POLITECNICO DI TORINO
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Caratterizzazione semantica di una collezione di documenti mediante tecniche probabilistiche

Relatore
Prof.ssa Tania Cerquitelli

Candidata
Elena Citerà

Sessione
Ottobre 2018

Sommario

In questi ultimi anni il volume dei dati generati nella rete è cresciuto enormemente, trasformando radicalmente le nostre abitudini. Lo stesso trend può essere applicato anche a collezioni di dati testuali che oggigiorno sono prodotti ad una velocità molto alta, dai social network alle librerie e enciclopedie digitali. Data la complessità dei dati testuali e la difficoltà di interpretare i testi, il text mining e il topic modeling rappresentano due delle attività più impegnative del data mining, nonché aree di grande interesse nella comunità scientifica. In letteratura, diverse tecniche di analisi sono state proposte al fine di classificare i documenti in base ai contenuti tematici, trovare associazioni nascoste e estrarre informazione utile. Poiché il text mining è un processo composto da numerose fasi, ciascuna delle quali richiede parametri e configurazioni specifiche, sono necessarie delle soluzioni innovative nell'analisi, nella visualizzazione e nella validazione dei risultati, al fine di estrarre una conoscenza che sia efficace ed effettiva. Il lavoro di tesi ha quindi lo scopo di implementare un nuovo framework in grado di raggruppare automaticamente i documenti di testo in gruppi coesi e ben separati, in base al loro contenuto. Per fare questo, è stato creato CONCEPT (*CharacterizatiON of a textual Collection through the Evaluation of Probabilistic Techniques*), un framework innovativo che utilizza il modello statistico PLSA, in grado di descrivere rapidamente grandi collezioni di documenti senza ricorrere alla riduzione di dimensionalità dei dati. In particolare, gli argomenti e le parole sono descritti come distribuzioni probabilistiche dalle quali poi i documenti saranno creati. Nella sua implementazione, CONCEPT include diverse combinazioni di pesi locali e globali assegnate alle parole del corpus ed è stato convalidato su diverse collezioni di documenti, con strutture diverse: articoli di Wikipedia molto lunghi e vaste collezioni di tweet. I risultati sperimentali ottenuti sono stati valutati prendendo in considerazione diverse metriche come Log-likelihood, Silhouette (ASI e GSI) e Rand Index, ma anche mediante tecniche di visualizzazione innovative (i.e. grafici t-SNE, stacked-bar, scatter plot, word cloud e grafo). Grazie al framework sviluppato in supporto all'utente, quest'ultimo riesce a visualizzare in maniera ottimale e semplice i risultati prodotti, senza essere un esperto di dominio. Infine, i risultati hanno mostrato che CONCEPT risulta essere efficace nel processo di clustering dei documenti e fanno premettere che questo algoritmo possa essere applicato anche a collezioni di dati più grandi.

Indice

Capitolo 1

1 Introduzione

Con le moderne applicazioni e tecnologie, il volume dei dati generati è in continua crescita. Oggi la tecnologia sta radicalmente trasformando le nostre abitudini: viviamo in un mondo sempre più connesso e digitalizzato in cui, ad esempio, l'utilizzo degli smartphone, dei pc, dei social network, della tessera dei trasporti pubblici, e ogni altra nostra singola attività crea una grande quantità di dati. Questo trend vale anche per *collezioni di dati testuali*, che, ad esempio per mezzo di librerie e enciclopedie digitali, oggigiorno sono prodotti ad un ritmo sempre maggiore. I dati raccolti generano dataset molto grandi, i cui domini e distribuzioni sono per lo più ignoti e variano considerevolmente tra di loro. Per questo motivo le operazioni di data mining, e in particolare nel caso testuale le operazioni di text mining e topic modeling, sono complesse e le scelte riguardanti le migliori metodologie da usare nell'analisi, nella validazione dei risultati e nell'estrazione di una conoscenza che sia efficace ed effettiva, sono un problema noto e attuale che necessita approfondimenti e nuovi strumenti per aiutare gli utenti finali, senza che essi siano esperti di dominio. Essendo il text mining un processo a più fasi che richiede configurazioni e parametri specifici per ogni algoritmo coinvolto nel processo, dovrebbe essere richiesta la presenza di esperti e analisti del settore dell'analisi testuale. Per questo motivo, data la complessità di tali fasi, obiettivo di questa tesi è sviluppare un framework in grado di visualizzare i risultati prodotti dalla clusterizzazione in maniera ottimale, intuitiva e semplice, in modo tale che l'utente possa visualizzare e validare i risultati dell'analisi efficacemente.

Il lavoro di tesi ha prodotto CONCEPT (*CharacterizatiON of a textual Collection through the Evaluation of Probabilistic Techniques*), un sistema per l'analisi di dati testuali con l'obiettivo di raggruppare automaticamente i documenti di testo in gruppi coesi e ben separati, in base al loro contenuto.

Per fare questo è stato utilizzato un nuovo approccio basato sul modello statistico *Probabilistic Latent Semantic Analysis*. Dopo la progettazione del modello e dei suoi singoli componenti, i risultati prodotti sono visualizzati attraverso rappresentazioni approfondite quali *grafici t-SNE*, *scatter plot*, *stacked-bar*, *grafo* e *word cloud*, e validati attraverso indici di qualità.
Gli esiti ottenuti sono promettenti e in gran parte soddisfacenti: grazie al framework sviluppato in supporto all'utente, quest'ultimo riesce analizzare con semplicità le diverse configurazioni. Infine, i risultati fanno premettere che tale framework possa essere applicato anche a grandi collezioni di dati.

Il lavoro di tesi è strutturato come segue. Il Capitolo 2 presenta una panoramica generale sull'analisi dei dati testuali. Descrive i principali modelli statistici, noti in letteratura, per l'analisi tematica di documenti, in particolare quello algebrico LSA e quello probabilistico LDA, e per entrambi sono citati esempi di tool sviluppati. Il Capitolo 3 introduce ai lettori il framework CONCEPT, la sua implementazione e nuovo approccio nel text mining. Il capitolo descrive il modello probabilistico utilizzato, PLSA, descrivendo nel dettaglio il flusso analitico e le fasi di elaborazione del framework. Nel Capitolo 3 sono anche presentate le tecniche di visualizzazione e validazione per valutare i risultati ottenuti. Il Capitolo 4 presenta i dataset che il framework ha utilizzato, Wikipedia e Twitter, caratterizzandoli per mezzo di feature statistiche. Continuando, vengono mostrati e commentati i risultati sperimentali ottenuti, visualizzandoli con molte tecniche intuitive. Il capitolo si conclude con una valutazione generale del framework e di alcune osservazioni finali per valutarne l'efficacia. Il Capitolo 5 è conclusivo del lavoro e contiene spunti su possibili lavori di ricerca futuri in quest'area.

Capitolo 2

2 Stato dell'arte

In questo capitolo verrà introdotto il contesto in cui l'analisi dei dati testuali va ad inserirsi. Dapprima verrà data una definizione di text mining, per poi analizzare differenti modelli di analisi del testo. La Sezione 2.2 illustra *Latent Semantic Analysis*, un modello algebrico di analisi, mentre la Sezione 2.3 introduce *Latent Dirichelt Allocation*, un modello probabilistico.
Di entrambi i modelli sono riportati alcuni esempi di framework sviluppati.

2.1 Analisi dei dati testuali

Con il termine *data mining*, letteralmente *estrazione di dati*, si fa riferimento al processo di estrazione, con tecniche analitiche all'avanguardia, di informazione implicita, nascosta, disseminata senza ordine in un database [1]. Il data mining costituisce la parte di modellazione del processo di *Knowledge Discovery in Databases*, ovvero l'intero processo di estrazione di informazione dai dati, che va dalla selezione e pre-processing dei dati fino all'interpretazione e valutazione del modello ottenuto [2].

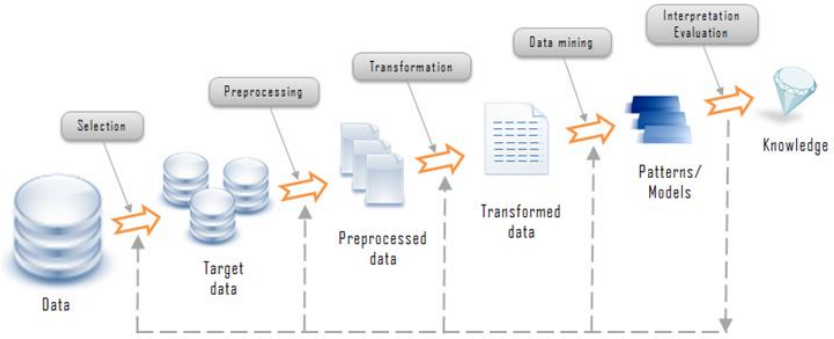


Figura 2.1: KDD [2]

In Figura 2.1 viene mostrato il processo di Knowledge Discovery; questo si suddivide in cinque fasi, le quali hanno una serie di dati in input e restituiscono in output i dati processati. Le fasi sono selezione, pre-processing, trasformazione, data mining e valutazione.
Nella fase di selezione viene creato il dataset, selezionando un sottoinsieme dei dati totali. Nella fase di pre-processing vengono eseguite diverse operazioni atte alla *pulizia* del dataset, come la rimozione di eventuali dati non rilevanti, considerati rumore. Durante la fase di trasformazione vengono selezionati gli attributi rilevanti per rappresentare il dataset; questi dipendono dall'obiettivo che si vuole raggiungere e sono spesso utilizzate tecniche di riduzione della dimensionalità. Nella fase relativa al data mining vengono ricercati nel dataset pattern significativi e informazioni nascoste. Infine, l'ultima fase è dedicata alla visualizzazione e validazione dei risultati ottenuti.

Il data mining è diventata oggigiorno una tecnica sempre più rivelante a causa della mole e complessità dei dati prodotti. Questo andamento crescente di dati può essere applicato anche ai testi. Dai social network alle enciclopedie digitali, una volta raccolti i dati, estrarre informazione utile non è semplice. Per questo motivo si utilizzano tecniche di text mining, che consiste nell'applicazione di tecniche di data mining a testi non strutturati (pagine web, e-mail, ecc.) e più in generale a qualsiasi corpus di documenti [3], allo scopo di:

- individuare i principali gruppi tematici
- classificare i documenti in categorie predefinite
- scoprire associazioni nascoste e estrarre informazione utile

Data la complessità dei dati testuali e la difficoltà di interpretare testi non strutturati, il text mining rappresenta una delle attività più impegnative del data mining, nonché un'area di grande interesse nella comunità scientifica. L'analisi di dati testuali è resa possibile grazie a numerose tecniche di elaborazione del linguaggio e metodi analitici, che coinvolgono numerosi passaggi e fasi, come l'analisi sintattica e semantica.

Nella letteratura scientifica sono stati proposti diversi modelli di analisi: *teorici* (come i modelli booleani, che rappresentano i documenti come insiemi di parole o frasi), *algebrici* (che rappresentano i documenti come vettori o matrici, come *Latent Semantic Analysis*) e *probabilistici* (come *Latent Dirichelt Allocation* e *Propabilistic Latent Semantic Analysis*, che rappresentano i documenti come probabilità delle parole).

2.2 Latent Semantic Analysis

LSA (*Latent semantic analysis*, anche conosciuta come *Latent semantic indexing*) è una tecnica di analisi semantica utilizzata nel natural language processing che consente di approfondire la conoscenza del contenuto di un documento, oltre ad individuare la relazione tra i termini che lo compongono [4]. Più precisamente, dai singoli documenti vengono estrapolati i concetti rilevanti di cui trattano.

I documenti del corpus vengono rappresentati tramite la matrice documenti-termini, successivamente vengono utilizzate tecniche di decomposizione e semplificazione matriciale per ottenere una significativa riduzione delle dimensioni delle matrici di partenza in modo da meglio caratterizzare i documenti contenuti nel corpus. LSA sfrutta principalmente una tecnica di decomposizione matriciale chiamata *Singular Value Decomposition* (*SVD*)[4].

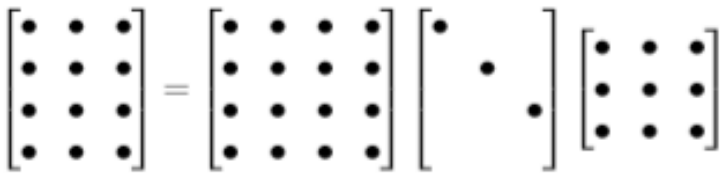


Figura 2.2: Esempio di decomposizione ottenuta tramite SVD, nel caso M > N

La necessità principale è quella di avere, in fase definizione dei cluster, un problema di dimensioni molto ridotte rispetto all'originale: data una matrice $X = M \times N$ e un intero positivo k, si vuole trovare una nuova matrice C_k di rango al più k tale che la differenza della matrice originale e della matrice approssimata sia più piccola possibile, ovvero che la norma Frobenius della matrice $X = C - Ck$ sia minima. La scelta del fattore k non è banale, poiché poche dimensioni porterebbero a una scarsa rappresentazione dei dati, mentre troppe dimensioni porterebbero a un dataset rumoroso.
Data la matrice *X*, SVD la decompone nel prodotto di tre matrici: U, S e V^T (Figura 2.2).
La matrice S rappresenta i valori singolari del dataset in analisi, uno per ogni termine. In base ai valori singolari, è possibile approssimare la matrice mantenendo solo i termini corrispondenti ai più significativi valori singolari e ignorando le altre dimensioni in S.

Alcuni tool che utilizzano il modello LSA per l'analisi testuale sono PASTA e PARAFAC.

PASTA è un motore distribuito di analisi dei dati, il cui obiettivo è quello di effettuare l'analisi di enormi set di dati testuali senza l'intervento di esperti o di analisti di dati [5]. Il framework imposta automaticamente i parametri e regola gli algoritmi di clustering del testo estraendo le conoscenze e informazioni utili per conto dell'utente finale, ottenendo risultati di qualità ottimali. PASTA è quindi in grado di suggerire all'utente lo schema di ponderazione ottimale e le funzioni di riduzione per il set di dati in analisi, insieme ai parametri per la configurazione di clustering più idonea per descrivere al meglio il dataset fornito.
L'attuale implementazione di PASTA funziona su Apache Spark, un framework di calcolo distribuito all'avanguardia.

Altre applicazioni basate sul modello LSA sono sviluppate in supporto al *cross-language information retrival*, ad esempio il framework PARAFAC [6]. Landauer e Littman (1990) furono i primi a descrivere tecniche di analisi testuali multi-lingua automatiche. Con il crescente sviluppo delle moderne applicazioni, i contenuti presenti sul web sono infatti diventati sempre più numerosi e diversi, disponibili in molte lingue. Obiettivo del framework è di rendere completamente automatico il recupero di documenti in un'altra lingua rispetto a quella utilizzata per la query. Questo viene realizzato costruendo automaticamente uno spazio semantico multilingua con LSA, nel quale sono rappresentati i termini di entrambe le lingue. Parole che hanno un significato coerente tra loro (ad esempio, *Libya* e *Libye*) sono rappresentate nello spazio in maniera identica, allo stesso modo parole che sono frequentemente associate ad un'altra (*not* e *pas*, rispettivamente in inglese e francese) avranno rappresentazioni simili.

2.2.1 Preprocessing e pesatura

Per procedere alla pesatura dei dati occorre una prima fase di preparazione dei dati. Prima di tutto viene effettuata l'operazione di **splitting**: a seconda delle necessità, i documenti possono essere analizzati nella loro interezza o singoli paragrafi. La seconda fase è quella di **tokenizzazione**, che consiste nella divisione di sequenze testuali in unità minime, dette token: parole, date, numeri e sigle possono essere token, indipendentemente dalla loro complessità. Dopo averlo ripulito e segmentato in token, il testo viene rielaborato attraverso una serie di tecniche per rimuovere elementi inutili e ridondanti.

• **Normalizzazione**: tramite questa operazione, ad esempio, le lettere maiuscole vengono sostituite con lettere minuscole in modo da uniformare i token estratti, eliminando possibili fonti di duplicazione.

• **Eliminazione stopwords**: consiste nell'eliminazione di quelle parole che sono utili per comporre una frase di senso compiuto ma che, prese da sole, non danno alcuna informazione (es. preposizioni e articoli). Questi token vengono anche detti *parole vuote* o *stopwords* e vengono rimosse poiché non danno alcuna informazione aggiuntiva.

• **Stemming**: le parole vengono ridotte alla loro "radice" (stem), eliminando, ad esempio, suffissi e prefissi. Questo permette di ridurre notevolmente il dizionario.

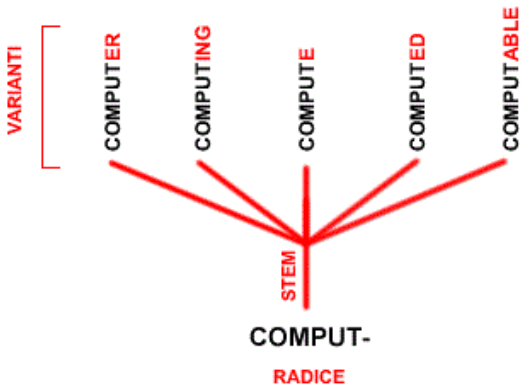


Figura 2.3: Esempio di stemming

Per questa fase preliminare sono spesso utilizzate delle *espressioni regolari* (i.e. RegEx, regular expression), ovvero delle nozioni algebriche che descrivono dei pattern di stringhe. Tali funzioni sono utilizzate per filtrare e confrontare stringhe testuali tra loro, ad esempio in maniera semplice si può sostituire le lettere maiuscole con quelle minuscole.

Dopo la fase di preparazione dei dati, i documenti sono pronti per essere analizzati. La rappresentazione *Bag of Words*[1] (BoW) è una tecnica molto utilizzata nel natural language processing per descrivere i testi senza considerare l'ordine delle parole e la grammatica dei termini, ma analizza i documenti semplicemente come contenitori (come una Borsa) di parole.
La BoW rappresenta quindi i documenti come array contenenti le occorrenze delle singole parole.

[1] Letteralmente, borsa di parole

Oltre ad essa, il modello calcola degli indici statistici (*feature*) in modo tale da caratterizzare il dataset [7]:

- *Numero di documenti* (N): numero di documenti nel corpus.

- *Numero di termini* (M): numero di parole nel corpus, considerando le ripetizioni.

- *Massima frequenza e minima frequenza*: la più grande e la più piccola frequenza di una parola all'interno del corpus.

- *Frequenza media*: la frequenza media tra le occorrenze di tutte le parole all'interno del corpus.

- *Dizionario*(D): numero di parole nel corpus, senza considerare le ripetizioni.

- *Percentuale Hapax*: il rapporto tra il numero di parole con una sola occorrenza (hapax) e la cardinalità del dizionario.

- *Type-Token Rapport* (TTR): è il rapporto tra la cardinalità del dizionario e il numero totale di termini. Rappresenta la variazione del vocabolario all'interno di un testo.

- *Coefficiente di Guiraud*: il rapporto tra la cardinalità del dizionario e la radice quadrata del numero di termini, al fine di evidenziare la ricchezza lessicale del corpus.

Al fine di identificare correttamente l'argomento di un documento ed aiutare quindi il processo di clustering, i dati vengono trasformati tramite un processo di pesatura. Ogni singolo documento del corpus viene considerato come l'insieme dei suoi token e delle rispettive occorrenze.

Ad esempio, se consideriamo la frase:

"Il Politecnico di Torino è una prestigiosa università"

l'array risultante dopo il processo di pre-processing sarà:

politecnico	torino	prestigiosa	università
1	1	1	1

I pesi misurano la rilevanza delle parole all'interno del corpus e sono dati dal prodotto di un peso locale e di uno globale [8]. Per peso locale si intende il peso di una parola in un singolo documento. Se, ad esempio, si considera il peso *Term Frequency* e la parola "*università*" compare una volta all'interno di un testo, allora al termine sarà assegnato un punteggio pari a uno. Il concetto di peso globale è molto simile a quello di peso locale, in quanto viene semplicemente esteso da un singolo documento all'intero corpus. Viene così individuata la ricorrenza di una parola rispetto all'intero dataset di documenti, la *Document Frequency*. I pesi vengono poi memorizzati in una matrice documenti-termini dove le righe sono associate ai documenti e le colonne alle parole: data la matrice $X = M \; x \; N$, $X_{m,n}$ corrisponde al peso assegnato al termine nesimo del documento m. Diverse combinazioni di pesi locali e globali permettono di ottenere diverse funzioni di pesatura e a seconda della struttura intrinseca del corpus possiamo osservare differenti comportamenti e scegliere come migliore una combinazione piuttosto che un'altra.

I pesi locali utilizzati nel framework PASTA sono TF (*Term Frequency*), che si basa sul concetto semplicissimo di ricorrenza di una parola *i* all'interno di un testo *j*, e LogTF (*Logarithmic Term Frequency*), che sfrutta il logaritmo per dare meno peso alle parole che compaiono molte volte all'interno del documento. Il peso globale è IDF (*Inverse Document Frequency*) che prende in considerazione l'occorrenza di una parola all'interno di tutti i documenti del corpus, e *Entropy*.

Peso locale	$TF = tf_{ij}$ $LogTF \; = \; log_2(tf_{ij} \; + \; 1)$
Peso globale	$IDF = log(\frac{\lvert D \rvert}{df_i})$ $Entropy = 1 + \frac{\sum_j p(i,j) \log p(i,j)}{\log(M)}$

Tabella 2.1: Tabella dei pesi utilizzati in PASTA

2.2.2 Clustering

Per ridurre la dimensionalità della matrice documenti-termini e focalizzare il calcolo solo sui concetti più rilevanti dei documenti, è necessaria una trasformazione dei dati. Una volta che la riduzione è stata fatta, ad esempio applicando *Singular Value Decomposition,* possono essere utilizzate diverse tecniche di clusterizzazione. PASTA, ad esempio, clusterizza i documenti del corpus utilizzando l'algoritmo K-Means [9]. Questo algoritmo, data una collezione di elementi, permette l'identificazione di K gruppi (o cluster). Per il K-means, un cluster è rappresentato dal centro di tutti i punti che lo costituiscono. Nel nostro caso, questi valori sono quelli ottenuti dalla riduzione della matrice documenti-termini e rappresentano i concetti. Il centro del cluster prende il nome di centroide ed è la media aritmetica dei punti.
Inizialmente l'algoritmo prende in maniera totalmente casuale alcuni punti come centroidi e ogni punto è assegnato al centroide più vicino. Vengono poi calcolati i nuovi centroidi dai cluster appena ottenuti e viene ripetuto il processo di assegnazione, fino a quando i centroidi non cambiano.
Il K-Means è un algoritmo che lavora bene essendo computazionalmente veloce, tuttavia il valore di K è un parametro settato a priori e sceglierlo correttamente è uno dei punti critici di questa tecnica.

2.3 Latent Dirichelt Allocation

Un approccio completamente diverso dai modelli di tipo algebrico è quello che prevede l'utilizzo di modelli probabilistici. Questa tecnica rappresenta i documenti testuali non più come matrici documenti-termini, ma come probabilità di parole e mira a scoprire informazioni tematiche all'interno di collezioni di documenti. Gli algoritmi probabilistici si basano su metodi statistici che analizzano i testi e le loro parole al fine di scoprire gli argomenti che affrontano e a quali documenti sono correlati. Queste tecniche sono in grado di operare senza una precedente conoscenza dei dati e quindi lavorano su dati senza etichette. Gli argomenti (*topic*) sono definiti come distribuzioni su un vocabolario fisso, mentre i documenti come distribuzioni di più argomenti diversi. Tuttavia, come la maggior parte degli algoritmi di analisi testuale, anche quelli probabilistici richiedono che il numero di topic sia settato a priori. Trovare il valore ottimale di configurazione non è facile e rappresenta un problema aperto nella comunità scientifica.

Latent Dirichelt Allocation (LDA) è uno dei più famosi e più utilizzati modelli probabilistici generativi non supervisionati [10]. Con lo scopo di migliorare gli approcci esistenti di analisi del testo, D. Blei, A. Ng e M. Jordan hanno creato LDA, un modello in grado di descrivere ampie raccolte di dati senza ricorrere a riduzioni di dimensionalità dei dati come nel caso dell'LSA. I documenti e le parole, rappresentati attraverso la *Bag of Words* come vettori sparsi di occorrenze, devono essere rappresentati come distribuzioni probabilistiche.

Le probabilità utilizzate nel modello sono le seguenti:

- Poisson(λ), rappresenta la distribuzione delle lunghezze dei documenti.

- θ, rappresenta le distribuzioni document-topics, ossia la probabilità che un documento appartenga ad uno specifico topic k.

- φ, rappresenta le distribuzioni topic-parole. Rappresenta la proprietà di una parola di appartenere allo specifico topic.

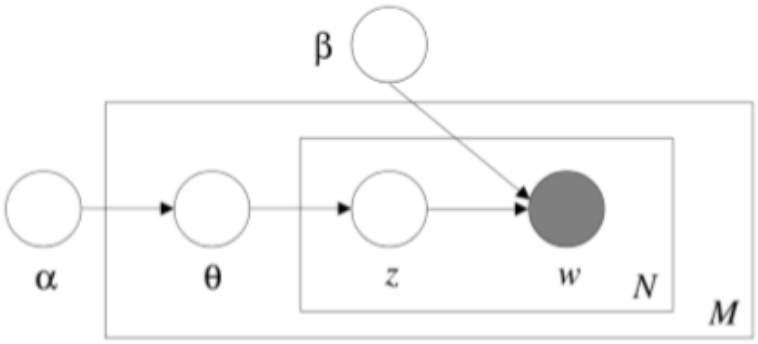


Figura 2.4: Rappresentazione grafica del modello LDA

Come mostra la Figura 2.4, il modello LDA può essere rappresentato da uno schema a tre livelli: il corpus, i documenti e i termini. Nel livello più interno, corrispondente alle parole*, z* e *w* sono campionati per ogni *n,* in modo tale che ogni documento possa essere descritto come composizione di argomenti multipli.

I parametri che l'algoritmo LDA deve settare sono: α, β e K:

• Il parametro α rappresenta la concentrazione delle distribuzioni dei documenti rispetto agli argomenti (θ). Ciò significa che valori bassi di α creeranno documenti che probabilmente contengono una combinazione di pochi argomenti, mentre valori alti porranno più peso sull'avere documenti composti da molti argomenti dominanti.

• Il parametro β descrive la concentrazione delle distribuzioni degli argomenti rispetto ai termini. Ciò significa che valori bassi di β probabilmente produrranno argomenti che sono ben descritti solo con poche parole, mentre valori alti creeranno argomenti composti da una miscela della maggior parte delle parole.

• K, il numero di topic. In letteratura sono state esplorate e proposte diverse soluzioni per trovare il valore K più adatto. Questo numero ha una grande influenza sui risultati del processo di clustering e deve quindi essere impostato con attenzione. Valori di K troppo bassi farebbero diventare LDA poco rilevante per poter identificare i cluster appropriati, mentre valori di K troppo grandi porteranno a un modello molto complesso, difficile da visualizzare e da validare.

Un framework che utilizza il modello LDA per l'analisi del testo è ToPIC (*Tuning of Parameters for Inference of Concepts)*, un motore distribuito di configurazione automatica il cui scopo è quello di clusterizzare collezioni di documenti, formando gruppi coesi e ben separati di testi correlati tra loro, a seconda del loro contenuto tematico [11]. Il framework cerca di determinare automaticamente il numero più adatto di topic K per l'algoritmo LDA, utilizzando un nuovo indice di valutazione chiamato ToPIC Similarity.

Utilizzando questo metodo, la fase di decomposizione dell'LSA (SVD) non è più necessaria poiché LDA riduce già i documenti e può essere quindi utilizzata come tecnica di riduzione.
Le strategie offerte da ToPIC supportano l'utente nel processo di analisi testuale.

2.3.1 Preprocessing e pesatura

Prima di applicare l'algoritmo probabilistico LDA al set di dati, il pre-processing dei dati che è stato eseguito nei framework precedenti, prima della riduzione matriciale, è eseguito anche in questo nuovo framework. A partire dai dati grezzi, i documenti sono splittati e le parole trasformate in token. Vengono quindi rimosse dal corpus le stopword e si continua con le fasi di stemming e normalizzazione.
Si procede poi ad una ulteriore tecnica di riduzione della dimensionalità che coincide con la rimozione degli hapax. Gli hapax sono quei termini che all'interno del corpus appaiono una sola volta e che quindi nella *Bag of Words* avranno una sola occorrenza. Rimuovere gli hapax permette di ridurre la varietà del dizionario e rendere il modello più affidabile.

ToPIC comprende una serie di funzioni di pesatura, basate su diverse combinazioni di pesi globali e locali. Il framework utilizza tre pesi locali, *Term-Frequency* (TF), *Logarithmic Term Frequency* (LogTF) e *Unitary* (Boolean), e tre pesi globali, *Inverse Document Frequency* (IDF), *Entropy* (Entropy) e *Term-Frequency* (TF_{glob}).

Nella tabella sono rappresentati i pesi locali e globali utilizzati, per il documento *j* e la parola *i*.

Peso locale	$TF = tf_{ij}$ $LogTF = log_2(tf_{ij} + 1)$ $Boolean = \begin{cases} 1, & if\ tf_{ij} > 0 \\ 0, & if\ tf_{ij} < 0 \end{cases}$
Peso globale	$IDF = log(\frac{\lvert D \rvert}{df_i})$ $TF_{glob} = tf_i$ $Entropy = 1 + \frac{\sum_j p(i,j) \log p(i,j)}{\log(M)}$

Tabella 2.2: Tabella dei pesi utilizzati in ToPIC

Data la matrice dei pesi X, il modello LDA è calcolato per un numero specifico di argomenti K, al fine di calcolare l'inferenza della distribuzione delle variabili latenti per il dato corpus. Il modello probabilistico ottenuto sarà analizzato con *ToPIC-Similarity*, algoritmo per la configurazione ottimale del modello LDA in modo automatico, e successivamente il flusso analitico può essere reinserito nell'algoritmo per produrre un nuovo modello con una nuova configurazione algoritmica.
Questo processo viene ripetuto finché non si ottiene un buon compromesso tra la qualità dei risultati ottenuti e il tempo di esecuzione.

ToPIC-Similarity non considera solo gli indici quantitativi probabilistici dell'intero modello, ma i contenuti e la descrizione degli argomenti; valuta come gli argomenti sono semanticamente diversi/simili e quindi sceglie le configurazioni appropriate.

Poiché φ modella le distribuzioni argomento-termini, è disponibile una descrizione degli argomenti in base al loro contenuto. Avendo quindi i termini e gli argomenti a cui questi fanno parte, è possibile analizzare la loro somiglianza, e quindi scegliere K in maniera tale da massimizzare la differenza tra loro.
Dato un numero minimo e massimo di argomenti impostati dall'analista (cioè [Kmin, Kmax]) viene generato un nuovo modello LDA per ogni K.
Per ognuno di questi processi, ToPIC-Similarity calcola tre passaggi intermedi:

- *topic characterization*: ciascun argomento *t* viene descritto utilizzando *n* parole rappresentative.

- *similarity computation*: è necessario valutare la somiglianza tra topics, in maniera tale da minimizzarla. La similutidine viene calcolata attraverso la cosine-similarity, una tecnica molto utilizzata nel data mining e in particolare nell'analisi di documenti testuali. La cosine-similarity è una tecnica euristica che calcola il coseno tra due vettori: nel nostro caso, il contenuto di essi è la frequenza, positiva o uguale a zero, dei termini che ricorrono nel testo. La similarità è calcolata per tutte le combinazioni di topic, costruendo una matrice K x K dove ogni cella rappresenta la similarità tra il topic in riga *i* e quello in colonna *j*.

- *K identification*: per trovare le configurazioni di clustering ottimali da proporre all'analista, vengono ripetuti i passaggi precedenti per diversi valori di K e poi confrontati.

I risultati vengono infine visualizzati e validati con una serie di indici di qualità, come Perplexity e metriche di clustering (e.g. Silhouette e Entropy), e tecniche di rappresentazione per valutare la bontà della modellazione, sia in termini di raggruppamento dei documenti nei cluster, sia in termini di caratterizzazione dei singoli argomenti.
I risultati sperimentali ottenuti dall'applicazione di LDA mostrano l'efficacia e l'efficienza di ToPIC nel raggruppare collezioni di documenti in gruppi tematici coesi, in base al loro contenuto. Per questo motivo, visti gli esiti positivi del modello in analisi, ci si è focalizzati nell'analizzare un nuovo approccio basato sul modello probabilistico PLSA, che come l'LDA può essere utilizzato per analizzare i testi e le loro parole al fine di scoprire gli argomenti che affrontano e a quali documenti sono correlati. Questa tecnica è in grado di operare senza una precedente conoscenza dei dati e quindi lavora su dati senza etichette, e definisce i documenti come distribuzioni di più argomenti diversi.

Capitolo 3

3 CONCEPT

In questo capitolo viene presentato il framework CONCEPT, *CharacterizatiON of a textual Collection through the Evaluation of Probabilistic Techniques*, la sua implementazione e nuovo approccio nell'analisi dei dati testuali. L'algoritmo si basa sul modello *Probabilistic Latent Semantic Analysis* [11]. Come il nome suggerisce, questo approccio è stato largamente ispirato e influenzato dall'LSA (*Latent Semantic Analysis* – Capitolo 2), sebbene esistano delle differenze evidenti. L'attuale implementazione di CONCEPT è sviluppata in Java.
Nella Sezione 3.1 viene presentato il modello probabilistico PLSA. La Sezione 3.2 fornisce una panoramica sull'intera implementazione del sistema, dalla fase di pre-processing a quella di pesatura. Infine, nella Sezione 3.3 vengono presentate le principali tecniche – indici e grafici – per visualizzare e validare efficacemente i risultati prodotti dal processo di clustering.

3.1 Modello probabilistico

PLSA deriva da una visione statistica dell'LSA e definisce un proprio modello generativo: invece di utilizzare matrici matematiche e la *Singular Value Decomposition* per ridurre le dimensioni del problema, utilizza un modello probabilistico. Questo nuovo approccio presenta molti vantaggi grazie alla solida base statistica su cui fonda i propri principi e può quindi essere considerato un nuovo promettente metodo di apprendimento non supervisionato, con una vasta gamma di applicazioni nell'analisi testuale.

Il punto di partenza per la PLSA è un modello statistico che prende il nome di *aspect model*, proposto da Hofmann, Puzicha, & Jordan [12]. L'*aspect model* è un modello variabile latente per co-occorrenze di dati, il cui scopo è quello di costruire un modello in grado di descrivere rapidamente grandi collezioni di documenti senza ricorrere alla riduzione della dimensionalità dei dati. Data la matrice X documenti-termini, il modello viene calcolato per un numero K di argomenti, settato a priori.

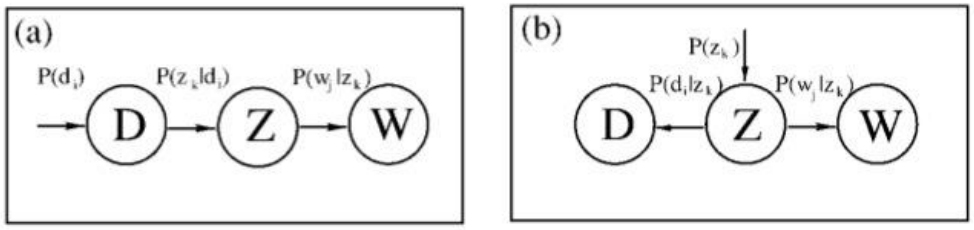


Figura 3.1: Rappresentazione grafica del modello PLSA

Come per l'LDA, gli argomenti e le parole sono descritti come distribuzioni probabilistiche dalle quali poi i documenti saranno creati.
Utilizzando lo schema in figura possiamo definire un modello generativo che si basa sulle seguenti fasi:

- Si seleziona un documento d_i con probabilità $P(d_i)$,
- Dato un documento d, si sceglie un topic z con probabilità $P(z_k|d_i)$,
- A partire dal topic z, si genera una parola w_j con probabilità $P(w_j|z_k)$.

La probabilità condizionata risultante è descritta dall'espressione:

$$P(d_i, w_j) = P(d_i)P(w_j|d_i)$$

$$P(w_j|d_i) = \sum_{k=1}^{K} P(w_j|z_k)P(z_k|d_i)$$

Come la maggior parte dei modelli variabili latenti, l'*aspect model* introduce un'assunzione di indipendenza condizionale, vale a dire che d e w sono indipendenti condizionati sullo stato della variabile latente associata z.
Questo processo di generazione di documenti deve essere ripetuto per tutti i documenti appartenenti al corpus. In altre parole, per ogni documento d nel corpus e per ogni parola w, viene scelto un argomento in base alla distribuzione degli argomenti calcolata per quel documento.

La parte centrale del modello si basa sull'algoritmo di *Expectation Maximization* (EM) per il calcolo della massima Log-likelihood [13]. EM alterna due fasi:

- una fase di attesa (*Expectation)* dove sono calcolate le probabilità a posteriori per le variabili latenti, applicando semplicemente la formula di Bayes.

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)\,P(z_l|d_i)}$$

- una fase di massimizzazione (*Maximization)*, che cerca di massimizzare la Log-likelihood associata ai dati, in cui i parametri vengono aggiornati.

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)\,P(z_k|d_i, w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i, w_m)\,P(z_k|d_i, w_m)}$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)\,P(z_k|d_i, w_j)}{n(d_i)}$$

3.2 Architettura

In Figura 3.2 è mostrata la struttura di CONCEPT che applica il modello PLSA a collezioni di testi al fine di raggruppare i documenti del corpus in base ai loro contenuti. Utilizzando questo metodo per modellare gli argomenti dei documenti di testo, la fase di riduzione (SVD) utilizzata dall'algoritmo LSA non è più necessaria, poiché PLSA riduce già i documenti. L'architettura del framework include tre componenti principali: (i) Pre-processing, (ii) Modello PLSA applicato ai documenti, (iii) Visualizzazione e validazione.

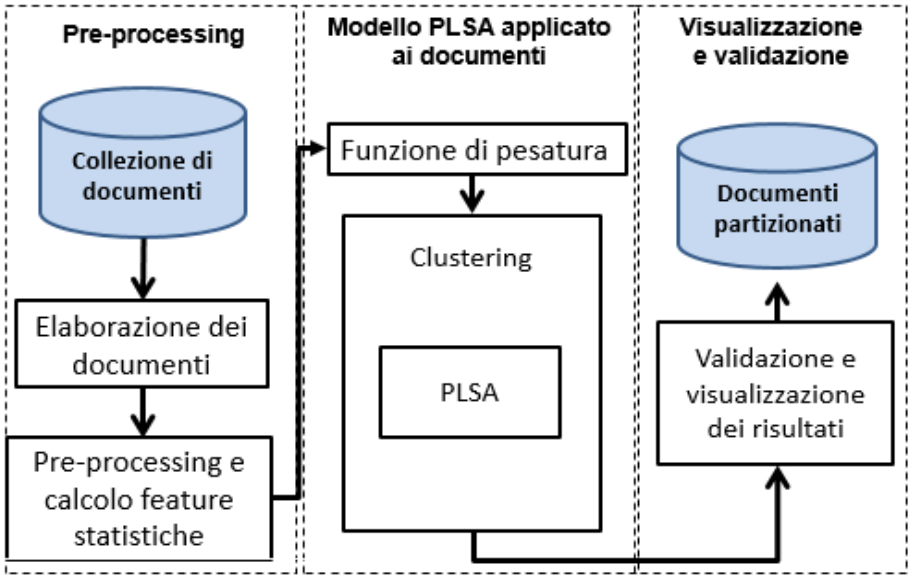


Figura 3.2: Architettura del framework CONCEPT

In particolare, nella fase di pre-processing i dati grezzi sono sottoposti ad una fase preliminare in modo da diventare conformi all'analisi. Su tali dati sono poi calcolate le feature statistiche di base che caratterizzano il corpus. Alle parole del corpus viene poi assegnato un peso che nasce dal prodotto tra un peso locale ed uno globale. Il modello PLSA viene quindi applicato al dataset, associando agli argomenti e ai termini delle distribuzioni di probabilità. Infine, i risultati prodotti vengono analizzati con diverse tecniche di visualizzazione e indici di qualità per comprenderne il significato.

3.2.1 Preprocessing e pesatura

Come presentato per gli altri framework, prima di assegnare dei pesi alle parole dei documenti, vi sono delle fasi di pre-processing essenziali per garantire dei risultati affidabili negli step successivi. I dati sono quindi manipolati secondo le seguenti fasi: *splitting*, *tokenizzazione*, *rimozione delle stopwords*, *stemming*, *normalizzazione* e *rimozione degli hapax*. Oltre agli hapax, per alcuni dataset può essere necessaria anche una fase di rimozione delle parole più frequenti: una parola che si ripete tante volte all'interno del corpus potrebbe essere non discriminante al fine della clusterizzazione. Per fare questo viene utilizzata la tecnica statistica delle 3*σ (sigma), dove σ = deviazione standard.
Dopo queste operazioni, sono calcolate le distribuzioni statistiche sull'intero corpus descritte nella Sezione 3.1. A questo punto il processo di pesatura può avere inizio. Per sottolineare l'importanza di ogni parola all'interno della collezione di documenti, si assegna un peso ad ogni termine del corpus. Le tecniche di pesatura utilizzate derivano dal prodotto di un peso locale, riferito allo specifico documento, e globale, relativo all'intero corpus.

Nel nostro studio utilizziamo le seguenti combinazioni di pesi:

- TF-IDF
- Boolean-TF_{glob}
- Boolean-Entropy
- Boolean-IDF

Le funzioni di pesatura utilizzate sono due tecniche di pesi locali e tre tecniche di pesi globali. TF, *Term Frequency*, è pari alla frequenza, in termini di numero di occorrenze, del termine *i* nel documento *j*. Boolean corrisponde al caso binario della *Term Frequency*: se la parola compare almeno una volta nel testo, si assegna peso pari a uno, viceversa zero.

IDF, *Inverse Document Frequency*, calcola il rapporto tra il numero di documenti del corpus e il numero di documenti in cui il termine considerato appare. TF_{glob} corrisponde al peso *Term Frequency* calcolato su tutti i documenti. Infine per l'Entropy si calcola la probabilità condizionata *p(i,j),* data dal rapporto tra *Term Frequency* locale e globale, a cui poi si somma uno.

Peso locale	$TF = tf_{ij}$ $Boolean = \begin{cases} 1, & if\ tf_{ij} > 0 \\ 0, & if\ tf_{ij} < 0 \end{cases}$
Peso globale	$IDF = log(\frac{\lvert D \rvert}{df_i})$ $TF_{glob} = tf_i$ $Entropy = 1 + \frac{\sum_j p(i,j) \log p(i,j)}{\log(M)}$

Tabella 3.1: Tabella dei pesi utilizzati in CONCEPT

3.3 Visualizzare e validare i risultati

Data la complessità delle operazioni di text-mining, la valutazione dei risultati prodotti dal modello probabilistico è difficile. A tale scopo possono essere utilizzati diversi indici teorici e metodi di visualizzazione per valutare la qualità del processo di clustering e quindi visualizzare intuitivamente il partizionamento ottimale. Per valutare la bontà della clusterizzazione mediante PLSA, è possibile utilizzare delle metriche di valutazioni quantitative, *Log-likelihood* (misura la qualità del modello probabilistico), *Silhouette* (valuta la coesione e separazione tra clusters) e *Rand Index* (valuta la similarità tra i cluster generati), e di visualizzazione, tra cui *grafici t-SNE* e rappresentazioni grafiche con *word cloud*, *scatter plot*, *stacked-bar* e *grafi.*

3.3.1 Log-likelihood

La Log-likelihood (verosimiglianza) è una misura ampiamente utilizzata per valutare la qualità di modelli probabilistici, equivalente all'inverso della perplexity [14].
Osservato un determinato campione $x_1, x_2 \ldots x_n$ estratto da una popolazione X, la cui distribuzione dipende da un parametro θ, la funzione di verosimiglianza L(θ) indica la probabilità di osservare il campione al variare del parametro θ. Il metodo della massima verosimiglianza (EM) consiste nel prendere il valore di θ che è il più verosimile degli altri, cioè che massimizza la funzione L(θ). Per convenienza si utilizza massimizzare il logaritmo della verosimiglianza, che prende il nome di Log-likelihood.

$$Log\text{-}likelihood(L) = \sum_m^M log\ p(w_m)$$

3.3.2 Silhouette

La Silhouette è una misura molto utilizzata nel data mining per valutare la bontà del risultato di un algoritmo di clustering [15]. In particolare, per ogni punto viene calcolata la distanza media dagli altri punti appartenenti allo stesso cluster (intra-cluster) e la distanza media da tutti i punti del cluster più vicino (nearest-cluster).
Dette rispettivamente a_i e b_i queste due misure per il punto i, la Silhouette per quel punto risulta essere:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

che può anche essere riscritta come

$$s_i = \begin{cases} 1 - \dfrac{a_i}{b_i} & se\ a_i \leq b_i \\ \dfrac{b_i}{a_i} - 1 & se\ a_i > b_i \end{cases}$$

La misura Silhouette varia da -1 a 1; valori vicini a 0 indicano cluster sovrapposti, valori negativi generalmente indicano che un campione è stato assegnato al cluster sbagliato. Nel nostro scenario, l'indice Silhouette viene calcolato utilizzando il topic più probabile di un documento come etichetta di clustering e tutte le probabilità del documento associate a ciascun topic come coordinate dei punti stessi.

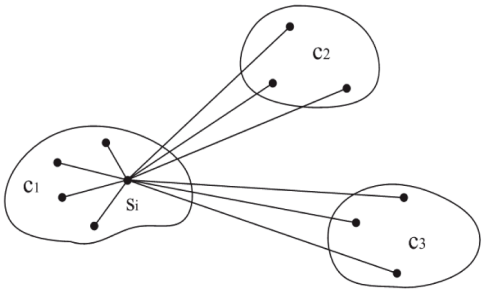


Figura 3.3: Costruzione della Silhouette

Sulla definizione di Silhouette sono basati due indici [16]:

- **Average Silhouette Index** (ASI): corrisponde alla silhouette mediata per il numero totale di elementi clusterizzati.

$$ASI = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in L_k} s_i$$

- **Global Silhouette Index** (GSI): indice globale che normalizza la somma delle silhouette di ciascun cluster per la cardinalità di ogni cluster.

$$GSI = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|L_k|} \sum_{i \in L_k} s_i$$

Dove L_k è il set di elementi appartenenti al cluster k=1,2…K; $|L_k|$ è la cardinalità del cluster e N il numero totale di elementi clusterizzati.

3.3.3 Rand Index

Applicando l'algoritmo di clustering per diversi valori di K si ottengono molteplici risultati. Un'informazione molto rilevante che può essere utilizzata per valutare e validare i risultati è relativa alla similarità che esiste tra i cluster generati. Per quantificare questo valore si è deciso di sfruttare l'indice statistico *Rand Index* [17].
In particolare, questo indice permette di confrontare l'assegnazione dei documenti ai cluster all'interno di ogni soluzione, identificando gli elementi veri positivi, veri negativi, falsi positivi e falsi negativi.
Per comprendere il significato dell'indice consideriamo $S = s_1,...,s_n$ l'insieme degli n elementi da clusterizzare, e $X = x_1,...,x_n$ e $Y = y_1,...,y_n$ due diverse partizioni di S da confrontare.
Si definiscono quindi:

- TP (veri positivi): il numero di coppie di elementi di S che compaiono nello stesso cluster sia in X che in Y.
- TN (veri negativi): il numero di coppie di elementi di S che compaiono in cluster diversi in X e in Y.
- FP (falsi positivi): il numero di coppie di elementi di S che compaiono nello stesso cluster in X ma che vengono invece collocati in cluster diversi in Y.
- FN (falsi positivi): il numero di coppie di elementi di S che compaiono in cluster diversi in X ma che vengono invece collocati in cluster uguali in Y.

Il Rand Index misura la percentuale di decisioni corrette, secondo la formula:

$$Rand\ Index = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\binom{n}{2}}$$

Il valore *Rand Index* identifica quindi la similarità che esiste tra due diverse soluzioni dato lo stesso dataset iniziale. L'indice varia tra 0 e 1, indicando con zero due soluzioni completamente differenti e con uno due soluzioni identiche. Applicando questo indice a soluzioni con numero di cluster diverso non sarà mai possibile ottenere 1, tuttavia questo indice risulta essere un buon approccio per verificare l'aggregazione e similarità tra due diverse soluzioni. L'indice viene calcolato per ogni coppia di esperimenti possibili.

3.3.4 t-SNE

Visualizzare i risultati prodotti dal processo di clustering non è banale, soprattutto per dati di grandi dimensioni quali sono i documenti.
Per questo motivo, si è scelto di utilizzare il *t-Distributed Stochastic Neighbor Embedding* (t-SNE), una tecnica in grado di visualizzare i dati ad alta dimensione su uno spazio bi o tridimensionale [18].
t-SNE è un algoritmo non lineare per la riduzione della dimensionalità, in grado di catturare molto bene la struttura locale dei dati, mentre allo stesso tempo rivela la struttura globale come la presenza di cluster di diverse dimensioni. Questa funzione consente di rappresentare punti di dati simili vicini tra loro e, allo stesso tempo, dati diversi lontani tra loro. L'algoritmo converte le distanze euclidee tra i punti dati in probabilità condizionali che rappresentano le somiglianze. Questa tecnica si articola in due fasi principali. Nella prima fase viene costruita una distribuzione di probabilità che ad ogni coppia di punti nello spazio originale ad alta dimensionalità associa un valore di probabilità elevato se i due punti sono simili, basso se sono dissimili. Quindi viene definita una seconda distribuzione di probabilità, analoga alla prima, nello spazio a dimensione ridotta. L'algoritmo quindi minimizza la divergenza di Kullback-Leibler delle le due distribuzioni tramite discesa del gradiente, riorganizzando i punti nello spazio a dimensione ridotta.

In Figura 3.4 è mostrato un esempio di visualizzazione del dataset con t-SNE. È possibile vedere in maniera intuitiva che i punti rappresentanti i documenti sono raggruppati in diversi cluster, in base alle loro caratteristiche semantiche. I diversi colori dei cluster indicano l'assegnazione del documento ad un argomento specifico.

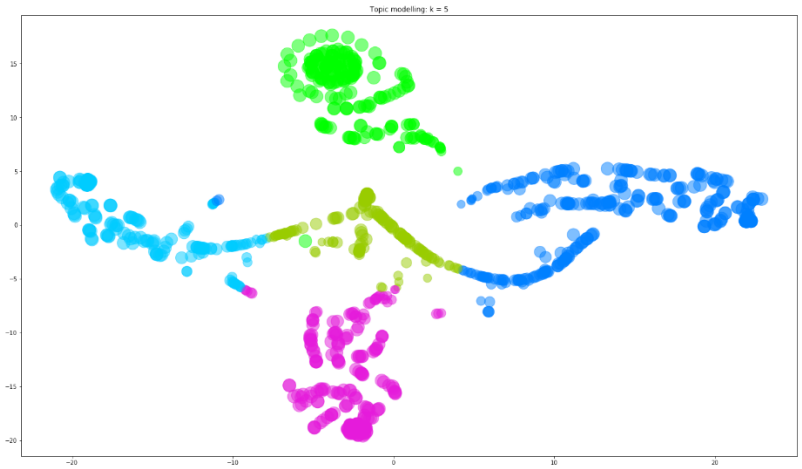


Figura 3.4: t-SNE, D1, TF-IDF, K=5

3.3.5 Word Cloud

Per visualizzare il contenuto di ciascun topic, si utilizza la tecnica delle word cloud, che rappresentano i termini rappresentativi che secondo il modello PLSA ottenuto descrivono con più probabilità l'argomento. Le *word cloud* rappresentano le matrici argomento-termini. Il confronto tra word cloud ottenute da diversi modelli è lasciato all'occhio umano. In questo modo è possibile visualizzare in modo semplice e intuitivo se i risultati della classificazione sono buoni o se questa non ha dato risultati accettabili, inserendo nella stessa word cloud parole appartenenti a topic diversi.

3.3.6 Stacked bar

Lo stacked-bar è un particolare tipo di istogramma simile a quello tradizionale. La barra che descrive la grandezza misurata sull'asse delle ordinate, nel nostro caso i valori di probabilità tra 0 e 1, è suddivisa in caselle la cui area esprime un valore parziale della grandezza. Questa rappresentazione è stata utilizzata per visualizzare le probabilità associate ai documenti di appartenete a ciascun argomento K settato. Essendo i dataset composti da molti documenti, lo stacked-bar risultante ci fa capire l'andamento delle probabilità assegnate, che sono state ordinate per cluster (ogni cluster è associato ad un colore).
Nella figura 3.5 è illustrato un esempio di grafico. In particolare, la figura 3.6 mostra uno zoom della rappresentazione per farne comprendere la costruzione.

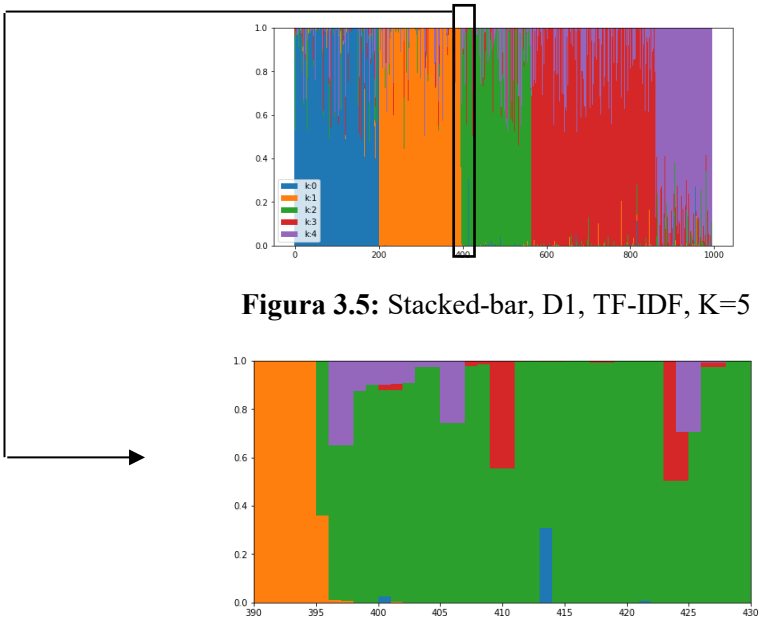


Figura 3.5: Stacked-bar, D1, TF-IDF, K=5

Figura 3.6: Zoom stacked-bar

3.3.7 Scatter plot

Lo scatter plot, o *grafico di dispersione*, è un tipo di grafico in cui due variabili di un set di dati sono riportate su uno spazio cartesiano. I dati sono visualizzati tramite una collezione di punti ciascuno, con una posizione sull'asse orizzontale determinata da una variabile e sull'asse verticale determinata dall'altra. Nel nostro caso il grafico a dispersione è utilizzato per visualizzare l'andamento delle probabilità associate a ciascun documento, al variare del numero K di argomenti assegnati. È infatti possibile colorare i punti associando a questi una label che specifica la classe di appartenenza del punto. In Figura 3.7 è mostrato l'esempio di uno scatter plot per K = 3. Sull'asse x sono rappresentati i documenti, mentre sulle y le probabilità (valore massimo 1). Ad ogni argomento è infine associato un colore diverso.

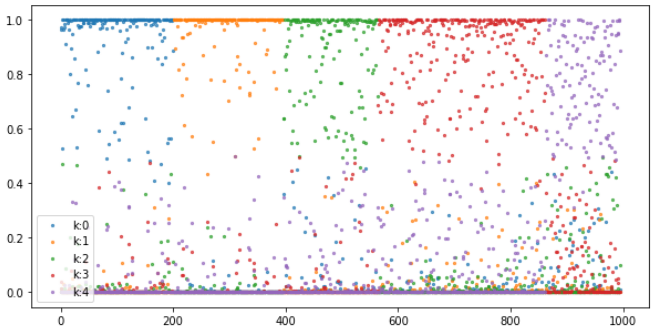


Figura 3.7: Scatter plot, D1, TF-IDF, K=5

3.3.8 Grafo

In matematica un grafo è una configurazione formata da nodi interconnessi da archi. Nel nostro caso, si è scelto di utilizzare un grafo per visualizzare in maniera intuitiva le top-parole appartenenti a ciascun argomento, in maniera tale da poter sfruttare le numerose applicazioni del grafo, calcolandone la densità. Nella teoria dei grafi, la densità di un grafo $G = (V, E)$ è definita come il rapporto tra il numero di archi di un dato grafo e il numero totale di archi che il grafo potrebbe avere. Poiché il numero massimo di archi dipende dai nodi, un grafo si dice denso se:

$$|E| \approx |V|$$

Per l'implementazione del grafo si è fatto riferimento alla libreria python-igraph[2] di Python 2.7, la quale supporta anche il calcolo della densità.
Il metodo ritorna un valore compreso tra 0 e 1.
Più sono numerosi i nodi direttamente collegati fra loro, più un grafo è denso e il valore vicino a 1; se il grafo è isolato è pari a 0 [19].
In Figura 3.8 è presentata un esempio di visualizzazione del grafo. Ai nodi, che corrispondono agli argomenti, sono collegate le parole rappresentative per ciascun topic. I topic sono i nodi gialli; le parole ad essi associati i nodi rossi. In questo modo è possibile calcolare la densità del grafo come misura del valore di similarità tra topic. Se questi condividono molte parole, i cluster non sono ben separati e il valore di densità è alto; viceversa, più i grafi risultano isolati, più la clusterizzazione è efficiente.

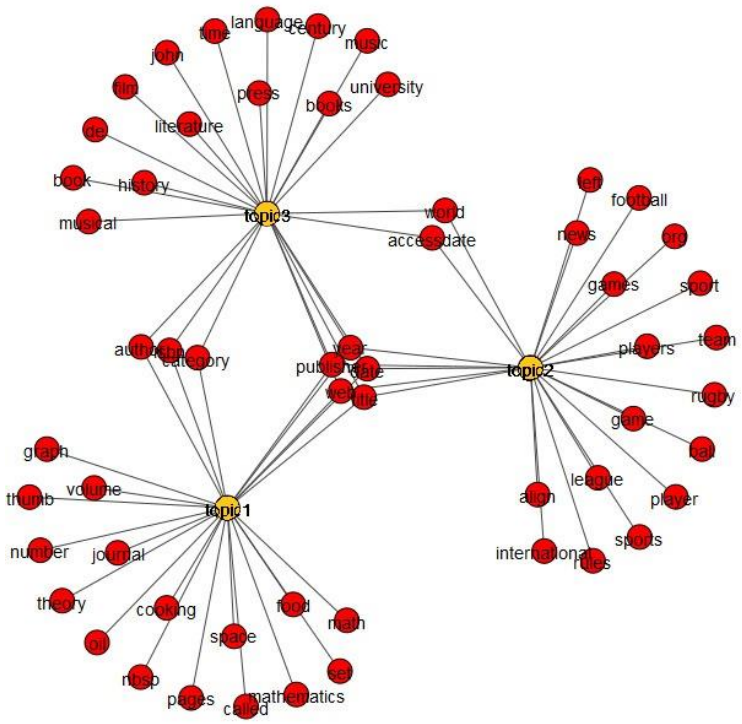


Figura 3.8: Grafo, D1, TF-IDF, K=3.

[2] http://igraph.org/python/

Capitolo 4

4 Risultati sperimentali

In questo capitolo sono riportati i risultati sperimentali ottenuti nell'applicazione della PLSA. La sezione 4.1 descrive i dataset utilizzati e per ciascuno riporta le feature statistiche. La sezione 4.2 descrive nel dettaglio i risultati prodotti sul dataset D1, insieme alle differenti tecniche per la visualizzazione e validazione. La sezione 4.3 illustra i risultati prodotti sugli altri dataset.

4.1 Scelta del dataset

La scelta dei dataset per valutare in maniera efficace le funzionalità dell'algoritmo considerato è stata fatta prendendo in considerazione alcune importanti proprietà. Il dataset deve avere delle caratteristiche strutturali tali che la lunghezza dei documenti in esso sia omogenea, ma i contenuti e autori siano eterogenei tra loro, in modo da garantire una buona varietà del dizionario. Queste caratteristiche consentono ai risultati di essere comparabili e generici, evitando l'*overfitting* [20]. Per overfitting si intende un errore di modellazione che si verifica quando il modello si adatta troppo a un insieme limitato di dati e potrebbe non riuscire a modellarne di nuovi.
Per ciascun dataset vengono riportate le feature specifiche per l'analisi testuale prima e dopo della rimozione degli hapax. Gli hapax sono quelle parole che appaiono nel corpus una sola volta e che posso essere quindi considerate "rumore", la cui rimozione permette di ottenere un modello più preciso e affidabile. Oltre agli hapax, si riportano anche le statistiche dopo la rimozione delle parole molto frequenti: una parola che si ripete molte volte all'interno del corpus potrebbe essere non discriminante al fine della clusterizzazione. Per fare questo viene utilizzata la tecnica delle $3*\sigma$ (sigma), dove σ = *deviazione standard* [21]. In statistica, sigma, o deviazione standard, corrisponde ad una misura della dispersione dei dati attorno al valore medio.

Come si osserva dalla Figura 4.1, togliendo le parole più frequenti, da 3*σ in poi sulla curva, si riduce la distribuzione dello 0.27%.

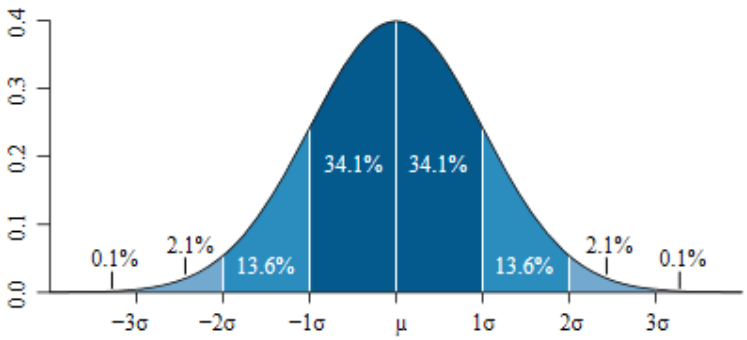


Figura 4.1: Rappresentazione grafica del metodo 3-sigma

4.1.1 Dataset Wikipedia

Per soddisfare tutte queste caratteristiche si è deciso di estrarre alcuni articoli dalla piattaforma web Wikipedia [3] . In generale infatti gli articoli enciclopedici, messi a disposizione da Wikipedia gratuitamente, hanno tutte le proprietà descritte precedentemente: la lunghezza dei documenti estratti è più o meno sempre comparabile, ogni articolo tratta un argomento specifico e più autori posso contribuirne alla stesura.
Per creare i dataset D1 e D2 sono state selezionate alcune collezioni di documenti appartenenti a categorie che potessero essere sufficientemente lontane come argomentazioni in modo da poter essere facilmente clusterizzabili.

Il primo dataset è costruito estraendo circa 200 documenti per cinque macro-categorie:

- *Cooking*
- *Literature*
- *Mathematics*
- *Music*
- *Sports*

3 http://www.wikipedia.en

Il secondo dataset è invece costituito da 10 categorie e 250 documenti per ognuna di esse:

- *Astronomy*
- *Cooking*
- *Geography*
- *History*
- *Literature*
- *Mathematics*
- *Music*
- *Politics*
- *Religion*
- *Sports*

Per i due dataset Wikipedia sono riportati nelle tabelle sottostanti le feature caratterizzanti l'intero corpus.

Dataset 1	**Hapax**	**WHapax**	**WTerminiFreq**
#documenti	995	995	995
#termini	1653102	1606393	1006923
Max frequenza	8530	8530	6955
Min frequenza	1	2	2
Avg frequenza	16	34	25
Dizionario |V|	102712	56004	55335
% Hapax	45.5	0	0
TTR	0.06	0.03	0.03
Coeff. Guiraud	79.89	44.19	45.08

Dataset 2	**Hapax**	**WHapax**	**WTerminiFreq**
#documenti	2477	2477	2477
#termini	3195762	3109576	2095128
Max frequenza	10512	10512	9862
Min frequenza	1	2	2
Avg frequenza	21	44	32
Dizionario |V|	145871	78369	76387
% Hapax	59.6	0	0
TTR	0.04	0.02	0.03
Coeff. Guiraud	81.58	44.56	52.77

4.1.2 Dataset Twitter

Gli altri dataset che sono stati oggetto di studio sono costruiti a partire da collezioni di messaggi provenienti dal social network Twitter [22]. Questo set di dati presenta tuttavia caratteristiche complesse. Per definizione, un tweet è costituito infatti da poche parole che racchiudono il topic (stringa di massimo 140 caratteri) e la varietà del dizionario è molto inferiore rispetto a quella generata dagli articoli Wikipedia.
In particolare, si è scelto di analizzare i tweet relativi sia a catastrofi[4], analizzando un dataset che contiene dati già etichettati come on-topic e off-topic, sia i tweet pubblicati durante una puntata della trasmissione *The Voice of Italy*.

Il dataset D3 è costruito analizzando solo i tweet on-topic. Il dataset originario è parsificato in modo tale da avere un documento per ogni tweet, il cui contenuto riguarda sei macro categorie:

- *Boston_bombing*
- *Oklahoma_tornado*
- *Queensland_floods*
- *West_texas_explosion*
- *Alberta_floods*
- *Sandy_hurricane*

Il dataset D4 è invece costruito scegliendo un sottoinsieme composto dal 20% dei documenti di ciascun topic, senza distringuire tra on-topic e off-topic.

Dataset 3	**Hapax**	**WHapax**	**WTerminiFreq**
#documenti	32853	32853	32853
#termini	289080	268359	154088
Max frequenza	28599	28599	24106
Min frequenza	1	2	2
Avg frequenza	45	62	54
Dizionario |V|	33155	12434	12280
% Hapax	62.9	0	0
TTR	0.1	0.04	0.07
Coeff. Guiraud	61.66	24.69	31.33

[4] https://crisislex.org/data-collections.html#CrisisLexT6

Dataset 4	Hapax	WHapax	WTerminiFreq
#documenti	12064	12064	12064
#termini	98180	84213	65269
Max frequenza	8966	8966	6429
Min frequenza	1	2	2
Avg frequenza	30	59	47
Dizionario |V|	22326	9959	8731
% Hapax	62.5	0	0
TTR	0.2	0.10	0.13
Coeff. Guiraud	71.25	34.31	34.17

Il dataset D5 riguarda una collezione di tweet pubblicati durante una puntata del noto show televisivo *The Voice of Italy*. La puntata presa in considerazione è l'ottava della prima stagione, andata in onda il 25 aprile 2013. Il programma è un talent show musicale dove quattro artisti di fama internazionale (*Carrà*, *Noemi*, *Pelù*, *Cocciante*) formano squadre composte da giovani aspiranti cantanti. Nella puntata si esibiscono quattro artisti per ogni coach. Inoltre, sono presenti anche due ospiti d'eccezione, *Antonacci* e *Mengoni*. Durante la puntata, che dura circa tre ore, i tweet si sono concentrati sulle esibizioni dei concorrenti e degli ospiti, dalla competizione tra i coach fino alle performance dei cantanti.
Lo scandirsi temporale di questi eventi caratterizza il dataset risultante.
A differenza dei dataset Wikipedia, i tweet sono in lingua italiana e le collezioni sono state perciò precedentemente elaborate con una lista di stopword in italiano.

Dataset 5	Hapax	WHapax	WTerminiFreq
#documenti	37981	37981	37981
#termini	396852	385963	206966
Max frequenza	18245	18245	13981
Min frequenza	1	2	2
Avg frequenza	76	98	85
Dizionario |V|	27213	13188	10228
% Hapax	52.6	0	0
TTR	0.06	0.03	0.04
Coeff. Guiraud	43.19	21.22	22.48

	Funzione di pesatura	K	Log-likelihood	ASI	GSI
D1	**TF-IDF**	3	-6307664.5334	0.8069	0.8055
		5	-6161099.9886	0.7865	0.7770
		8	-6034579.5317	0.7593	0.7274
		10	-5992059.6105	0.7305	0.7358
	Boolean-TF_{glob}	5	-2730914.1805	0.6866	0.6812
		6	-2725755.4904	0.6923	0.6829
		9	-2692499.7131	0.6361	0.6291
		13	-2651550.1501	0.6261	0.6199
D2	**TF-IDF**	3	-7128823.69145	0.6715	0.6716
		5	-6842176.87149	0.5964	0.5925
		9	-6709931.00257	0.3753	0.3733
	Boolean-TF_{glob}	3	-3956214.32251	0.6634	0.6639
		6	-3899610.25501	0.5755	0.5777
		13	-3782258.58994	0.5394	0.5273
D3	**Boolean-Entropy**	5	-302731.75087	0.6988	0.6814
		9	-299155.69825	0.6781	0.6770
		14	-290153.47752	0.4598	0.4556
D4	**Boolean-Entropy**	5	-258996.36100	0.5996	0.6025
		9	-238963.36998	0.3574	0.3677
D5	**Boolean-IDF**	5	-512027.36962	0.5992	0.5982
		6	-508963.25512	0.5124	0.5099

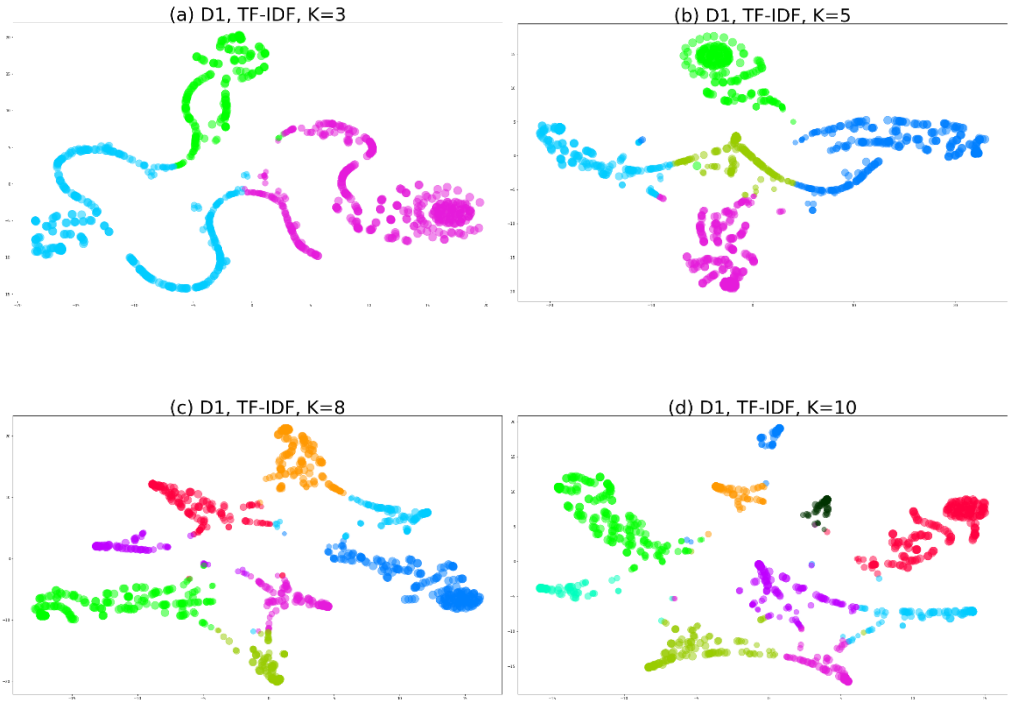
Tabella 4.1: Risultati CONCEPT

4.2 Dettagli di un dataset

In questa sezione vengono mostrati nel dettaglio i risultati ottenuti dal modello PLSA e le diverse tecniche di visualizzazione e validazione per il dataset D1. Questo dataset contiene circa 1000 documenti relativi ai temi di cucina, letteratura, matematica, musica e sport. L'algoritmo è calcolato per diversi valori di K, numero di topic, per confrontare la qualità dei risultati prodotti; questo numero è settato a priori, insieme al numero di iterazioni, fissato nel nostro caso a 100, per far convergere il modello.

4.2.1 Dataset D1, TF-IDF

I risultati ottenuti con la funzione di pesatura TF-IDF sono riportati nella Tabella 4.1, con i valori K selezionati 3, 5, 8, 10. Dalla tabella possiamo affermare che gli indici ASI, GSI e Log-likelihood riflettono le caratteristiche dei cluster trovati: all'aumentare di K la Log-likelihood diminuisce, mentre i valori di Silhouette locali e globali sono pressoché uguali. Come discusso in precedenza, le sole metriche quantitative non sono sufficienti per valutare la qualità complessiva del processo di clustering. Quindi, per esaminare correttamente i risultati, si utilizza il grafico t-SNE e alcune rappresentazioni argomento-termini molto intuitive.



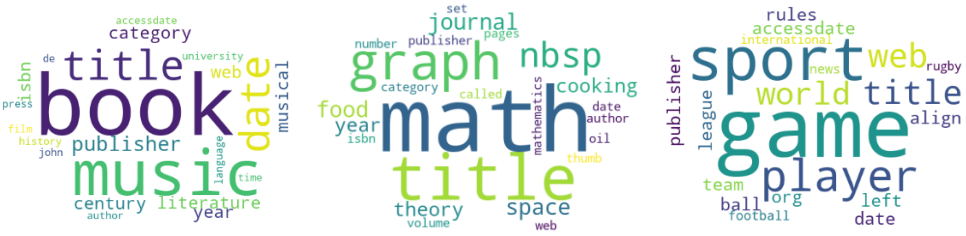


Figura 4.2: Word cloud, D1, TF-IDF, K=3

Dalla rappresentazione t-SNE è possibile vedere che i cluster sono quasi tutti bilanciati in termini di cardinalità di documenti al loro interno e il contenuto dei cluster ottenuti sta effettivamente riflettendo la struttura del dataset, che contiene cinque macro categorie. Per K uguale a 3 è possibile identificare i seguenti argomenti: matematica, sport e letteratura/musica. È chiaro che, essendo tre minore del numero effettivo di argomenti originali, alcuni di essi risultano mescolati e potrebbe essere necessario un numero maggiore K per rappresentare meglio il dataset. Nello scatter plot e stacked-bar sono rappresentate le probabilità associate ad ogni documento di appartenere al topic K: come si vede, le distribuzioni assegnate sono molto diverse tra loro e variano da valori molto piccoli ad uno grande a cui corrisponde l'argomento finale. Per questo motivo, ci aspettiamo cluster omogenei e coesi.

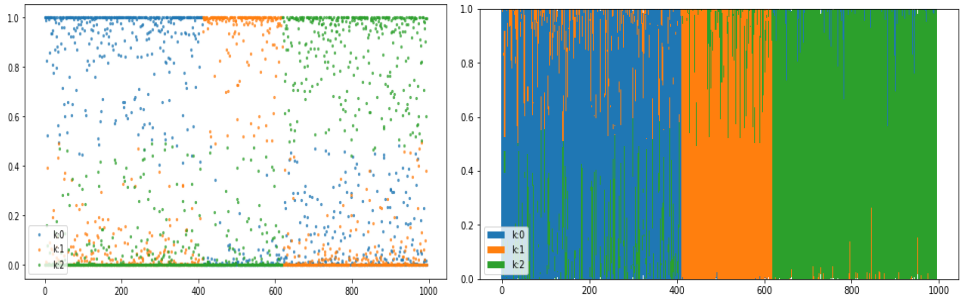


Figura 4.3: Scatter plot a sinistra e stacked bar a destra, D1, TF-IDF, K=3

Con 5 cluster la PLSA divide bene i documenti che appartengono ai seguenti argomenti: letteratura / musica, sport, matematica. L'argomento cucina, pur essendo rappresentato nel cluster, è ancora mescolato con l'argomento letteratura. Dalla visualizzazione con il grafo possiamo osservare che, per K=5, le cinque categorie condividono tra loro alcune parole, ad esempio "*title*", "*book*", che, relativamente al contesto, possono assumere dei significati e utilizzi diversi. La parola *libro* può infatti essere inglobata nella categoria letteratura, ma anche in quella di cucina se si pensa ad un libro di ricette.

Per il grafo sono state visualizzate le prime 25 parole che il modello valuta come caratterizzanti per ciascun topic. Il coefficiente di densità del grafo, che valuta quanto i termini sono legati fra loro, è pari a 0.028. Il valore è coerente con le visualizzazioni precedenti.

Figura 4.4: Word cloud, D1, TF-IDF, K=5

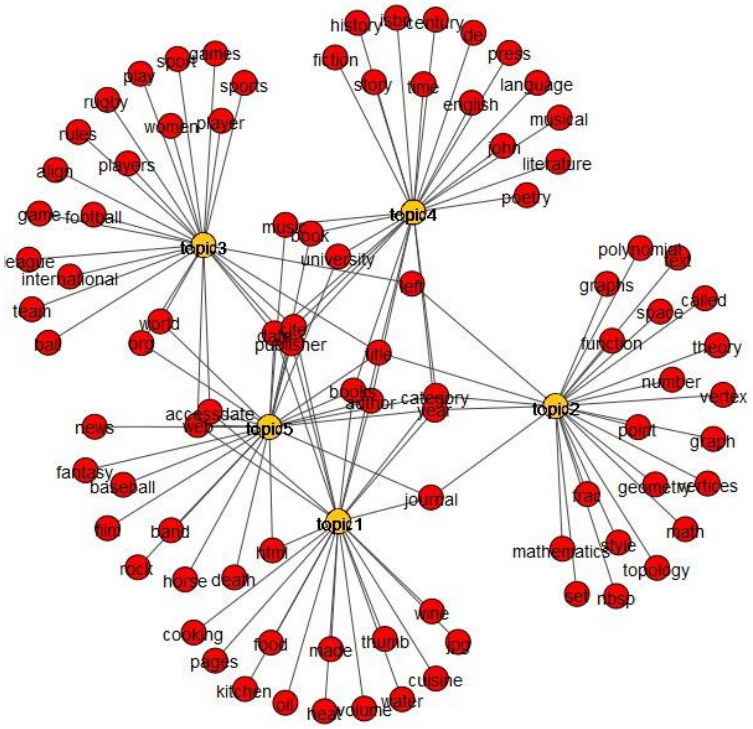


Figura 4.5: Grafo, D1, TF-IDF, K=5

Per visualizzare i risultati ottenuti per K pari a 10, è riportato un sottoinsieme delle word cloud che contengono le parole rappresentative. Come è possibile vedere anche dalla relativa rappresentazione t-SNE. i cinque cluster più grandi vengono visualizzati bene e descrivono le principali categorie del dataset (cioè letteratura, musica, matematica, cucina e sport). Come si vede dallo scatter plot e stacked bar, le cardinalità dei cluster sono meno omogenee, perciò più sbilanciate.

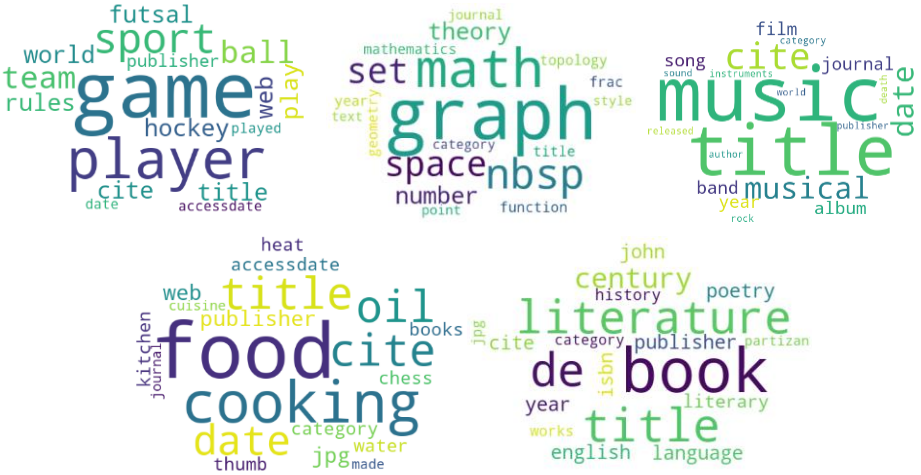


Figura 4.6: Word cloud, D1, TF-IDF, K=10

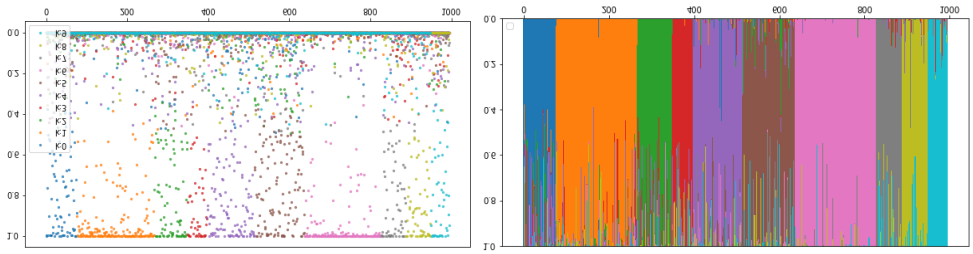


Figura 4.7: Scatter plot a sinistra e stacked bar a destra, D1, TF-IDF, K=10

Infine, per valutare la similarità tra gli esperimenti al variare di K, sono riportate le tabelle relative al Rand Index per K=5 e K=6. Si valutano tutte le possibili combinazioni tra K diversi: essendo il numero di cluster diverso, l'indice non raggiungerà mai il valore massimo 1, ma dai risultati ottenuti i valori si concentrano nella fascia media che va da 0.5 allo 0.9.

Cluster simili a K = 5

K	Rand Index
3	0.5190
6	0.6815
8	0.5283
10	0.6007

Cluster simili a K = 6

K	Rand Index
3	0.6235
5	0.6889
8	0.5571
10	0.5521

4.2.2 Dataset D1, Boolean-TF_{glob}

I risultati ottenuti con la funzione di pesatura Boolean-TF_{glob} sono riportati nella Tabella 4.1: i valori di K osservati sono pari a K = 5, 6, 9, 13. Osservando la tabella, i valori di ASI e GSI sono minori rispetto al calcolo ottenuto con TF-IDF: tuttavia i risultati sono da considerarsi buoni poiché superano il valore di 0.5. Per avere una visione più chiara dei risultati, utilizziamo le rappresentazioni grafiche viste in precedenza. La rappresentazione t-SNE mostra che i risultati prodotti dal processo di clustering sono per la maggior parte bilanciati in termini di cardinalità di ciascun cluster. Anche in questo caso, per K > 5, si distinguono cinque cluster più grandi che raggruppano i documenti appartenenti alle macro-categorie, e altri cluster più piccoli. Tuttavia, osservando la rappresentazione argomento-termini, alcuni cluster ottenuti sembrano contenere quasi le stesse parole, mescolandole, e vengono considerati outlier.

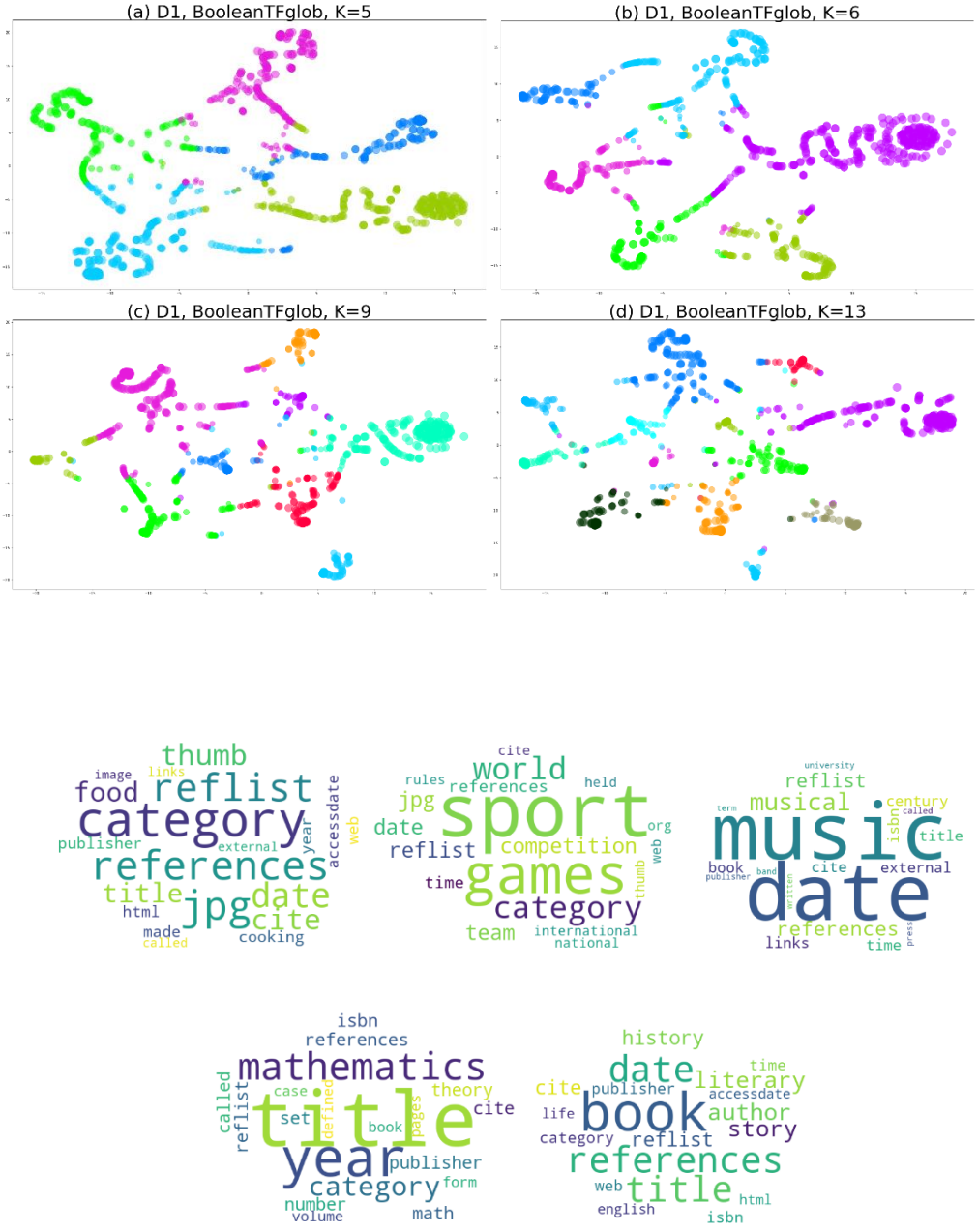


Figura 4.8: Word cloud, D1, Boolean-TF_{glob}, K=5

Considerando il clustering ottenuto con K uguale a 5, osserviamo che gli argomenti di letteratura, musica, matematica e sport sono tutti generati dalle word cloud, eccezione fatta per l'argomento di cucina che si trova mescolato in un unico cluster insieme alla letteratura. Ciò accade perché lo schema Boolean-TF_{glob} dà maggiore rilevanza alle parole che appaiono più nel corpus rispetto al singolo documento. Tuttavia, spesso accade che queste parole non appartengano ad alcun argomento specifico, o semplicemente non portano alcuna informazione aggiuntiva utile per la modellazione.

Per questo motivo, per non considerare tali termini, è stato applicato un ulteriore passaggio nella fase di pre-processing per eliminare le parole troppo frequenti e in seguito valutare i risultati. Per K = 5, sono state quindi rimosse le *code* che non portano informazioni specifiche sull'argomento: il grafo risultante risulta avere un valore di densità leggermente più elevato, 0.030, poiché le parole risultano più connesse tra loro.

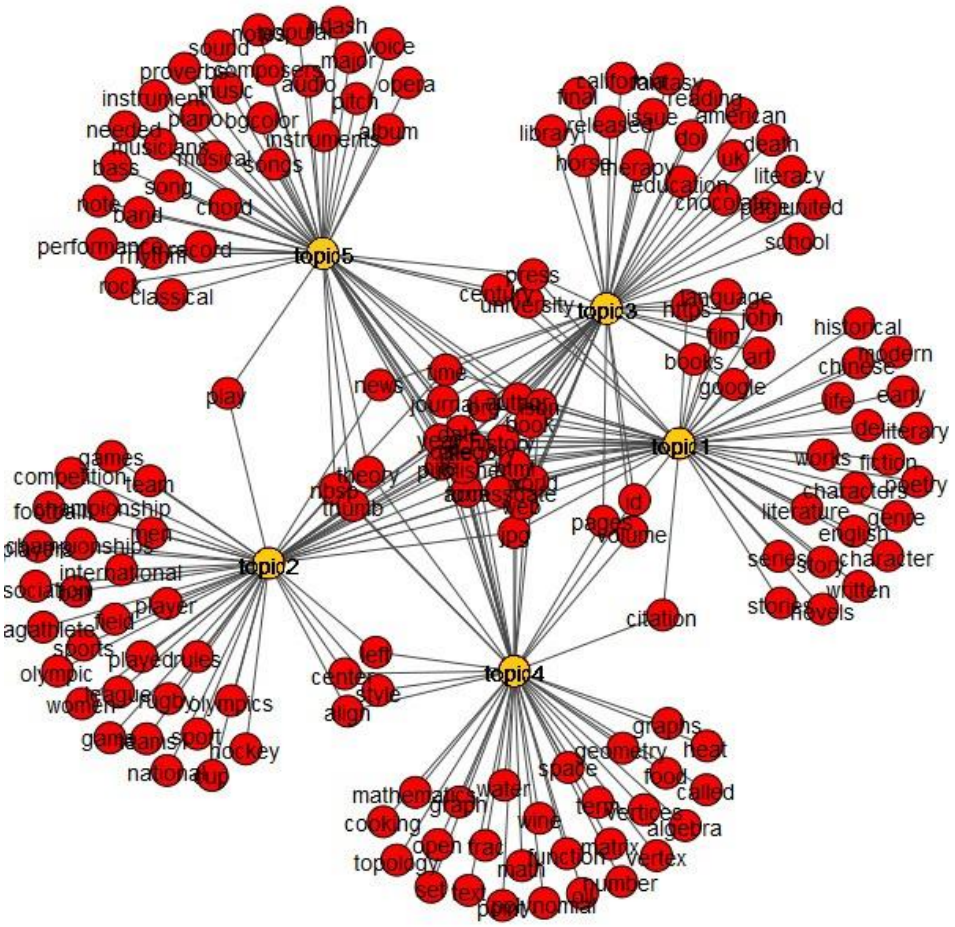


Figura 4.9: Grafo, D1, Boolean-TF_{glob}, K=5

Guardando la rappresentazione t-SNE dei risultati ottenuti con K pari a 13, la rappresentazione genera dei cluster non più omogenei, ma sbilanciati verso le cinque macro categorie tematiche. Guardando un sottoinsieme delle word cloud e le etichette assegnate ai documenti, è possibile capire meglio come CONCEPT ha operato. Molti dei cluster trovati sembrano avere un buon livello di coesione, con una chiara definizione dei cinque topic musica, sport, letteratura, matematica e cibo, come in Figura 4.10. Gli altri cluster generati mescolano insieme argomenti storico/letterari e legati all'arte, e hanno una cardinalità minore.

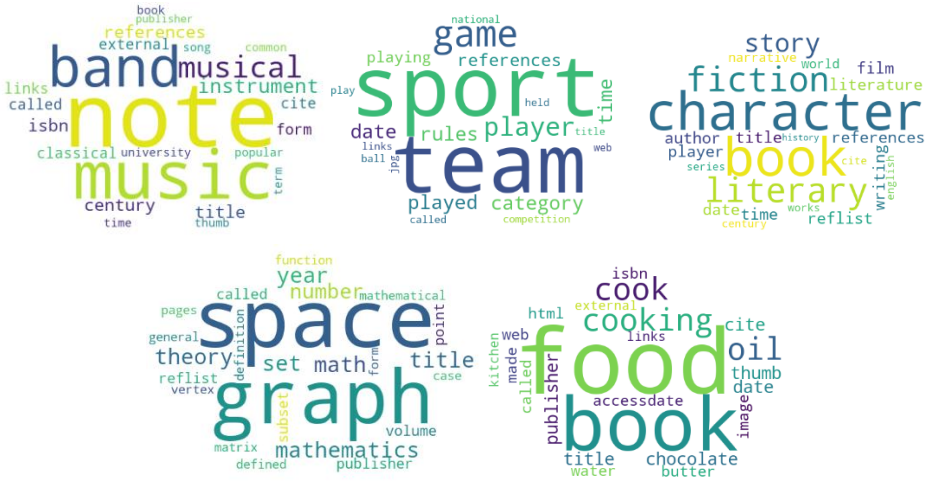


Figura 4.10: Word cloud, D1, Boolean-TF_{glob}, K=13

Anche per questa funzione di pesatura è possibile valutare la similarità tra i risultati prodotti dal processo di clustering al variare del parametro K. Sono riportate le tabelle relative al Rand Index per K=5 e K=9.

Cluster simili a K = 5

K	Rand Index
3	0,5869
9	0.6244
13	0.6258

Cluster simili a K = 9

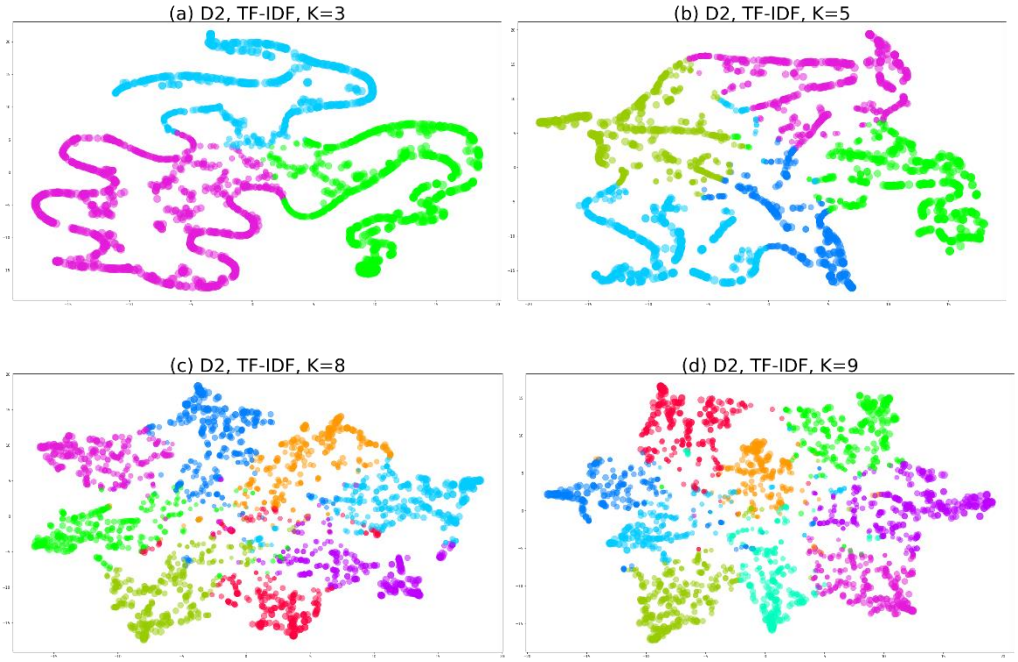
K	Rand Index
3	0.6522
5	0.5752
13	0.5801

4.3 Altri risultati

In questa sezione sono presentati i risultati ottenuti con i dataset D2, *2500_wiki_sample*, e D3, D4 e D5, *tweet* Per ciascuno di essi, sono riportate le visualizzazioni dei risultati migliori.

4.3.1 Dataset D2, TF-IDF

Le figure mostrano i risultati ottenuti applicando la funzione di pesatura TF-IDF, visualizzati con i grafici t-SNE e un sottoinsieme di word cloud. Osservando la Tabella 4.1, i valori di ASI e GSI seguono lo stesso trend e all'aumentare del numero di argomenti K subiscono una netta diminuzione. I valori di K= 8, 9 sono da considerarsi il caso peggiore. Allo stesso modo la Log-likelihood assegna una verosimiglianza minore all'aumentare di K. Osservando i grafici, questi riflettono la struttura del dataset che contiene dieci categorie. La rappresentazione t-SNE genera cluster in gran parte coesi e omogenei come cardinalità, mentre nelle word cloud sono mostrati una parte degli argomenti che CONCEPT riesce a generare per K=9. Infine, sono riportati anche i risultati per il calcolo di cluster simili a K=9, che sono buoni e coerenti con le rappresentazioni precedenti.



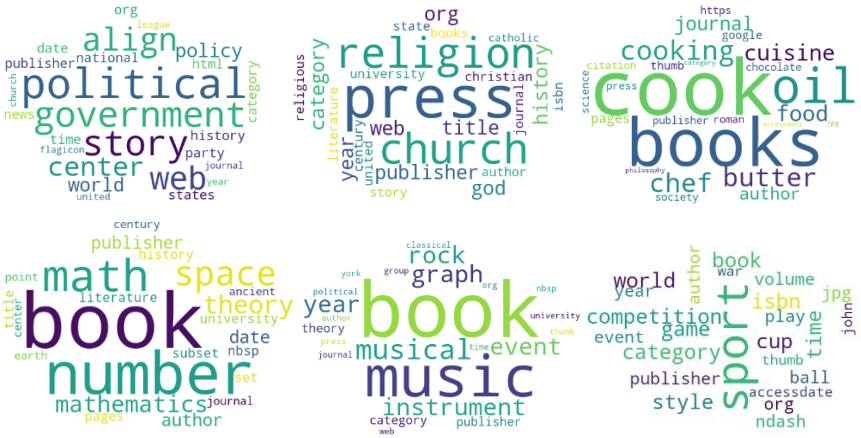


Figura 4.11: Word cloud, D2, TF-IDF, K=9

Cluster simili a K = 9

K	Rand Index
3	0.7456
5	0.8106
8	0.8044

4.3.2 Dataset D2, Boolean-TF$_{glob}$

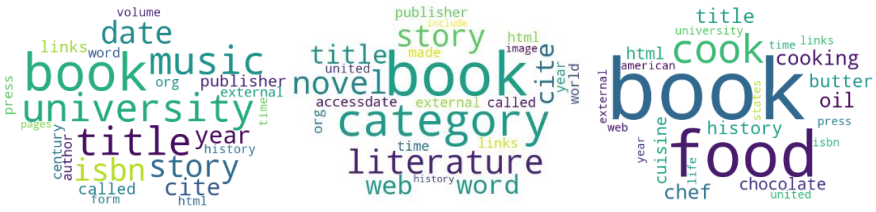


Figura 4.12: Word cloud, D2, Boolean-TF$_{glob}$, K=3

Di seguito sono riportati i risultati migliori prodotti dalla funzione di pesatura Boolean-TF$_{glob}$. Osservando il risultato delle word-cloud con K=3, come previsto, il modello non riesce a separare gli argomenti e le *word cloud* contengono praticamente le stesse parole, identificando un unico grande argomento, letteratura. Se si aumenta il numero K, come mostra la figura per K = 6, il modello separa i topic seppur con qualche errore, ad esempio la parola *book* compare con probabilità diverse in tutti i topic.

I grafici t-SNE generano dei cluster ben separati e omogenei. In particolare, per K=10 e K=13, i grafici mostrano una rappresentazione a stella, dove sulle punte sono concentrati molti documenti e nel centro diventano più sparsi con il rischio di non convergere.

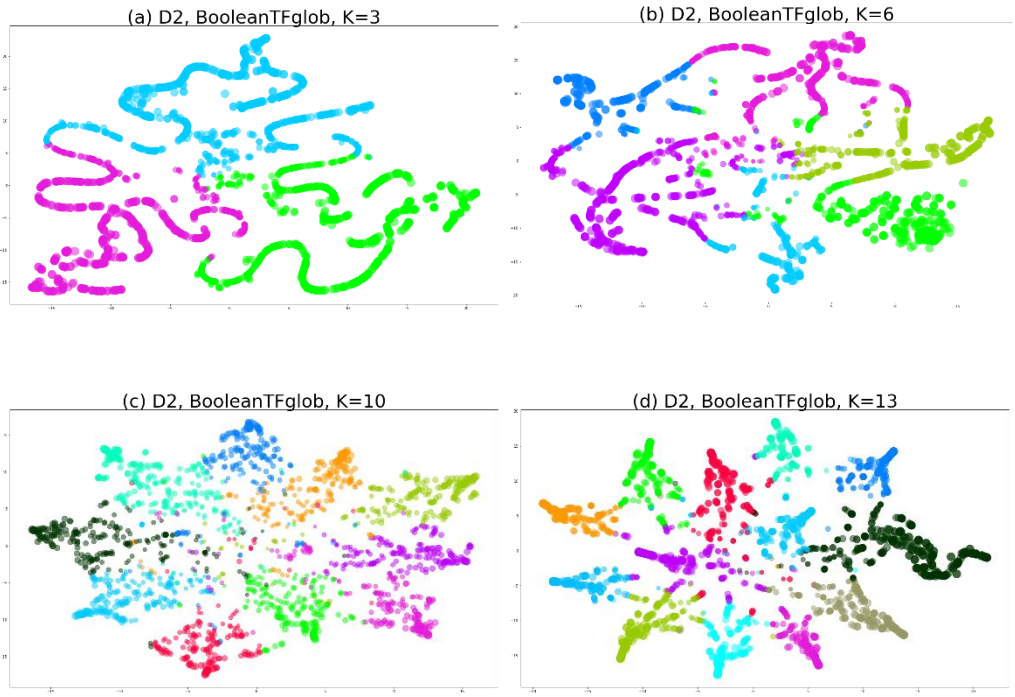


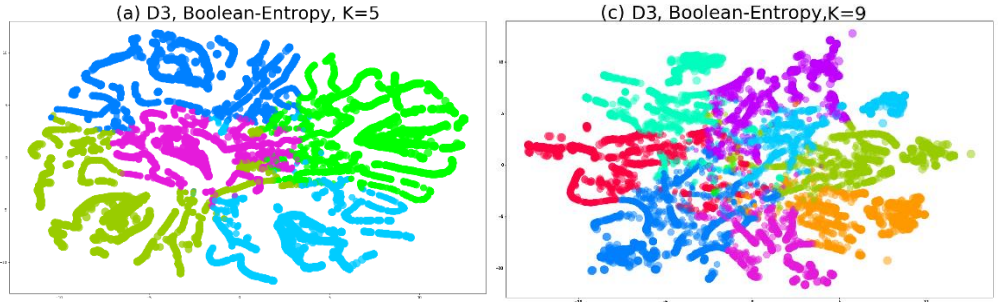
Figura 4.13: Word cloud, D2, Boolean-TF$_{glob}$, K=6

Cluster simili a K = 10

K	Rand Index
3	0.8422
6	0.8270
13	0.6392

4.3.3 Dataset D3, Boolean-Entropy

Le figure che seguono mostrano i risultati generati da una nuova tecnica di pesatura, il prodotto tra il peso locale Boolean e quello globale Entropy, applicati ad un nuovo tipo di dataset. Le visualizzazioni con t-SNE e word cloud generano dei cluster coesi e omogenei in termini di cardinalità dei documenti al loro interno. I grafici appaiono inoltre più *densi* rispetto ai dataset precedenti poiché contengono molti più documenti. Le *word cloud* generate identificano in maniera corretta le catastrofi del dataset originario: Boston_bombing, Texas_explosion e Oklahoma_tornado sono rappresentati come *cloud* perfette, contenenti termini appartenenti solo a quel topic. Il cluster relativo a Sandy_floods, invece, contiene sia la parola *flood* che *hurricane*, sintomo che il modello sta sbagliando nell'assegnazione dei cluster relativi a uragani e inondazioni.



È inoltre riportato in Figura 4.15 il grafo per K=5, contenente ai nodi gialli i topic, a cui sono collegate le parole caratterizzanti l'argomento. Come possiamo vedere, i risultati sono molto buoni poiché i cluster che CONCEPT ha creato condividono pochi termini. In particolare, i cluster relativi al topic 4 e topic 5 appaino quasi isolati dagli altri.

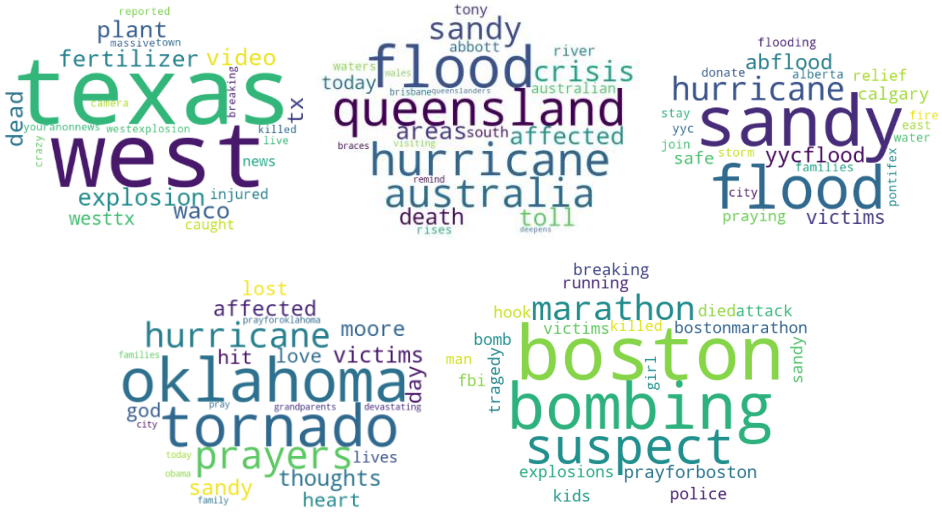


Figura 4.14: Word cloud, D3, Boolean-Entropy, K=5

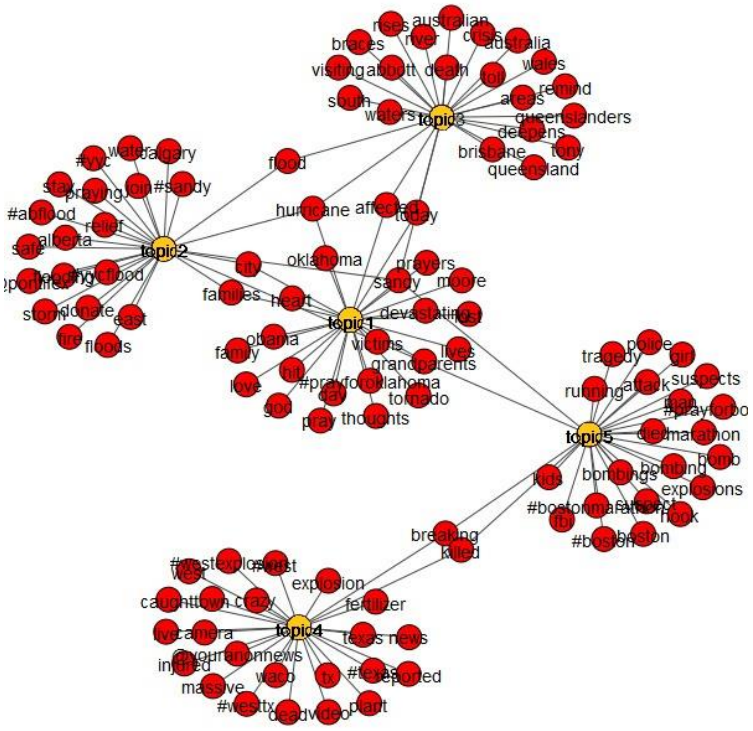
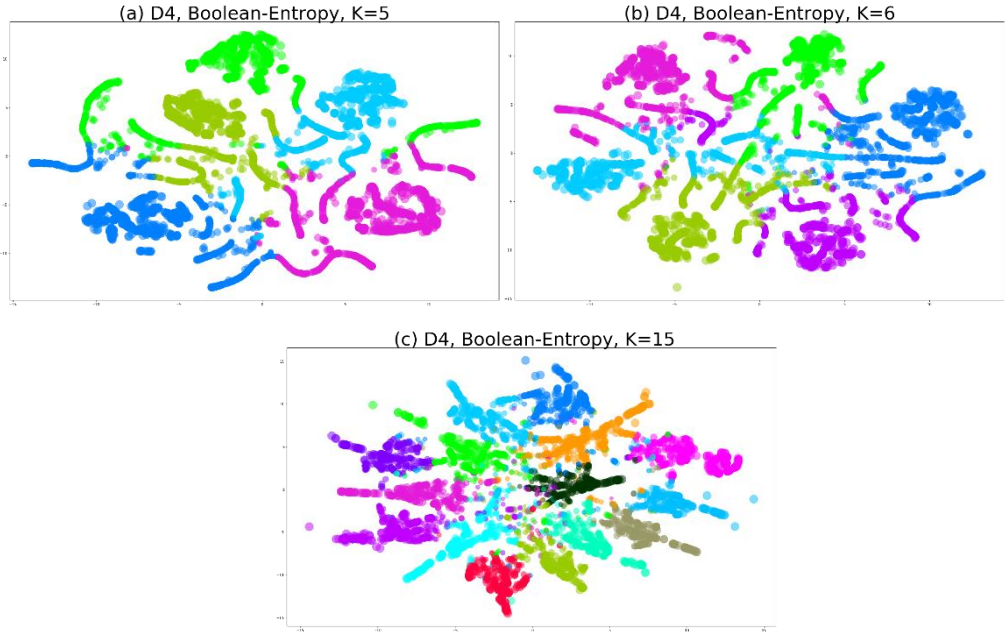


Figura 4.15: Grafo, D3, Boolean-Entropy, K=5

4.3.4 Dataset D4, Boolean-Entropy



Le figure mostrano i risultati generati dalla funzione di pesatura Boolean-Entropy, applicata al dataset D4. Le visualizzazioni con t-SNE mostrano dei raggruppamenti equilibrati e omogenei, bilanciati nel numero di documenti assegnati a ciascuno di essi. Le *word cloud* generate identificano in maniera corretta e in gran parte soddisfacente le catastrofi del dataset originario. Tuttavia, all'aumentare del numero K, solo un sottoinsieme dei cluster identifica gli argomenti del corpus, mescolandoli, mentre gli altri descrivono argomenti particolari, off-topic, non rilevanti per la modellazione. Questo accade non solo per la struttura intrinseca del dataset, che contiene anche tweet considerati *rumore*, ma perché il peso locale Boolean dà maggiore rilevanza alle parole che appaiono più nel corpus rispetto a quelli del singolo documento.

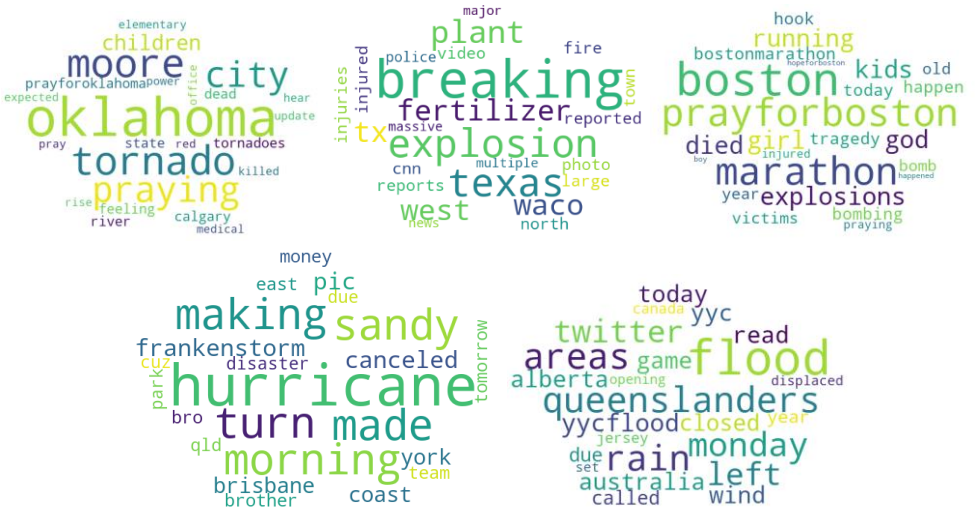


Figura 4.16: Word cloud, D4, Boolean-Entropy, K=15

4.3.5 Dataset D5, Boolean-IDF

I grafici mostrano i risultati prodotti dalla funzione di pesatura Boolean-IDF per il dataset D5. Il grafico t-SNE per K=5 mostra dei cluster coesi e in gran parte ben separati. Per K=6 riusciamo ancora a distinguere i gruppi che il modello ha creato, ma alcuni cluster si sovrappongono e il modello ne risente in termini di bontà: la Silhouette e Log-likelihood infatti diminuiscono. I grafici appaiono sempre molto *densi* per il numero di documenti del corpus. Per capire meglio come il modello ha lavorato, in Figura 4.17 sono riportate le word cloud per K=5. In tutte le *cloud* è riportato l'hashtag #tvoi, a cui tutti i tweet erano collegati. Il modello riesce in gran parte a dividere bene le parole in macro categorie: una figura contiene i tweet riguardanti la trasmissione in generale, i nomi di tutti i coach e ospiti che si sono esibiti durante la puntata. Un'altra *cloud* è invece riferita all'artista Noemi, che compare in molte figure con probabilità diverse, e al suo team.

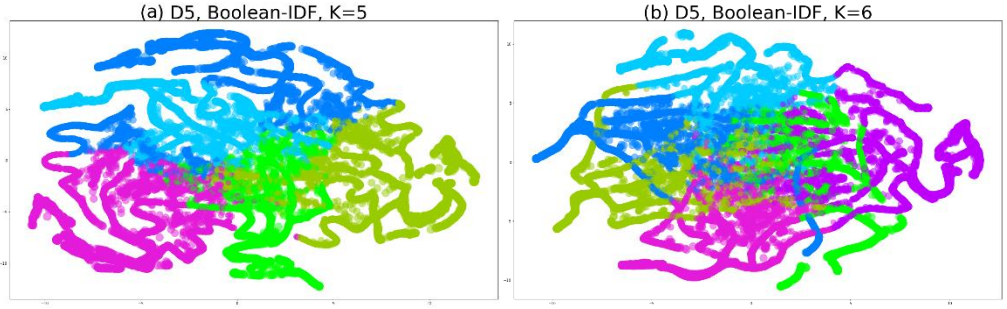


Figura 4.17: Word cloud, D5, Boolean-IDF, K=5

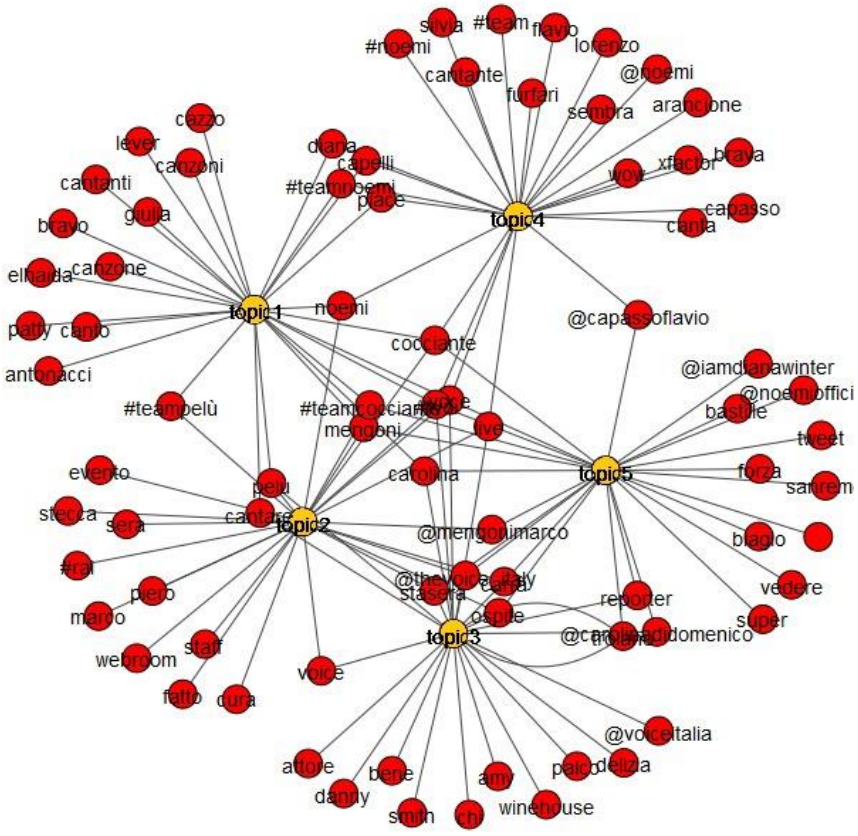


Figura 4.18: Grafo, D5, Boolean-IDF, K=5

4.4 CONCEPT considerazioni finali

Dai risultati ottenuti possiamo affermare che il framework CONCEPT riesce nel tentativo di raggruppare i documenti in base al loro contenuto. Infatti, i risultati mostrano cluster ben separati ed omogenei, e quindi il modello riesce a raggruppare i testi in argomenti distinti. Le funzioni di pesatura esplorate (e.g. TF-IDF, Boolean-TF_{glob}), note e ampliamente utilizzate in letteratura, hanno permesso di analizzare il corpus da punti di vista differenti.
La funzione TF-IDF riesce a trovare cluster ben separati e con parole differenti tra loro, infatti aumenta proporzionalmente al numero di volte che il termine è contenuto nel documento, ma cresce inversamente alla frequenza del termine nella collezione. Al contrario, Boolean-TF_{glob} rivela i termini più frequenti nell'intera collezione e genera cluster meno coesi e con più parole condivise tra i diversi argomenti. Tuttavia, questa funzione di pesatura non è da considerarsi negativa in quanto produce cluster omogenei e coerenti con la struttura del dataset.
I risultati ottenuti sono stati validati ed analizzati con l'utilizzo di diverse tecniche di rappresentazione e indici di qualità, che evidenziano aspetti diversi del clustering. Ad esempio, se *word cloud* evidenzia le parole più importanti che caratterizzano un topic, la rappresentazione con il *grafo* ci dà una stima di quanto questi risultati siano buoni in termini di similarità con le parole degli altri argomenti. Inoltre, il calcolo della densità del grafo restituisce un valore molto utile per interpretare la bontà globale del risultato. Per quanto riguarda gli indici di qualità, i valori di Silhouette e Rand Index sono stati necessari per valutare la bontà dei cluster creati. Il valore di Silhouette, mediato e globale, è necessario a valutare la coesione/separazione tra cluster, mentre il Rand Index permette di fare una stima della bontà della clusterizzazione rispetto a tutti gli esperimenti.

Avere molte tecniche di analisi dei risultati è stato un obiettivo del lavoro di tesi in gran parte raggiunto. L'inclusione di molte tecniche intuitive aumenta il livello di descrizione e interpretabilità dei risultati: l'analista, senza essere un esperto di dominio, può analizzare in maniera semplice ma efficace le configurazioni che meglio raggruppano i documenti in gruppi tematici.

Capitolo 5

5 Conclusione e sviluppi futuri

Questa tesi presenta un sistema di analisi dei dati testuali che estrae dai singoli documenti i concetti più rilevanti. Il sistema sviluppato include una serie di strategie mirate a semplificare il processo di visualizzazione e validazione dei risultati ottenuti dal modello, affinché gli utenti finali possano accedervi in maniera ottimale senza essere esperti di dominio.
Questo studio è iniziato dall'analisi di approcci già esistenti nel campo del text mining, partendo dal modello algebrico LSA e da quello probabilistico LDA, e si è focalizzato sull'implementazione del modello probabilistico PLSA. Intuitivamente, l'algoritmo prevede che i documenti possano essere associati a un particolare argomento (topic) o possono essere visti come una combinazione di argomenti in proporzioni diverse, a seconda delle parole che occorrono nel testo. Il framework sviluppato CONCEPT è in grado di descrivere gli argomenti (e quindi i documenti) per mezzo di cluster di parole simili e riesce a formare dei cluster coesi e omogenei di documenti appartenenti allo stesso topic. Poiché l'estrazione di informazione utile è complessa, l'obiettivo di questa tesi è stato sviluppare un framework in grado di visualizzare i risultati prodotti dalla clusterizzazione in maniera ottimale, intuitiva e semplice, capace di essere visualizzata efficacemente.
Il framework proposto è stato convalidato su diverse collezioni con strutture diverse, caratterizzate da diverse proprietà statistiche: articoli di Wikipedia, molto lunghi e con grande varietà di dizionario, e vaste collezioni di tweet, limitati dai 140 caratteri.
I risultati sperimentali ottenuti sono stati valutati prendendo in considerazione metriche quantitative come Log-likelihood, Silhouette (ASI e GSI) e Rand Index, ma anche mediante tecniche di visualizzazione innovative. I risultati hanno mostrato in gran parte l'efficacia del modello PLSA e fanno premettere che questo algoritmo possa essere applicato anche a collezioni di dati molto grandi, nell'ambito dei Big Data.

Per questo motivo, questo studio apre la strada a interessanti sviluppi futuri:

(i) la progettazione di una strategia auto-configurante per settare il numero di parametri K (topic), in grado di suggerire all'analista la configurazione che produce una conoscenza di qualità superiore.

(ii) trovare alternative che consentano una esecuzione parallela e distribuita, con una ridotta complessità computazionale, per garantire una maggiore scalabilità.

(iii) possibilità di valutare tecniche alternative, come l'analisi di modelli non parametrici (e.g. Deep Neural Network DNN), che abbiano caratteristiche migliori o semplicemente diverse, in modo da fornire all'analista sempre più alternative.

Bibliografia

[1] J. Han, Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[2] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), 27-34.

[3] Wang, C., & Blei, D. M. (2011, August). Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 448-456). ACM.

[4] Tasha N. Underhill, An Introduction to Information Retrieval using Singular Value Decomposition and Principal Component Analysis, 2007 [19] Herve Abdi, Lynne J. Williams, Principal component analysis, John Wiley & Sons, Inc., 2010

[5] E. Di Corso, T. Cerquitelli, and F. Ventura. Self-tuning techniques for large scale cluster analysis on textual data collections. In Proceedings of the Symposium on Applied Computing, pages 771–776. ACM, 2017.

[6] Bro, R. (1997). PARAFAC. Tutorial and applications. Chemometrics and intelligent laboratory systems, 38(2), 149-171.

[7] T. Cerquitelli, E. Di Corso, F. Ventura, and S. Chiusano. Data miners' little helper: Data transformation activity cues for cluster analysis on document collections. In Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS '17, pages 27:1–27:6, New York, NY, USA, 2017. ACM.

[8] Preslav Nakov and Antonia Popova and Plamen Mateev, Weight functions impact on LSA performance, EuroConference RANLP'2001 (Recent Advances in NLP), 187–193.

[9] Ding, C., & He, X. (2004, July). K-means clustering via principal component analysis. In Proceedings of the twenty-first international conference on Machine learning (p. 29). ACM.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022, 2003.

[11] S. Proto, E. di Corso, F. Ventura, and T. Cerquitelli, Useful ToPIC: Self-tuning strategies to enhance Latent Dirichlet Allocation - (2018), pp. 33-40.

[12] Hofmann, T. (1999, July). Probabilistic latent semantic analysis. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 289-296). Morgan Kaufmann Publishers.

[13] De Pierro, A. R. (1995). A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. IEEE transactions on medical imaging, 14(1), 132-137.

[14] Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference, 90(2), 227-244

[15] Peter J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics, 53 - 65

[16] Nidheesh, N., Nazeer, K. A., & Ameer, P. M. (2018). A Hierarchical Clustering Algorithm Based on Silhouette Index for Cancer Subtype Discovery from Omics Data. bioRxiv, 309716.

[17] Santos, J. M., & Embrechts, M. (2009, September). On the use of the adjusted rand index as a metric for evaluating supervised classification. In International Conference on Artificial Neural Networks (pp. 175-184). Springer, Berlin, Heidelberg.

[18] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov):2579–2605, 2008.

[19] Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695(5), 1-9

[20] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

[21] Battaglia, R. (1954). L'espressione statistica della variabilità e la variabilità somatica negli ibridi. Zeitschrift für Morphologie und Anthropologie, (H. 3), 421-424.

[22] E. Di Corso, F. Ventura, and T. Cerquitelli. All in a twitter: Self-tuning strategies for a deeper understanding of a crisis tweet collection. pages 3722–3726, 12 2017