenti ed editori [62].

tra editori, utenti e articoli in quanto contengono informazioni aggiuntive al fine dell'identificazione delle notizie false. Tale metodo viene denominato **TriFN** e in figura 6.2 viene mostrata l'architettura relativa, costituita di cinque moduli.

Durante gli esperimenti, il dataset FakeNewsNet collezionato su PolitiFact e BuzzFeed viene usato e TriFN ottiene performance maggiori rispetto ad altri metodi presenti in letteratura, con un'accuratezza del 86,4% su BuzzFeed e del 87,8% su PolitiFact, dimostrando anche come ogni modulo influenzi il risultato (figura 6.3). Inoltre, gli autori hanno dimostrato che il loro metodo è in grado di ottenere accuratezza dell'80% dopo sole 48 ore dalla pubblicazione della notizia.

Un ulteriore metodo che si basa su contenuto e feedback è fornito da [12]; al contenuto è applicato il Porter Stemmer ed ogni post è rappresentato da

Tabella 6.2: Confronto tra TriFN [62] e metodo di [12] usando il contenuto e HC.

	TriFN[62]	**HC-CB** $\lambda = 3$ [12]
BuzzFeed	86,4%	85,6%
PolitiFact	87,8%	93,8%

vettori TF-IDF, mentre l'interazione sociale viene considerata solo se supera una certa soglia λ secondo la regola:

$$\begin{cases} interazioni < \lambda\text{: usa solo il classificatore basato sul contenuto} \\ interazioni \geq \lambda\text{: usa anche il classificatore basato sul feedback} \end{cases} \tag{6.1}$$

Tale regola è basata sul fatto che gli algoritmi utilizzati qui e proposti in [67] (Logistic Regression LR e Harmonic boolean label crowdsourcing HC) dimostrano performance migliori solo se le interazioni sono in numero sufficiente, altrimenti i risultati potrebbero peggiorare. Il metodo è confrontato direttamente con TriFN di [62] e, come si può vedere in tabella 6.2, sul dataset di BuzzFeed [12] ha performance minori di TriFN, ma per PolitiFact si ha un miglioramento del 6%.

6.1.1 Metodi per l'*Early Detection*

Il modello **ACAMI** proposto in [78] si avvale della distribuzione temporale degli eventi relativi alle notizie, in quanto è possibile mostrare che le caratteristiche temporali sono diverse tra notizie vere e false, presentando una maggiore quantità di Fake News nella fase intermedia di un evento (figura 6.4). La tecnica si sviluppa su due moduli:

1. *Event2vec*: considerato un evento come un insieme di molti post in cui ogni post comprende un testo e un timestamp, se ne determina una rappresentazione. Diversamente dal metodo **CAMI** presentato dagli stessi autori in [79], questo modulo si occupa di estrarre contenuto e importanza temporale a livello di post;

2. *CNN*: si usa una rete neurale convoluzionale per estrarre automaticamente feature significative e produrre una classificazione.

Grazie a questa struttura, ACAMI ottiene accuratezza del 94,8% sul dataset di Weibo e del 80,3% su Twitter, superando CAMI grazie al meccanismo di attenzione utilizzato. Per quanto riguarda early detection, ACAMI riesce a raggiungere l'accuratezza del 94,8% su Weibo dopo 8 ore dall'inizio dell'evento.

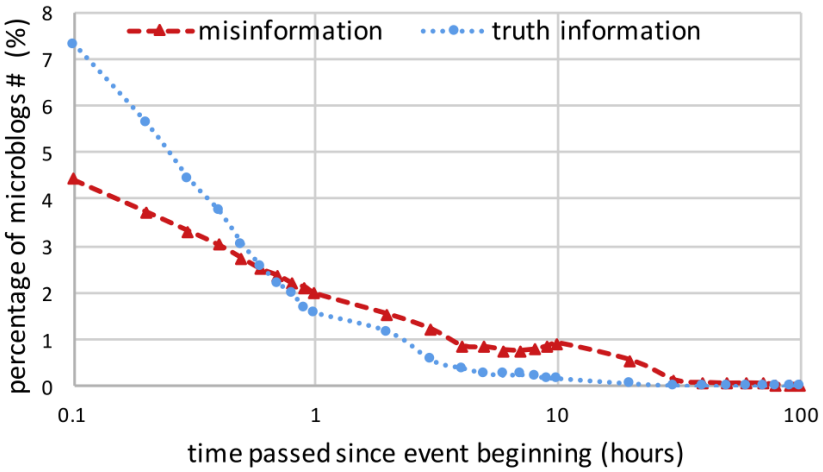


Figura 6.4: Distribuzione degli eventi nel tempo per notizie vere e false [78, 79].

Un interessante studio per quanto riguarda l'*early detection* è esposto in [50]; qui si considera che l'unica fonte disponibile per identificare se la notizia è vera o no è il testo stesso, privo di ogni commento. Gli autori, dunque, propongono di generare automaticamente delle risposte con una componente denominata **User Response Generator** (URG), la quale utilizza i commenti di articoli presenti nell'insieme di training per crearne di nuovi per l'articolo in esame. URG è composta da un Convolutional Variational Autoencoder che modella le relazioni tra l'articolo di training, la risposta dell'utente e una variabile latente generativa. Il risultato finale è un vettore della dimensione del vocabolario in cui ogni componente è la probabilità che quella parola appaia nei commenti degli utenti.
Per rappresentare il testo dell'articolo, invece, [50] ricorre ad una CNN a due livelli, condensando l'informazione contenuta nelle parole ad una rappresentazione delle frasi che poi verranno concatenate per catturare le informazioni semantiche e le feature testuali sia di articoli corti che lunghi.Ogni frase nell'articolo è rappresentata con un one-hot vector $\boldsymbol{f} \in \{0,1\}^{|V|}$, dove $|V|$ è la dimensione del vocabolario; in seguito, viene appicata una media degli embedding dei vettori delle parole nella frase per generare la rappresentazione desiderata. L'operazione è definita così:

$$\boldsymbol{v}(\boldsymbol{f}) = \frac{\mathbf{W}\boldsymbol{f}}{\sum_k f_k} \tag{6.2}$$

dove $\mathbf{W}$ è la matrice degli embedding delle parole provenienti da skip-gram e f_k è il k-esimo elemento di $\boldsymbol{f}$. La rappresentazione dell'articolo è data dalla concatenazione delle frasi che lo compongono e le feature c_i sono generate

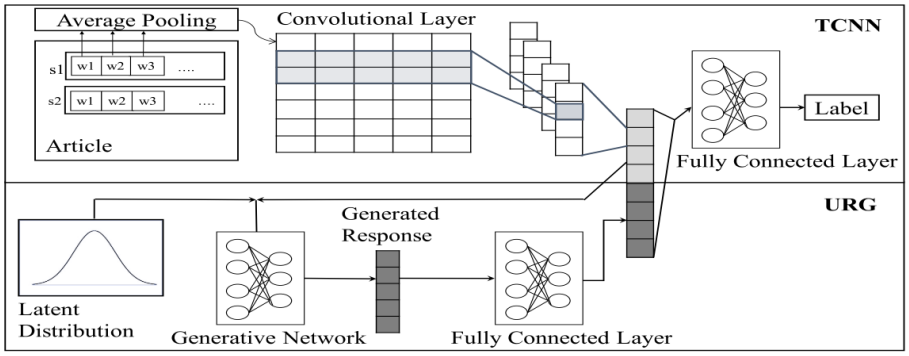


Figura 6.5: Architettura di [50].

Tabella 6.3: Risultati [50]. La percentuale nella prima riga indica la quantità di dati di training utilizzati.

	10%	30%	50%	70%	90%
TCNN	77,46%	78,21%	81,61%	84,29%	86,02%
TCNN+URG	77,47%	79,38%	81,92%	86,68%	88,83%

da un filtro t e una finestra di dimensione h lungo l'articolo:

$$c_i = g(t \cdot s_{i:i+h-1} + b)$$

dove g è una funzione d'attivazione e $b \in \mathbb{R}$ è il bias. Le feature ottenute sono poi concatenate alle risposte generate da URG ed usate come input in un fully connected layer per generare l'etichetta dell'articolo. L'intera architettura è raffigurata in 6.5 e in tabella 6.3 i risultati su un dataset collezionato agli autori. Si può notare che TCNN riporta buoni risultati anche quando utilizzato indipendentemente da URG, ma migliora le sue performance se ha a disposizione le risposte generate da URG.

6.1.2 Metodi con sorgente

Nell'ambito dell'identificazione delle Fake News tramite il testo e il feedback degli utenti, a volte viene tenuta in considerazione anche la sorgente dell'articolo, ossia l'URL da cui proviene, la credibilità del media che ha svolto la pubblicazione e il profilo del giornalista che l'ha scritta. Il metodo **CSI** presentato in [56] è un esempio di utilizzo delle sorgenti ed è formato da tre moduli:

1. *Capture*: cerca di catturare il numero di utenti che hanno interagito con l'articolo a_j e come queste interazioni sono spaziate nel tempo. Si utilizza una RNN il vettore di input è:

$$\mathbf{x}_t = (\eta, \Delta t, \mathbf{x}_u, \mathbf{x}_\tau) \tag{6.3}$$

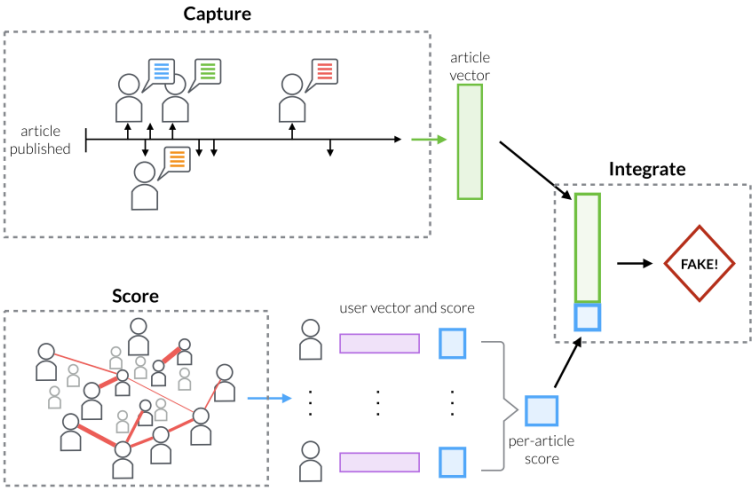


Figura 6.6: Intuizione dell'architettura di [56].

in cui η è il numero di interazioni, Δt è il tempo tra le interazioni, $\mathbf{x}_u$ è il vettore delle feature rappresentante l'utente e $\mathbf{x}_\tau$ cattura, tramite l'embedding Word2vec, il contenuto dell'interazione. Tale input è standardizzato tramite un livello di embedding prima di essere elaborato dalla LSTM usata. L'output è una rappresentazione delle risposte e delle caratteristiche testuali dell'articolo a_j;

2. *Score*: si occupa di capire se gli utenti sono inclini a promuovere una certa sorgente in base al comportamento. Il primo fully connected layer estrae la rappresentazione di ogni utente come segue:

$$\tilde{\mathbf{r}} = \tanh(\mathbf{W}_u \mathbf{r}_i + \mathbf{b}_u) \tag{6.4}$$

dove $\mathbf{W}_u$ è la matrice dei pesi e $\mathbf{b}_u$ è il bias. Lo score s_i per ogni utente è poi calcolato usando un vettore di pesi $\mathbf{w}_s$:

$$s_i = \sigma(\mathbf{w}_s^T \cdot \tilde{r} + b_s) \tag{6.5}$$

in cui σ è la sigmoide e b_s è il bias;

3. *Integrate*: combina tramite un fully connected layer i risultati dei due moduli precedenti al fine di produrre un'etichetta $\hat{y}_j$ per l'articolo considerato.

In figura 6.6 un'intuizione della struttura del modello. CSI risulta avere un'accuratezza del 89,2% sul dataset Twitter e del 95,4% su Weibo, valori superiori a quelli ottenibili senza l'utilizzo della sorgente.

6.1.3 Reputazione

Secondo [43], è importante anche considerare la reputazione dell'account della persona che ha pubblicato la notizia e del sito su cui la notizia si appoggia, ideando un approccio inspirato dall'individuazione delle spam email. Il metodo è denominato **Check-It** ed è costituito da quattro componenti:

- Flag-list Matcher: siccome alcuni siti sono noti per la pubblicazione di misinformazione, questo componente si occupa di verificare se l'URL della notizia contiene il nome di uno dei domini contenuti nella lista, detta flag-list, dei siti non affidabili;
- Fact Check Similarity: controlla se l'articolo in esame è presente in una lista di fatti già verificati da organizzazioni quali PolitiFact e Snopes, e genera un avvertimento se l'articolo fa parte della lista;
- Online Social Network User Analysis: lo scopo di questo modulo è creare una blacklist degli utenti in maniera dinamica. Tale obiettivo è raggiunto grazie ad un processing continuo dei dati nei social network e all'applicazione del modello probabilistico dell'utente di DeGroot presentato in [11]. Il punteggio di falsità di un determinato utente aumenta se quell'utente pubblica o condivide un post sospetto, ossia un post contenente un URL presente nella flag-list;
- Linguistic Model: analizza il contenuto della notizia, estraendo feature linguistiche dal titolo e dal corpo con l'utilizzo di una rete neurale allenata per predirre la veracità dell'articolo.

Per quanto riguarda la valutazione, Check-It viene confrontato con tre altri metodi della letteratura sui dataset di BuzzFeed e di PolitiFact, mostrando accuratezze rispettivamente del 70,3% e del 72,2%. Gli autori, però, sostengono che i dataset utilizzati non sono sufficientemente ampi per allenare una rete neurale e sostengono di ottenere un'accuratezza del 93% sulle categorie *fake news* e *credible* del *Fake News Corpus* per una totalità di 3 milioni di articoli. Oltre ad avere ottime performance quando allenato correttamente, Check-It è implementato come plugin per il browser e preserva la privacy dell'utente in quanto l'unica comunicazione esterna necessaria avviene all'installazione.

In tabella 6.4 si riassumono le principali caratteristiche e le relative accuratezze delle tecniche presentate nel capitolo.

Tabella 6.4

Metodo	**Caratteristiche**	**Acc.**
HBTP [28]	Indice di interesse assegnato ad utenti tramite processi di Dirichlet, indice di omogeneità composto da contesto e interesse, classificazione multipla supervisionata	78%
FakeNewsTracker [60]	Collezione giornaliera di notizie e commenti relativi, autoencoder per rappresentare articoli e LSTM per i commenti, classificazione binaria, visualizzazione (ad esempio, word heatmap e geolocalizzazione)	74,1%
TriFN [62]	Studio interazioni tra utenti-articoli e editori-articoli, classificazione binaria semi-supervisionata, early detection (performance 80% dopo 48 ore)	87,8%
HC-CB [12]	Contenuto stemmato e rappresentato con TF-IDF, feedback degli utenti rilevante solo se superiore a soglia, classificazione binaria	93,8%
ACAMI [78]	Contesto e timestap rappresentati tramite Event2vec, estrazione automatica di feature tramite CNN, classificazione binaria, early detection (performance massime entro 8 ore)	94,8%
User Response Generator [50]	Estrazione automatica di feature con CNN, URG per generazione automatica di commenti, classificazione binaria, early detection (non necessita di aspettare i commenti degli utenti)	88,83%
CSI [56]	Include reputazione della sorgente, embedding word2ve, RNN e LSTM per estrazione delle feature degli utenti e del testo, classificazione binaria	89,2%
Check-It [43]	Include reputazione della sorgente, basato sulle tecniche di individuazione di email spam, blacklist dinamica degli utenti, feature testuali automatiche tramite NN, privacy dell'utente rispettata, browser plug-in	93%

Capitolo 7

Confronto delle metodologie

Dopo aver esaminato e riassunto alcune delle tecniche di identificazione per le Fake News, in questo capitolo verranno introdotti alcuni risultati. Innanzitutto, in sezione 7.1 si riporta una tabella riassuntiva di tutti i metodi analizzati, segnalando quali delle caratteristiche principali sono fornite da una determinata metodologia. In seguito, verrà introdotta una metrica introdotta in [68] per valutare le varie performance dei vari algoritmi su dataset diversi, in maniera da fornire un confronto valido e determinare la tecnica migliore.

7.1 Caratteristiche delle tecniche analizzate

Nella presente sezione si riassumeranno in una tabella le varie tecniche precedentemente analizzate, mettendo in evidenza le caratteristiche principali che un particolare algoritmo possiede. Le caratteristiche incluse sono:

- **Attenzione**: per attenzione si intende che l'algoritmo in questione utilizza un meccanismo di attenzione per direzionare la concentrazione su alcune parti importanti degli articoli;
- **Classificazione Binaria**: significa che l'algoritmo presentato si concentra sul distinguere le notizie vere da quelle false utilizzando solo due classi;
- **Crowdsource**: si intende che la tecnica si avvale dell'aiuto degli utenti sottoforma di flagging o in altro modo per raggiungere l'obiettivo;
- **Early Detection**: l'algoritmo è in grado di distinguere le notizie vere in un tempo relativamente breve, a partire da qualche minuto fino ad alcune ore;
- **Embedding**: la metodologia utilizza un word embedding presentati in 2.1.3 o altre famose tecniche per l'embedding;

- **Classificazione Multiclasse**: significa che l'algoritmo presentato distingue le notizie vere da quelle false suddividendo la veracità in gradi diversi, ad esempio Vero, Quasi Vero, Mezzo Vero, Quasi Falso, Falso e Pants-on-Fire;
- **Propagazione**: in questo caso, la tecnica utilizza la propagazione della notizia per raggiungere l'obiettivo, rappresentando il sistema sottoforma di grafo;
- **Stance**: significa che i metodi sono utilizzati per risolvere il problema della Stance Detection come presentato in sezione 4.1;
- **Visualizzazione**: significa che l'algoritmo fornisce una rappresentazione visuale sul motivo del valore di verità assegnato per dare una spiegazione più chiara all'utente;
- **Web Search**: l'algoritmo sfrutta la ricerca web per ottenere maggiori informazioni sull'articolo da classificare.

Si può notare in tabella 7.1 che la maggior parte degli algoritmi di identificazione concentrano i loro sforzi in distinguere le notizie vere da quelle false in una classificazione binaria. Naturalmente, si tratta di un obiettivo importante, ma è altrettanto interessante riconoscere che gli articoli possono essere solo parzialmente falsi o parzialmente veri, problema che non viene affrontato spesso nella letteratura.

Un'altra osservazione importante riguarda gli embedding; come precedentemente accenato in sezione 2.1.3, questi vengono utilizzati in Reperimento dell'Informazione per ottenere rappresentazioni più compatte e per catturare i sinonimi. Per questo motivo, nonostante non siano ancora di uso comune nel campo delle Fake News, dimostrano di avere una buona potenzialità negli studi che sfruttano la tecnica del word embedding.

Al contrario, la visualizzazione sembra essere una caratteristica poco utilizzata da parte della ricerca in quanto solamente tre metodi ne sfruttano le potenzialità. Una visualizzazione grafica del grado di veracità e delle statistiche che hanno portato al risultato potrebbero essere essenziali per rendere i metodi di identificazione più approciabili da parte degli utenti e per fornire loro una visuale più critica nella lettura di altri articoli.

Infine, per quanto riguarda l'early detection, non sembra che molti studi si concentrino sulla questione, nonostante la sua enorme importanza. Individuare le notizie false prima che raggiungano la maggior parte delle persone dovrebbe essere una priorità da non sottovalutare, in modo da evitare tutte le conseguenze negative che le Fake News possono avere sulle persone. Naturalmente, si tratta di un problema più difficile da risolvere in quando se si vuole assegnare la veracità al più presto, si possono disporre solo di alcune caratteristiche, quali il contenuto dell'articolo e la sorgente, riducendo dunque le feature disponibili ai modelli.

Tabella 7.1

	Attenzione	**Binaria**	**Crowdsource**	**Early Det.**	**Embedding**
ACAMI [78]		X		X	X
Ajao [2]		X			
Best Combination [19]		X			X
CAMI [79]		X		X	X
Check-It [43]		X			
Classificatori Combinati [8]		X			X
ComLex [24]		X			X
Crowd Signals [71]			X		
CSI [56]		X			X
CURB [29]			X		
DeClarE [47]	X	X			X
DistrustRank [75]		X			
FActCheck [63]					
FakeNewsTracker [60]		X			
Grounded Truth [70]		X			
HBTP [28]					
HC-CB [12]		X			
Liu-Wu [32]		X		X	
MMFD [27]					
QuickStop [73]		X		X	
Rashkin [52]		X			X
Related Fact Check [20]					
Souvick-Chirag [16]		X			
Stance Detection Ranking [82]					
Trace Miner [76]		X		X	
TriFN [62]		X		X	
Undeutsch Hypothesis [83]		X		X	
User Response Generator [50]		X		X	
Web Search [26]	X				X
XFake [77]		X			X

Tabella 7.2

	Multiclasse	**Propagazione**	**Stance**	**Visual.**	**Web Search**
ACAMI [78]					
Ajao [2]					
Best Combination [19]					
CAMI [79]					
Check-It [43]					
Classificatori Combinati [8]			X		
ComLex [24]					
Crowd Signals [71]					
CSI [56]					
CURB [29]					X
DeClarE [47]					
DistrustRank [75]					
FActCheck [63]		X			
FakeNewsTracker [60]				X	
Grounded Truth [70]					
HBTP [28]	X				
HC-CB [12]					
Liu-Wu [32]		X			
MMFD [27]	X				
QuickStop [73]		X			
Rashkin [52]	X				
Related Fact Check [20]				X	X
Souvick-Chirag [16]	X				
Stance Detection Ranking [82]			X		
Trace Miner [76]		X			
TriFN [62]					
Undeutsch Hypothesis [83]					
User Response Generator [50]					
Web Search [26]					X
XFake [77]				X	

7.2 Metodo per comparare performance calcolate su dataset diversi

Molti degli algoritmi studiati per l'identificazione delle Fake News vengono testati su dataset diversi e ciò rende impossibile svolgere un semplice confronto tra le performance per determinare il miglior algoritmo. Per questo motivo si presenta ora una possibile metrica di confronto ideata nel campo del learning to rank.

7.2.1 Winning Number

Il *Winning Number* è una metrica introdotta in [51] per gli algoritmi di ranking nella situazione ideale in cui tutte le tecniche forniscono misure delle performance per ogni dataset disponibile. Gli autori osservano che alcuni algoritmi hanno performance migliori su alcuni dataset rispetto ad altri. Per valutare la performance generale di un algoritmo, viene proposto di utilizzare il numero di altri algoritmi che può battere su tutti i sette dataset disponibili. Tale valore viene denominato Winning Number ed è definito da:

$$\mathrm{WN}_i(M) = \sum_{j=1}^{n} \sum_{k=1}^{m} \mathbf{1}_{(M_i(j) > M_k(j))} \tag{7.1}$$

dove M è una misura delle performance (ad esempio, MAP per il ranking e F1-score per il campo delle Fake News), j è l'indice di un dataset, i e k sono indici degli algoritmi, $M_i(j)$ è il valore della misura della performance M per l'algoritmo i nel dataset e $\mathbf{1}_{(M_i(j) > M_k(j))}$ è la funzione indicatrice seguente:

$$\mathbf{1}_{(M_i(j) > M_k(j))} = \begin{cases} 1 & \text{se } M_i(j) > M_k(j), \\ 0 & \text{altrimenti} \end{cases}$$

È evidente che maggiore sarà il Winner Number, migliore saranno le performance dell'algoritmo.

7.2.2 Normalized Winning Number

La situazione presentata in [51] non è applicabile quando gli algoritmi non sono testati su tutti i dataset disponibili. Per questo motivo, gli autori di [68] propongono una versione normalizzata del Winning Number per permettere il confronto in un insieme sparso di risultati. Innanzitutto, viene modificata la funzione indicatrice come segue:

$$\mathbf{1'}_{(M_i(j)>M_k(j))} = \begin{cases} 1 & \text{se } M_i(j) \text{ e } M_k(j) \text{ sono entrambi definiti} \\ & \text{e } M_i(j) > M_k(j), \\ 0 & \text{altrimenti} \end{cases}$$

Il *Normalized Winning Number* è il rapporto tra il Winning Number con la funzione indicatrice modificata e il Winning Number Ideale (IWN, Ideal Winning Number), ossia il Winning Number che avrebbe avuto se l'algoritmo fosse stato il migliore su tutti i dataset su cui è stato valutato.

$$\mathrm{NWN}_i(M) = \frac{\mathrm{WN}_i(M)}{\mathrm{IWN}_i(M)} \tag{7.2}$$

dove $\mathrm{IWN}_i(M)$ è il Winning Number Ideale definito da:

$$\mathrm{IWN}_i(M) = \sum_{j=1}^{n} \sum_{k=1}^{m} \mathbf{D}_{(M_i(j),M_k(j))} \tag{7.3}$$

dove $\mathbf{D}$ è la funzione di valutazione è:

$$\mathbf{D}_{(M_i(j),M_k(j))} = \begin{cases} 1 & \text{se } M_i(j) \text{ e } M_k(j) \text{ sono entrambi definiti,} \\ 0 & \text{altrimenti} \end{cases}$$

7.2.3 Confronto degli algoritmi analizzati

Nella seguente sottosezione si riporteranno i valori di Normalized Winning Number per gli algoritmi precedentemente analizzati. Le performance verificate riguardano le due metiche più comuni: accuratezza e F1-score.

7.2.3.1 Valori di NWN per l'accuratezza

La figura 7.1 riporta in forma grafica i risultati del Normalized Winning Number per quanto riguarda l'accuratezza. Si può notare che classificatori combinati [8], Web Search [26] e CSI [56] ottengono un NWN pari a 1. Tra questi, solo [56] riporta risultati su due dataset, mentre gli altri testano la tecnica su solo un dataset. Un altro risultato importante è raggiunto da Undeutsch Hypothesis [83].

Alcuni degli studi riportati nei capitoli 4, 5 and 6 non sono stati riportati in questo confronto in quanto non forniscono dati sull'accuratezza del loro algoritmo o non si concentrano sull'identificazione, ma sull'inviduazione di articoli da inviare agli esperti per la verifica.

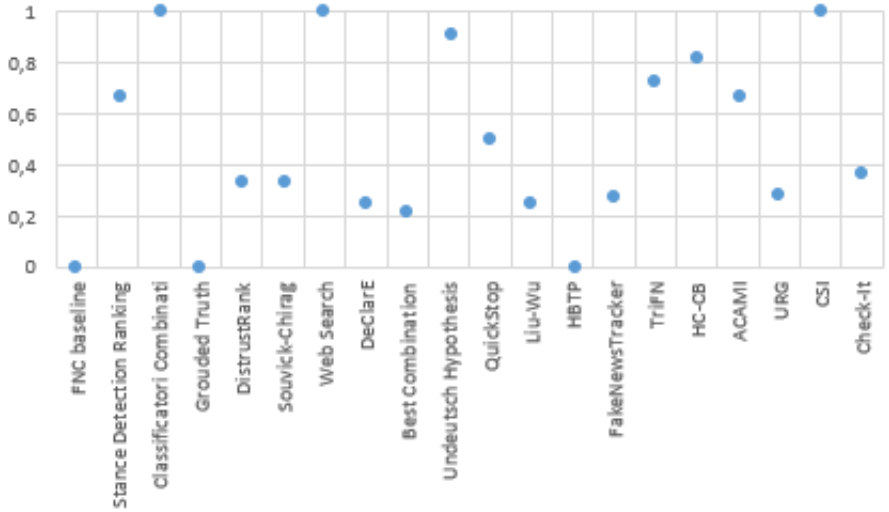


Figura 7.1: Normalized Winning Number per l'accuratezza.

7.2.3.2 Valori di NWN per l'F1-score

La figura 7.2 riporta in forma grafica i risultati del Normalized Winning Number per quanto riguarda l'F1-score. Si può notare che DistrustRank [75] e TraceMiner [76] ottengono un NWN pari a 1. Entrambi sono valutati su un solo dataset. Altri risultati importanti sono ottenuti da Undeutsch Hypothesis [83] e HC-CB [12], entrambi valutati su due dataset.

Alcuni degli studi riportati nei capitoli 4, 5 and 6 non sono stati riportati in questo confronto in quanto non forniscono dati sulla F1-measure del loro algoritmo o non si concentrano sull'identificazione, ma sull'inviduazione di articoli da inviare agli esperti per la verifica.

7.2.3.3 Tabella dei risultati

In tabella 7.3 si riportano i valori di Normalized Winning Number relativi ad entrambe le misure; si riporta inoltre anche il numero di dataset su cui una data tecnica è stata testata.

Il valore *undefined* riportato per la tecnica presentata in [2] indica che non esistono algoritmi analizzati che utilizzano lo stesso dataset, mentre il valore *n/a* significa che gli autori dell'articolo non riportano quella metrica per la loro tecnica. Anche in questo caso si sono omessi quegli studi che concentrano i loro sforzi nell'individuare articoli da inviare a degli esperti e le tecniche che forniscono sorgenti esterne per permettere agli utenti di giudicare la vericità di un articolo autonomamente.

Tabella 7.3: Valori di Normalized Winning Number per le tecniche analizzate

	Dataset	NWN (Accuratezza)	NWN (F1-score)
FNC Baseline	1	0	n/a
Stance Detection Ranking [82]	1	0,667	n/a
Classificatori Combinati [8]	1	1	n/a
Grounded Truth [70]	1	0	0
DistrustRank [75]	1	0,333	1
Souvick-Chirag [16]	1	0,333	n/a
Web Search [26]	1	1	0
DeClarE [47]	4	0,25	0,286
Best Combination [19]	5	0,214	n/a
Ajao-Bhowmik-Zangari [2]	1	undefined	n/a
Undeutsch Hypothesis [83]	2	0,909	0,9
QuickStop [73]	1	0,5	n/a
Liu-Wu [32]	3	0,25	0
HBTP [28]	1	0	0
FakeNewsTracker [60]	2	0,273	0,2
TriFN [62]	2	0,727	0,7
HC-CB [12]	2	0,818	0,8
ACAMI [78]	2	0,667	0,4
User Response Generator [50]	2	0,286	n/a
CSI [56]	2	1	0,8
Check-It [43]	3	0,364	0,4

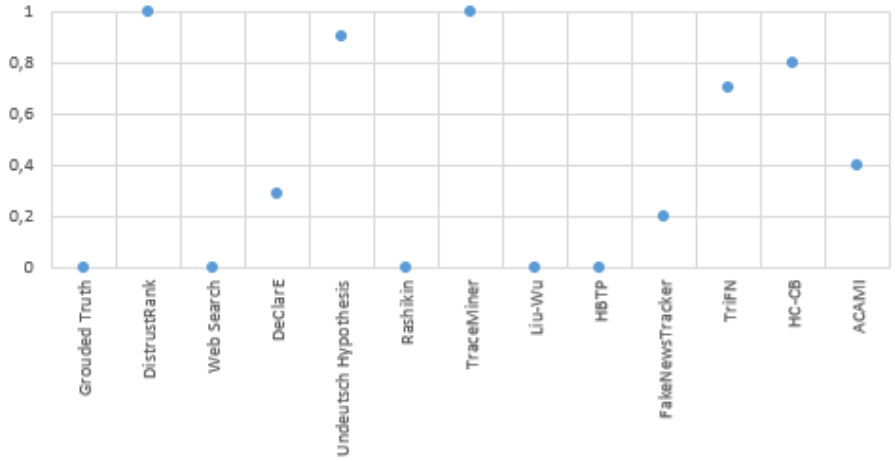


Figura 7.2: Normalized Winning Number per l'F1-score.

In figura 7.3 si riportano tutti i valori di Normalized Winning Number calcolati.

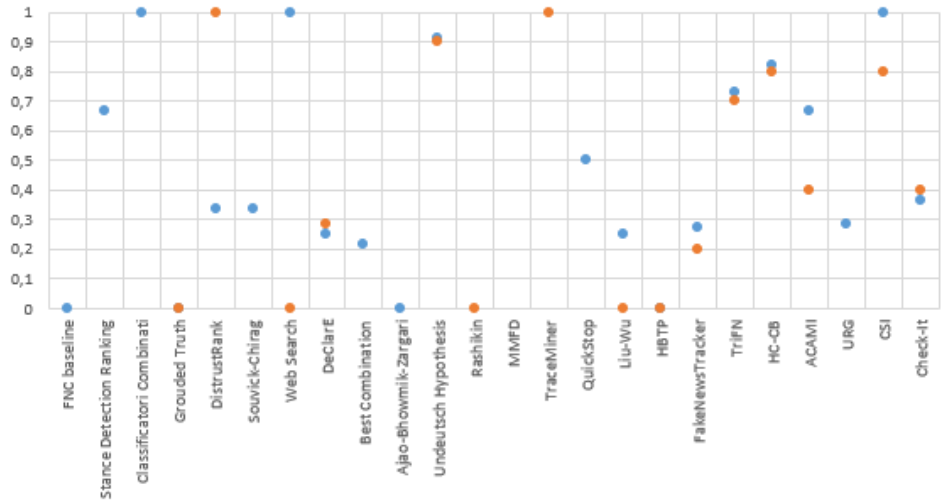


Figura 7.3: Visualizzazione dei risultati del Normalized Winning Number. In blu, i valori di NWN per l'accuratezza e in arancio quelli per l'F1-measure.

Alcuni metodi ottengono un Normalized Winning Number pari a 1, ma non possono essere considerati i migliori in quando si tratta di tecniche verificate su un solo dataset. Allo stesso modo, alcune tecniche risultano essere peggiori, con valore zero o indefinito di NWN, poiché testate su un numero limitato di dataset. Queste metodologie non sono necessariamente

da scartare in quando potrebbero dimostrare performance migliori in altre collezioni.

In base ai risultati ottenuti, i metodi migliori tra quelli presentati sono CSI [56] e Undeutsch Hypothesis [83] in quanto ottengono un buon NWN per entrambe le metriche utilizzate. Entrambe sono tecniche di classificazione binaria e, mentre Undeutsch Hypothesis utilizza solo il contenuto, CSI sfrutta anche il feedback da parte degli utenti. Importante notare che [83] è un metodo di early detection, a dimostrare che è possibile ottenere buone performance anche con una quantità limitata di informazioni sull'articolo.

Un discorso a parte va fatto per metodi come Check-It [43] perché sfruttano tecniche di Deep Learning e, per questo motivo, hanno bisogno di dataset di ampie dimensioni per avere sufficienti dati per allenare la rete neurale. Tali tecniche raggiungono performance spesso deludenti nelle collezioni disponibili, ma [43] dimostra che con l'utilizzo di un dataset più grande come il Fake News Corpus può raggiungere un'accuratezza maggiore al 90%. Questo dimostra ancora una volta che i dataset sviluppati per le Fake News non sono sufficienti a soddisfare le necessità dei ricercatori.

In generale, i risultati ottenuti tramite il Normalized Winning Number dimostra che le tecniche che utilizzano il contenuto (da solo o accompagnato da altre feature) sono superiori rispetto a quelle basate sulla propagazione o sul feedback. Tra queste, molte sono state svilupate con l'obiettivo di raggiungere un'identificazione in breve tempo.

Purtroppo, però, il Normalized Winning Number non è una misura perfetta di confronto e presenta delle limitazioni.

7.2.4 Limitazioni

Gli autori di [68] notano che il Normalized Winning Number presenta le seguenti limitazioni:

- Il modo in cui NWN è calcolato assume implicitamente che i metodi analizzati siano approssimatamente distribuiti uniformemente lungo i dataset, ma quest'assunzione potrebbe essere falsa. Nel caso delle Fake News, per i casi esaminati in questa tesi, l'assunzione è falsa in quanto esistono pochi dataset e, spesso, i ricercatori testano i loro algoritmi solo su uno o due dei dataset disponibili;

- I dataset scelti dai ricercatori solitamente non rappresentano una scelta casuale. Potrebbe essere che venga scelto uno o più particolari dataset per mostrare i risultati migliori dell'algoritmo proposto;

- La metodologia di confronto tramite NWN assume che i risultati pubblicati nella letteratura siano corretti. Questo introduce il rischio di includere risultati troppo ottimistici nel risultato finale.

Oltre alle limitazioni presentate, in questa tesi sono state analizzate solo un numero limitato di tecniche, riducendo naturalmente la qualità del NWN.

Capitolo 8

Conclusioni e sviluppi futuri

Lo scopo di questa tesi è stato studiare e riassumere varie tecniche di identificazione e mitigazione per le Fake News sviluppate negli ultimi anni, dimostrando quanto la ricerca abbia concentrato i propri sforzi per ridurre gli effetti negativi che le notizie false possono causare.

Si è parlato nel capitolo 1 della definizione di Fake News e delle varie tipologie di notizia falsa identificabili, mentre nel capitolo 2 si sono introdotti alcuni concetti fondamentali provenienti da vari campi scientifici spesso utilizzati nel campo delle Fake News; nel capitolo 3 si è presentato il problema della mancanza dei dataset, le organizzazioni che si occupano di verificare manualmente le notizie e le challenge nate allo scopo di stimolare la ricerca. Nei capitoli 4, 5 e 6 si sono analizzate e riassunte alcune delle tecniche per l'identificazione delle Fake News, classificate in base alle caratteristiche ritenute essenziali dai vari autori, come il contenuto, la propagazione nella rete e i feedback da parte dagli utenti.

Infine, nel capitolo 7 si sono riportate tabelle riassuntive delle caratteristiche principali appartenenti ai metodi analizzati, individuando come la maggior parte si concentrano nella classificazione binaria nonostante le notizie false possano talvolta presentare un grado di veracità non binario. In seguito, si è introdotta una metrica denominata *Normalized Winning Number* per poter confrontare i metodi nonostante i diversi dataset utilizzati, individuando quali tecniche presentano performance migliori, ma riportando anche le possibili limitazioni che tale metrica può avere.

È, però, importante notare come esistano ancora molte sfide aperte che richiedono maggiore sviluppo. La prima sfida riguarda i dataset che, come riportato precedentemente, rappresentano una delle maggiori difficoltà nel campo per la loro scarsità o per la loro specificità ad un certo utilizzo. Una delle più grandi sfide, dunque, è data dalla costruzioni di dataset più generici, similmente a quando fatto dal Fake News Corpus presentato in sezione 3.1. Inoltre, tali collezioni dovrebbero essere mantenute aggiornate per evitare che diventino inutilizzabili nel giro di breve tempo per via della produzione

giornaliera di notizie e del continuo evolversi delle Fake News allo scopo di evitare di essere identificate.

Una seconda sfida riguarda la classificazione con vari gradi di veracità; nonostante esistano degli studi sull'argomento, spesso i risultati risultano essere non sufficienti da ritenerli affidabili, perciò ulteriori ricerche potrebbero essere interessanti; in particolare, oltre a creare dei dataset contenenti non solo etichette binarie, si potrebbero cercare metodi per distinguere tra loro le varie categorie (ad esempio, la satira dalle Fake News).

Infine, nei recenti anni si sono sviluppate maggiormente tecniche per fabbricare audio, immagini e video. Particolarmente preoccupante nel campo risulta essere il Deepfake, tecnica basata sull'intelligenza artificiale usata per combinare e sovrapporre immagini e video esistenti con video o immagini originali tramite tecniche di Machine Learning e Deep Leaning [57]. Questi audio o video falsi possono essere davvero credibili, anche se finora la maggior parte sono limitati a principianti che lo fanno per hobby. Il problema, però, diventa preoccupante nel momento in cui i Deepfake vengono sfruttati per annunciare attacchi terroristici imminenti o per minare delle elezioni politiche.

Secondo [49], i Deepfake sfruttano le tendenze umane di credere a ciò che si vede tramine un Rete Generativa Avversaria (GAN) formata da due modelli di Machine Learning. Un modello si occupa di creare il video falso e l'altra si occupa di individuare le falsificazioni. La prima crea falsi finché il secondo modello non è più in grado di identificare che il video o l'audio è falso. Per questo determinare quali video siano falsi può essere un problema molto difficile se il lavoro proviene da dei professionisti. Non riuscire a capire quali video siano Deepfake risulterebbe in una situazione in cui non sarebbe più possibile fidarsi di ciò che si sente e vede. Per questo motivo, è essenziale che vengano concentrati gli sforzi della ricerca in questo campo per contrastare gli effetti negativi generati dal Deepfake.

Bibliografia

[1] Perceptrons and multi-layer perceptrons: The artificial neuron at the core of deep learning. https://missinglink.ai/guides/neural-network-concepts/perceptrons-and-multi-layer-perceptrons-the-artificial-neuron-at-the-core-of-deep-learning/.

[2] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. Fake News Identification on Twitter with Hybrid CNN and RNN Models. *Proceedings of the 9th International Conference on Social Media and Society*, pages 226–230, 2018.

[3] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[4] Arunava. The perceptron. 2018. https://towardsdatascience.com/the-perceptron-3af34c84838c.

[5] Christopher M Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.

[6] Nicoletta Boldrini. Deep learning, cos'è l'apprendimento profondo, come funziona e quali sono i casi di applicazione. 2019. https://www.ai4business.it/intelligenza-artificiale/deep-learning/deep-learning-cose/.

[7] Alessandro Bondielli and Francesco Marcelloni. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55, 2019.

[8] Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. From clickbait to fake news detection: an approach based on detecting the stance of headlines to articles. *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 84–89, 2017.

[9] Chandra Churh Chatterjee. Implementation of RNN, LSTM, and GRU. 2019. https://towardsdatascience.com/implementation-of-rnn-lstm-and-gru-a4250bf6c090.

[10] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[11] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[12] Marco L Della Vedova, Eugenio Tacchini, Stefano Moret, Gabriele Ballarin, Massimo DiPierro, and Luca de Alfaro. Automatic Online Fake News Detection Combining Content and Social Signals. *2018 22nd Conference of Open Innovations Association (FRUCT)*, pages 272–279, 2018.

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer series in statistics New York, 2001.

[14] David Fumo. Types of machine learning algorithms you should know. 2017. `https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861`.

[15] J. Micheal Garbade. A simple introduction to Natural Language Processing. 2018. `https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32`.

[16] Souvick Ghosh and Chirag Shah. Towards Automatic Fake News Classification. *Proceedings of the Association for Information Science and Technology*, 55(1):805–807, 2018.

[17] Jennifer Golbeck, Matthew Mauriello, Brooke Auxier, Keval H Bhanushali, Christopher Bonk, Mohamed Amine Bouzaghrane, Cody Buntain, Riya Chanduka, Paul Cheakalos, Jennine B Everett, et al. Fake news vs satire: A dataset and analysis. *Proceedings of the 10th ACM Conference on Web Science*, pages 17–21, 2018.

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

[19] Georgios Gravanis, Athena Vakali, Konstantinos Diamantaras, and Panagiotis Karadais. Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, 128:201 – 213, 2019.

[20] Sreya Guha. Related Fact Checks: a tool for combating fake news. *arXiv preprint arXiv:1711.00715*, 2017.

[21] Shubham Gupta. Decision tree. `https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/`.

[22] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. *Proceedings of the 2012 siam international conference on data mining*, pages 463–474, 2012.

[23] Benjamin D Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[24] Shan Jiang and Christo Wilson. Linguistic Signals under Misinformation and Fact-Checking: Evidence from User Comments on Social Media. *Proceedings of the ACM on Human-Computer Interaction*, 2:82, 2018.

[25] Mark Johnson. How the statistical revolution changes (computational) linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 3–11. Association for Computational Linguistics, 2009.

[26] Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. Fully Automated Fact Checking using External Sources. *arXiv preprint arXiv:1710.00341*, 2017.

[27] Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. Multi-Source Multi-Class Fake News Detection. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1546–1557, 2018.

[28] Jooyeon Kim, Dongkwan Kim, and Alice Oh. Homogeneity-Based Transmissive Process to Model True and False News in Social Networks. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 348–356, 2019.

[29] Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Leveraging the Crowd to Detect and Reduce the Spread of Fake News and Misinformation. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 324–332, 2018.

[30] Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*, 2018.

[31] Nathan Landman, Hannah Pang, Christopher Williams, and Eli Ross. k-means clustering. 2018. `https://brilliant.org/wiki/k-means-clustering/`.

[32] Yang Liu and Yi-Fang Brook Wu. Early Detection of Fake News on Social Media through Propagation Path Classification with Recurrent and Convolutional Networks. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[33] Gabriel Loye. Attention mechanism. 2019. `https://blog.floydhub.com/attention-mechanism/`.

[34] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. *Ijcai*, pages 3818–3824, 2016.

[35] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717, 2017.

[36] Pankaj Mathur. A simple multilayer perceptron with tensorflow. 2016. `https://medium.com/pankajmathur/a-simple-multilayer-perceptron-with-tensorflow-3effe7bf3466`.

[37] Milos Miljanovic. Comparative analysis of recurrent and finite impulse response neural networks in time series prediction. *Indian Journal of Computer Science and Engineering*, pages 180–191, 2012.

[38] Tom M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[39] Subhabrata Mukherjee and Gerhard Weikum. Leveraging joint interactions for credibility analysis in news communities. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 353–362, 2015.

[40] Preslav Nakov, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Lluís Màrquez, Wajdi Zaghouani, Pepa Atanasova, Spas Kyuchukov, and Giovanni Da San Martino. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 372–387, 2018.

[41] Nam P Nguyen, Guanhua Yan, My T Thai, and Stephan Eidenbenz. Containment of misinformation spread in online social networks. *Proceedings of the 4th Annual ACM Web Science Conference*, pages 213–222, 2012.

[42] Vibhor Nigam. Understanding neural networks. from neuron to RNN, CNN, and deep learning. 2018. `https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90`.

[43] Demetris Paschalides, Alexandros Kornilakis, Chrysovalantis Christodoulou, Rafael Andreou, George Pallis, Marios D. Dikaiakos, and Evangelos P. Markatos. Check-It: A Plugin for Detecting and Reducing the Spread of Fake News and Misinformation on the Web. *CoRR*, 2019.

[44] Ashish Patel. Bagging - ensemble meta algorithm for reducing variance. 2019. `https://medium.com/ml-research-lab/bagging-ensemble-meta-algorithm-for-reducing-variance-c98fffa5489f`.

[45] Ashish Patel. Bagging - ensemble meta algorithm for reducing variance. 2019. `https://medium.com/ml-research-lab/bagging-ensemble-meta-algorithm-for-reducing-variance-c98fffa5489f`.

[46] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1003–1012, 2017.

[47] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. *arXiv preprint arXiv:1809.06416*, 2018.

[48] Martin F Porter. An algorithm for suffix stripping. *Program*, 1980.

[49] J.M. Porup. How and why deepfake videos work - and what is at risk. 2019. `https://www.csoonline.com/article/3293002/deepfake-videos-how-and-why-they-work.html`.

[50] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. Neural User Response Generator: Fake News Detection with Collective User Intelligence. *IJCAI*, pages 3834–3840, 2018.

[51] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

[52] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[53] Sunil Ray. Commonly used machine learning algorithms (with python and R codes). 2017. `https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/`.

[54] Alessandro Rezzani. *Big Data: Architettura, tecnologie e metodi per l'utilizzo di grandi basi di dati.* Maggioli Editore, 2013.

[55] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[56] Natali Ruchansky, Sungyong Seo, and Yan Liu. CSI: A Hybrid Deep Model for Fake News Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806, 2017.

[57] Oscar Schwartz. You thought fake news was bad? deep fakes are where truth goes to die. *The Guardian*, 2018.

[58] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms.* Cambridge university press, 2014.

[59] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM Trans. Intell. Syst. Technol.*, 10(3):21:1–21:42, April 2019.

[60] Kai Shu, Deepak Mahudeswaran, and Huan Liu. Fakenewstracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 25(1):60–71, 2019.

[61] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[62] Kai Shu, Suhang Wang, and Huan Liu. Beyond News Contents: The Role of Social Context for Fake News Detection. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 312–320, 2019.

[63] Ajitesh Srivastava, Rajgopal Kannan, Charalampos Chelmis, and Viktor K Prasanna. FactCheck: Keeping Activation of Fake News at Check. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2079–2081, 2018.

[64] Tavish Srivastava. Introduction to k-Nearest Neighbors: A powerful machine learning algorithm (with implementation in python & R). 2018. `https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/`.

[65] James H Stock, Mark W Watson, et al. *Regression with a Binary Dependent Variable, in Introduction to Econometrics,*. Addison Wesley Boston, 2003.

[66] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.

[67] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.

[68] Niek Tax, Sander Bockting, and Djoerd Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Information processing & management*, 51(6):757–772, 2015.

[69] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[70] Terry Traylor, Jeremy Straub, Nicholas Snell, et al. Classifying Fake News Articles Using Natural Language Processing to Identify In-Article Attribution as a Supervised Learning Estimator. pages 445–449. IEEE, 2019.

[71] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. Fake News Detection in Social Networks via Crowd Signals. *Companion Proceedings of the The Web Conference 2018*, pages 517–524, 2018.

[72] William Yang Wang. "liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *CoRR*, 2017.

[73] Honghao Wei, Xiaohan Kang, Weina Wang, and Lei Ying. QuickStop: A Markov Optimal Stopping Approach for Quickest Misinformation Detection. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(2):41, 2019.

[74] Terry Winograd. Procedures as a representation for data in a computer program for understanding natural language. Technical report, 1971.

[75] Vinicius Woloszyn and Wolfgang Nejdl. DistrustRank: Spotting False News Domains. *Proceedings of the 10th ACM Conference on Web Science*, pages 221–228, 2018.

[76] Liang Wu and Huan Liu. Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 637–645, 2018.

[77] Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Ben Hu. XFake: Explainable Fake News Detector with Visualizations. *The World Wide Web Conference*, pages 3600–3604, 2019.

[78] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security*, 83:106–121, 2019.

[79] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. A Convolutional Approach for Misinformation Identification. 2017.

[80] Sixie Yu, Yevgeniy Vorobeychik, and Scott Alfeld. Adversarial Classification on Social Networks. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 211–219, 2018.

[81] Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans. *J. Data and Information Quality*, 11(3):10:1–10:37, 2019.

[82] Qiang Zhang, Emine Yilmaz, and Shangsong Liang. Ranking-based method for news stance detection. *Companion Proceedings of the The Web Conference 2018*, pages 41–42, 2018.

[83] Xinyi Zhou, Atishay Jain, Vir V. Phoha, and Reza Zafarani. Fake News Early Detection: A Theory-driven Model. *CoRR*, 2019.

[84] Xinyi Zhou and Reza Zafarani. Fake News: A Survey of Research, Detection Methods, and Opportunities. *ACM Comput. Surv. 1*, 2018.

[85] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3), 2016.