



Università degli Studi di Bari - “Aldo Moro”

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in

Informatica e Tecnologie per la Produzione del Software

TESI DI LAUREA
IN
PROGRAMMAZIONE II

Applicazione di tecniche di Word Embedding e Text Mining per il Clustering di pagine in un grafo Web

Laureando:

Andrea Del Fante

Relatore:

Chiar.mo Prof. Michelangelo Ceci

Correlatore:

Dott.ssa Pasqua Fabiana Lanotte

Indice

1	Informazioni latenti nel Web	1
1.1	Web Mining	2
1.2	Rappresentazioni di pagine Web	7
1.2.1	Word space model	8
1.2.2	Word2Vec	14
1.2.3	Doc2Vec	16
1.2.4	LINE	16
1.3	Clustering	19
1.3.1	Funzioni (o misure) di distanza	21
1.3.2	Algoritmi usati	23
1.4	Obiettivi della tesi	26
2	Stato dell'Arte	27
2.1	Altre metodologie di Clustering	30
2.1.1	Combinazione di embeddings	31
3	Metodologia	33
3.1	Web Crawling	33
3.1.1	Crawling delle pagine di un sito Web	35
3.1.2	Normalizzazione degli URL	37
3.2	Costruzione del dataset	38
3.2.1	Random Walks	38
3.2.2	Generazione delle sequenze	41
3.2.3	Modifica dell'implementazione di Word2Vec	43

3.2.4	Scaling degli embeddings	44
3.3	Web page Clustering	46
3.4	Esempio di dataset	47
4	Sperimentazione	51
4.1	Dataset	52
4.2	Metriche	53
4.3	Configurazioni	57
4.3.1	Testo	57
4.3.2	Random Walk	58
4.3.3	Combinato	60
4.4	Parametri degli algoritmi di Clustering	60
4.5	Analisi dei risultati	61
4.5.1	cs.illinois.edu	61
4.5.2	cs.ox.ac.uk	63
4.5.3	cs.princeton.edu	64
4.5.4	cs.stanford.edu	66
4.6	Considerazioni finali sui risultati	67
5	Conclusioni e sviluppi futuri	110

Elenco delle tabelle

1.1	Matrice di co-occorrenze parola per parola [31]	9
4.1	Descrizione dei siti Web	53
4.2	Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Illinois	69
4.3	Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Illinois	70
4.4	Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Illinois	71
4.5	Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Illinois	72
4.6	Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Illinois	73
4.7	Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Illinois	74
4.8	Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Illinois	75
4.9	Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Illinois	76
4.10	Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Illinois	77
4.11	Risultati di Doc2Vec e TF-IDF in Illinois	77
4.12	Risultati della configurazione Combinato in Illinois	78

4.13	Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Oxford	79
4.14	Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Oxford	80
4.15	Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Oxford	81
4.16	Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Oxford	82
4.17	Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Oxford	83
4.18	Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Oxford	84
4.19	Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Oxford	85
4.20	Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Oxford	86
4.21	Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Oxford	87
4.22	Risultati di Doc2Vec e TF-IDF in Oxford	87
4.23	Risultati della configurazione Combinato in Oxford	88
4.24	Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Priceton	89
4.25	Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Priceton	90
4.26	Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Priceton	91
4.27	Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Priceton	92
4.28	Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Priceton	93
4.29	Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Priceton	94

4.30 Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Priceton	95
4.31 Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Priceton	96
4.32 Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Priceton	97
4.33 Risultati di Doc2Vec e TF-IDF in Priceton	97
4.34 Risultati della configurazione Combinato in Priceton	98
4.35 Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Stanford	99
4.36 Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Stanford	100
4.37 Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Stanford	101
4.38 Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Stanford	102
4.39 Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Stanford	103
4.40 Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Stanford	104
4.41 Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Stanford	105
4.42 Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Stanford	106
4.43 Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Stanford	107
4.44 Risultati di Doc2Vec e TF-IDF in Stanford	107
4.45 Risultati della configurazione Combinato in Stanford	108
4.46 Risultati finali di Illinois	108
4.47 Risultati finali di Oxford	108
4.48 Risultati finali di Priceton	109
4.49 Risultati finali di Stanford	109

Elenco delle figure

1.1	Categorie di Web Mining	5
1.2	Esempi di spazi di parole, rispettivamente mono-dimensionale (1) e bi-dimensionale (2)	8
1.3	Frase di esempio per la matrice di co-occorrenze [31]	9
1.4	Esempio di grafico di dispersione	12
1.5	Esempio di spazio degli embeddings	13
1.6	I modelli di apprendimento di Word2Vec	14
1.7	Un esempio di grafo/network di informazioni [36]	18
1.8	Prossimità di tipo Single-link	21
1.9	Prossimità di tipo Average-link	22
1.10	Prossimità di tipo Complete-link	22
1.11	Distanza tra centroidi	23
1.12	Esempio di Clustering utilizzando K-Means	24
1.13	Principio su cui si basa DBScan	25
3.1	Architettura di un web Crawler	35
3.2	Esempio di liste Web	37
3.3	Esempio di otto Random Walks in una dimensione	39
3.4	Esempio di Random Walks in due dimensioni	40
3.5	Esempio di Random Walks in tre dimensioni	40

Introduzione

L'Informatica è diventata uno dei pilastri fondamentali su cui si regge la società moderna, ovvero quella dell'informazione. Ogni giorno vengono prodotti un gran numero di dati liberamente fruibili da chiunque, che vengono archiviati e organizzati in maniera automatica ed economica. Produrre nuova conoscenza significa eseguire un processo di elaborazione dei dati che possa arricchire il sapere pregresso. Aumentare il bagaglio della conoscenza significa, quindi, facilitare i processi decisionali.

Questo problema è particolarmente sentito nel World Wide Web. Il Web è la più grande, eterogenea e dinamica sorgente di informazione liberamente fruibile da chiunque. Queste caratteristiche, però, rendono il processo di elaborazione dei dati e di produzione di nuova conoscenza una sfida impegnativa.

Sorge, quindi, una nuova problematica: accedere all'enorme mole di dati archiviati nel Web in maniera veloce e mirata. Una possibile soluzione consiste nell'utilizzo di tecniche di Clustering su pagine Web, ossia assegnarle a gruppi in cui si trovano elementi appartenenti alla stessa tipologia semantica (per esempio pagine di professori, corsi e prodotti).

Il Clustering di pagine Web non si propone come una nuova metodologia: in letteratura, infatti, esistono algoritmi che sfruttano o la struttura organizzativa di un sito Web o il testo contenuto nelle pagine, considerandole come documenti indipendenti tra loro.

Ma nel contesto del Web, le pagine non possono essere trattate come docu-

menti a se stanti, bisognerebbe piuttosto cercare di utilizzare l'informazione codificata nella struttura ad hyperlink del sito.

La metodologia descritta in questa tesi non considera, infatti, un sito Web come una collezione di documenti testuali indipendenti tra loro, ma cerca di combinare informazioni relative al contenuto con quelle codificate nella struttura ad hyperlink, in modo che due pagine Web vengano considerate simili se caratterizzate da una simile distribuzione di termini e abbiano una relazione di tipo diretta o indiretta, diretta se un hyperlink porta immediatamente ad una pagina, indiretta se vi sono pagine intermedie tra quella di partenza e di destinazione.

Si definisce di seguito la struttura di questo lavoro di tesi.

Nel capitolo 1 ci si occuperà di descrivere i concetti essenziali per comprendere a pieno il campo applicativo in cui verrà effettuata la sperimentazione. Nel capitolo 2 si parlerà della frontiera attuale dell'Informatica in questo campo, descrivendo i risultati dei lavori correlati a quello descritto in questa tesi. Nel capitolo 3 verranno descritti i passi eseguiti per questa sperimentazione, spiegandoli concettualmente e riportando gli algoritmi in pseudocodice. Nel capitolo 4 si descriverà la sperimentazione effettuata, spiegando le metriche utilizzate, le configurazioni utilizzate e riportando le tabelle con i risultati utili per spiegare i valori ottenuti. Nel capitolo 5, infine, si parlerà dei possibili sviluppi futuri di questa attività di ricerca.

Capitolo 1

Informazioni latenti nel Web

Siamo sommersi dai dati.

Ogni giorno viene generata una quantità enorme di dati dalle applicazioni per smartphone, dalle carte di credito usate per gli acquisti, dai programmi eseguiti sui computer, dai sensori utilizzati nelle infrastrutture intelligenti della città.

Le grandi quantità di dati sopra citate vengono chiamate Big Data.

In generale, con il termine Big Data si intende una collezione di dati talmente estesa in termini di volume, velocità e varietà da richiedere metodologie e tecnologie non convenzionali (i.e. nuove) per la loro memorizzazione, gestione, interrogazione ed analisi. Queste grandi raccolte di dati sono caratterizzate da tre aspetti importanti, chiamati anche **3V**:

- *Volume*. Rappresenta la quantità dei Big Data. Ogni giorno vengono prodotti dati nell'ordine dei Terabytes e dei Petabytes, che devono essere salvati oppure processati e consumati in tempo reale. Entrambi sono casi problematici: se i Big Data devono essere salvati in qualche base di dati il problema è nel salvataggio; se, invece, devono essere consumati immediatamente, il problema risiede nella loro analisi massiva.
- *Velocità*. Rappresenta il tempo per la generazione dei dati. Signifi-

ca sia quanto velocemente questi dati sono stati prodotti, sia quanto velocemente i dati devono essere processati per soddisfare un qualche obiettivo o domanda. Per gestire questa caratteristica bisogna capire se i dati catturati devono essere salvati in una base di dati o direttamente processati: nel primo caso bisogna utilizzare un database che permetta di effettuare le operazioni di fetching e di inserimento ad alte velocità; nel secondo caso, invece, bisogna utilizzare un'infrastruttura che sia capace di gestire le massive operazioni che devono essere effettuate sui Big Data.

- *Varietà.* Rappresenta la tipologia dei dati, che provengono da fonti diverse: strutturate e non strutturate. Con dati strutturati si intendono dati che sono organizzati secondo schemi rigidi, che vengono generalmente salvati in basi di dati; con dati non strutturati, invece, sono dati non schematizzati, che vengono generalmente salvati in file. La varietà dei Big Data è data dalla loro non strutturazione: blog post e commenti sui social network ne sono qualche esempio.

La diffusione e la produzione dei Big Data è resa possibile grazie al Web. Tutte le persone che hanno un accesso al Web generano una quantità enorme di dati da azioni che essi compiono, come effettuare pagamenti online, commentare o postare uno stato su Facebook o su Twitter, o ancora i click che essi effettuano durante la navigazione tra i siti. L'enorme mole eterogenea di dati viene utilizzata da aziende ed organizzazioni per estrarre nuova conoscenza, che viene usata per estendere quella preesistente e facilitare i processi decisionali.

1.1 Web Mining

Il Web possiede numerose caratteristiche che rendono l'estrazione di informazioni utili un problema impegnativo. Queste proprietà sono:

- **Dimensione.** Il Web è il primo mezzo in cui il numero di produttori di informazioni è uguale al numero di consumatori. La quantità di dati e di informazioni sul Web è enorme ed in continua crescita.
- **Dinamicità.** Ogni secondo vengono create, distrutte e modificate migliaia di pagine. Queste azioni rendono il Web una rete informativa dinamica, in cui il contenuto e la struttura cambiano con frequenza. Tenere traccia, quindi, di questi cambiamenti e monitorarli rimane una sfida impegnativa per molte applicazioni.
- **Eterogeneità.** Il Web è eterogeneo, e tale caratteristica dipende fortemente sia dal formato delle pagine sia dal contenuto testuale.
Nel primo caso, l'eterogeneità è definita dal fatto che non esiste uno standard di formato, dividendo le pagine Web in 3 tipologie: *pagine non strutturate*, *pagine strutturate* e *pagine semi-strutturate*.
Le pagine *non strutturate* sono scritte in linguaggio naturale, non sono caratterizzate da nessuna struttura e possono essere applicate tecniche di estrazione dell'informazione con un certo grado di affidabilità.
Le pagine *strutturate* vengono generate normalmente da una sorgente dati di tipo strutturato (e.g. database): i dati vengono pubblicati una volta che vengono inseriti in una qualche struttura (e.g. forma tabellare). In questo caso, l'estrazione della conoscenza viene effettuata attraverso l'individuazione di regole sintattiche.
Le pagine *semi-strutturate* sono una via di mezzo delle tipologie descritte in precedenza: sono caratterizzate dalla presenza sia di sezioni strutturate che da testo libero. L'estrazione della conoscenza viene effettuata cercando dei pattern nei tag HTML, utilizzando i metadati o identificando solo l'informazione strutturata.
Nel secondo caso, l'eterogeneità del Web è definita dal fatto che le pagine vengono create da milioni di persone aventi differente cultura, abilità e linguaggio. Da questo si deduce che le pagine Web possono avere informazioni simili o uguali, ma presentata in maniera completamente differente.
- **Connessione.** Il Web viene generalmente rappresentato come una rete

di informazioni, in cui i nodi sono le pagine Web e gli archi gli hyperlink o collegamenti ipertestuali. Questi collegamenti hanno caratteristiche e funzionalità differenti in base al loro utilizzo, che può essere sia per connettere pagine di uno stesso sito, sia pagine di siti differenti. All'interno del sito, i link servono per organizzare i contenuti; fra siti diversi, invece, vengono usati per collegare argomenti simili o inerenti a quelli della pagina di partenza.

- **Rumore.** Il Web, a differenza da altri mezzi di informazione, ha la caratteristica di permettere a chiunque di pubblicare contenuti senza alcun tipo di approvazione. Questo permette al Web di espandersi enormemente e di arricchire e diversificare le informazioni, ma contribuisce anche alla creazione e diffusione di contenuti di bassa qualità, rindondanti ed erronei.
- **Società virtuale.** Il Web può essere considerato come un grande Social Network, dove le persone possono diffondere la loro conoscenza ed influenzarsi reciprocamente. Infatti non riguarda solo i dati, le informazioni o i servizi, ma anche le interazioni fra persone, organizzazioni o sistemi automatizzati.

Le caratteristiche sopra citate evidenziano la necessità di definire un processo per scoprire informazioni utili partendo dai dati del Web. Tale processo viene chiamato Web Mining ed ha l'obiettivo di estrarre nuova conoscenza dalla struttura ad hyperlink del Web, dal contenuto delle pagine e dalla navigazione che gli utenti effettuano nel Web.

Il Web Mining non utilizza solo il processo KDD (Knowledge Discovery in Databases) per estrarre nuova conoscenza da basi di dati, ma si avvale anche del Text Mining (ovvero una branca dell'Informatica che ha come scopo quello di estrarre informazioni utili dai testi), Machine Learning, Network Analysis, Information Retrieval. Spesso le metodologie dei campi precedentemente citati vengono combinate per estrarre informazioni utili, date le caratteristiche del Web.

L'obiettivo del Web Mining è scomponibile nei seguenti sotto-obiettivi [6]:

- **Scoprire le risorse:** gli strumenti per la scoperta di documenti e servizi nella rete, che vengono chiamati Spider, ovvero Web Robot, scandiscono milioni di documenti Web e costruiscono indici di ricerca in base alle parole che si trovano negli stessi.
- **Estrarre le informazioni:** i testi, che sono scritti in linguaggio naturale, vengono trasformati in rappresentazioni strutturate predefinite, dette template, che rappresentano un estratto dell'informazione presente nel testo.
- **Generalizzare:** i processi di navigazione nel Web devono essere generalizzati, ovvero applicabili in altri contesti.

Gli algoritmi di Web Mining possono essere classificati in tre principali categorie, basate sul tipo di dato usato per estrarre nuova conoscenza: Web Structure Mining, Web Content Mining e Web Usage Mining.

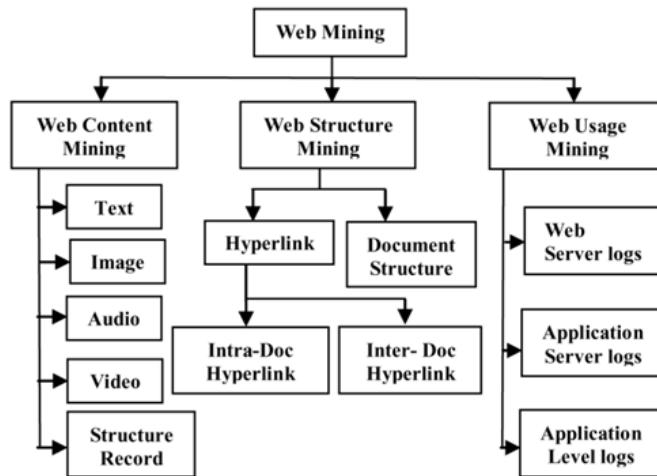


Figura 1.1: Categorie di Web Mining

Web Structure Mining Il Web Structure Mining è un processo di estrazione di informazioni utili partendo dalla struttura ad hyperlink di un sito Web, che viene considerato come un grafo, i cui nodi sono le pagine e gli archi sono gli hyperlink tra le pagine.

Basata sulla topografia degli hyperlink, il Web Structure Mining può categorizzare le pagine web e generare informazioni come la similarità e le relazioni tra i differenti siti Web [28].

Tecniche tradizionali di Data Mining non possono generare conoscenza utile perché non è presente una struttura a link in una tabella relazionale (i.e. database).

Tra i più importanti algoritmi che appartengono a questa tipologia si possono trovare Page Rank [27] e HITS [20], i quali sfruttano la struttura ad hyperlink del Web per assegnare un rank alle pagine, ovvero per restituirle in ordine di importanza relativamente ad una determinata query.

Web Content Mining Il Web Content Mining viene usato per cercare informazioni utili dai contenuti delle pagine Web, che possono essere collezioni di testi, immagini, audio, video o dati strutturati incapsulati in liste e tavole.

Per estrarre il sapere da contenuti più complessi, come le immagini, le tecniche di Web Content Mining sono molto limitate [34].

Le tecniche di questa tipologia di Web Mining possono sembrare abbastanza simili alle metodologie tradizionali di Data Mining o di Text Mining, ma le caratteristiche delle pagine Web (e.g. presenza di tag HTML) non permette a queste di essere direttamente applicabili sulle stesse.

Web Usage Mining Il Web Usage Mining è l'applicazione delle tecniche di Data Mining per la scoperta di pattern e informazioni utili attraverso l'analisi dei log, che sono immagazzinati nei Web server o nei sistemi che tracciano le attività degli utenti.

L'obiettivo di questo campo è la profilazione dell'utente, ovvero analizzare i suoi comportamenti sul web, sia per comprendere quali sono i suoi reali bisogni, sia per offrire dei servizi che possano soddisfare tali necessità e personalizzare l'esperienza Web.

Questo tipo di Web Mining viene usato in campi disparati, che vanno dalle aziende alle agenzie governative: ad esempio, i siti di e-commerce usano que-

sto tipo di tecnologia per presentare all’utente prodotti per i quali potrebbe essere interessato; le agenzie governative, invece, usano il Web Mining anche per classificare minacce e attentati terroristici.

Alcuni, però, criticano questa tecnologia: il problema etico di cui più si parla è la violazione della privacy, con il rischio di vedere diffusi i propri dati, anche sensibili, senza alcuna consapevolezza da parte dell’utente [37].

1.2 Rappresentazioni di pagine Web

Un sito è formato da pagine Web. Queste sono caratterizzate da numerose rappresentazioni, che sono:

- **Rappresentazione testuale.** Il testo è una componente fondamentale di una pagina Web, poiché ha come scopo il trasferimento dell’informazione. Durante la navigazione, infatti, l’utente estrapola conoscenza semplicemente leggendo il contenuto testuale delle pagine che visita.
- **Rappresentazione visuale.** Quando una pagina Web viene renderizzata da un browser, viene applicato uno stile di formattazione visuale, chiamato CSS, che definisce gli elementi della pagina Web come contenitori rettangolari che sono disposti o uno dopo l’altro o annidati tra loro formando un albero chiamato `Rendered Box Tree`. L’albero in questione è differente dalla struttura della pagina definita dai tag HTML, poiché una pagina può essere ricca di elementi invisibili, come il tag `<head>` o da elementi aventi come stile `display : none`. Inoltre, la generazione del rendered box tree richiede l’esecuzione di codice JavaScript e CSS.
- **Rappresentazione strutturale.** Questa è composta da elementi Web inscritti in tag HTML ed organizzata ad albero. I tag HTML possono essere applicati a porzioni di testo, hyperlink e dati multimediali per fornire loro un significato ed una renderizzazione differente della pagina da parte del browser.

In particolare, trattando una pagina Web come un documento testuale, è possibile produrre una rappresentazione vettoriale mediante algoritmi di Word space model, Vector space model, Word embedding (tra cui Word2Vec) per effettuare l'apprendimento. Non solo, trattando tale testo come un paragrafo, è possibile applicare un altro algoritmo di Word embedding, ovvero Doc2Vec.

Ma una pagina Web può anche essere trattata come un nodo del sito, il quale non è altro che un grafo, in cui i nodi sono le pagine del sito stesso e gli archi sono gli hyperlink tra i nodi. È possibile, quindi, produrre una rappresentazione vettoriale delle pagine basandosi sulla struttura ad hyperlink del sito. Un esempio è LINE.

Tutti gli algoritmi e le metodologie appena citate verranno descritte nelle sezioni successive.

1.2.1 Word space model

Il Word Space Model, come definito in [31], è una rappresentazione spaziale del significato delle parole. Si basa sul fatto che la similarità semantica viene rappresentata come prossimità, in uno spazio ad n dimensioni, dove n è un intero.

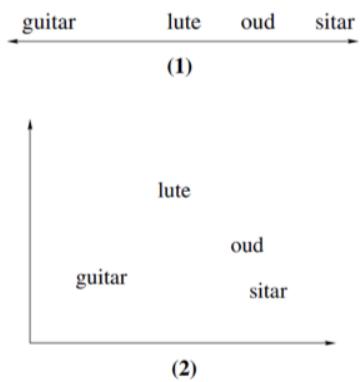


Figura 1.2: Esempi di spazi di parole, rispettivamente mono-dimensionale (1) e bi-dimensionale (2)

In Figura 1.2 sono riportati due esempi di spazi di parole, in cui la prossimità è data dalla posizione delle parole nello spazio. In entrambi i casi, si

può notare come il termine *sitar* sia più simile di significato a *oud*, e meno simile a *guitar*.

Ma come si costruisce uno spazio delle parole? Una modalità di costruzione è la **matrice di co-occorrenze**. Tale matrice può essere formata sia parola per parola ($w \times w$), dove w sono i tipi di parole nel set, sia parola per documento ($w \times d$), dove d sono i documenti nel set. Le celle di questa matrice registrano la frequenza di occorrenza della parola i -esima in un contesto j -esimo, oppure in un documento j -esimo nel caso di matrice parola per documento.

Per spiegare la matrice di co-occorrenze prendiamo come esempio la frase:

Whereof one cannot speak thereof one must be silent

Figura 1.3: Frase di esempio per la matrice di co-occorrenze [31]

Tabella 1.1: Matrice di co-occorrenze parola per parola [31]

	whereof	one	cannot	speak	thereof	must	be	silent
whereof	0	1	0	0	0	0	0	0
one	1	0	1	0	1	1	0	0
cannot	0	1	0	1	0	0	0	0
speak	0	0	1	0	1	0	0	0
thereof	0	1	0	1	0	1	0	0
must	0	1	0	0	1	0	1	0
be	0	0	0	0	0	1	0	1
silent	0	0	0	0	0	0	1	0

Le celle della matrice di co-occorrenze, visibile nella Tabella 1.1, registrano le occorrenze delle parola i -esima, le quali dipendono dal contesto. Con il termine *contesto* si intende un insieme di parole che si trovano nelle vicinanze della parola i -esima.

Queste liste di occorrenze sono dei veri e propri vettori. Un vettore, per definizione, è un elemento di uno spazio vettoriale, ed è definito da n componenti o coordinate $\vec{v} = (x_1, x_2, \dots, x_n)$ [31]. Tali coordinate definiscono la posizione nello spazio n-dimensionale.

Quindi, la matrice di co-occorrenze non è altro che una realizzazione del modello dello spazio vettoriale, chiamato Vector Space Model.

Vector Space Model Il Vector Space Model è una modalità algebrica per rappresentare documenti di testo come vettori, inseriti in uno spazio vettoriale.

Generalmente, per definire i vettori viene usato come peso *tf-idf*, una funzione che viene usata per misurare l'importanza di un termine rispetto ad un documento o ad una collezione di documenti. Come suggerisce il nome, è composta da due fattori: *tf* e *idf*.

Tf, abbreviazione di *term frequency*, misura quante volte un termine appare in un documento. Dato che ogni documento ha lunghezza differente, ovvero è composto da un differente numero di parole, è possibile che un termine possa apparire molte più volte nei documenti più lunghi rispetto a quelli più corti. Questo problema viene risolto dividendo la frequenza dei termini per la lunghezza del documento. La formula è:

$$tf_{i,j} = \frac{n_{i,j}}{|d_j|} \quad (1.1)$$

dove $n_{i,j}$ è il numero di occorrenze del termine t_i che si trova nel documento d_j , mentre il denominatore è la dimensione del documento d_j .

Idf, abbreviazione di *inverse document frequency*, misura invece i termini che si presentano più volte in un documento, ma con meno frequenza in tutta la collezione di documenti. Questo perché potrebbero esserci dei termini più significativi che appaiono raramente in un determinato elaborato, ma frequentemente nel set di documenti. La formula è:

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|} \quad (1.2)$$

dove $|D|$ è il numero di documenti presenti nella collezione, mentre il denominatore è il numero di documenti che contengono il termine t_i .

Il modello di spazio vettoriale sopra descritto, però, presenta particolari pro-

blematiche quando si vuole costruire uno spazio delle parole: da un lato, se non si hanno sufficienti dati, non sarà possibile costruire, in maniera fedele, un modello di distribuzione delle parole; dall'altro, se tale spazio vettoriale sarà dimensionalmente grande, innalzerà la complessità computazionale.

Secondo la sperimentazione portata avanti da [31], è stato riscontrato un altro problema: nella matrice di co-occorrenze circa il 99% delle celle conterrà zero come valore. Solo una parte delle parole apparirà realmente. Questo fenomeno è un esempio dell'applicazione della *Zipf's law*, una legge empirica che deriva dalla frequenza di un evento P_i , che fa parte di un insieme, in funzione della posizione i chiamata *rango* nell'ordinamento decrescente rispetto alla frequenza stessa di tale evento. La formula della legge è la seguente:

$$f(P_i) = \frac{c}{i} \quad (1.3)$$

dove i indica il rango, P_i indica l'evento che occupa l' i -esimo rango (ovvero l' i -esimo evento più frequente), $f(P_i)$ è il numero di volte (frequenza) che si verifica l'evento P_i , c è una costante di normalizzazione, pari al valore $f(P_1)$.

Una possibile soluzione alle situazioni problematiche sopra descritte è la riduzione della dimensione (*dimensional reduction*): un processo di compressione di uno spazio multi dimensionale ad uno avente bassa dimensione, causando anche la riduzione della grandezza della matrice e del numero di zero ivi inseriti.

Una tecnica di riduzione della dimensione più utilizzata è *t-SNE*, che è particolarmente adatta per comprimere uno spazio vettoriale di grandi dimensioni in uno avente uno, due o tre dimensioni, per poi essere visualizzato in un grafico di dispersione. Questo tipo di grafico è un sistema gli assi cartesiani di uno, due o tre dimensioni. Una volta effettuata la riduzione, i valori ottenuti per ciascun elemento del set vengono usati per posizionarlo nello spazio.

Word embedding La funzione peso *tf-idf* non è la sola usata per costruire uno spazio dei vettori: nelle tecniche di Data Mining, che sfruttano gli algoritmi di Machine Learning, ci si sta orientando sempre più sull'uso delle

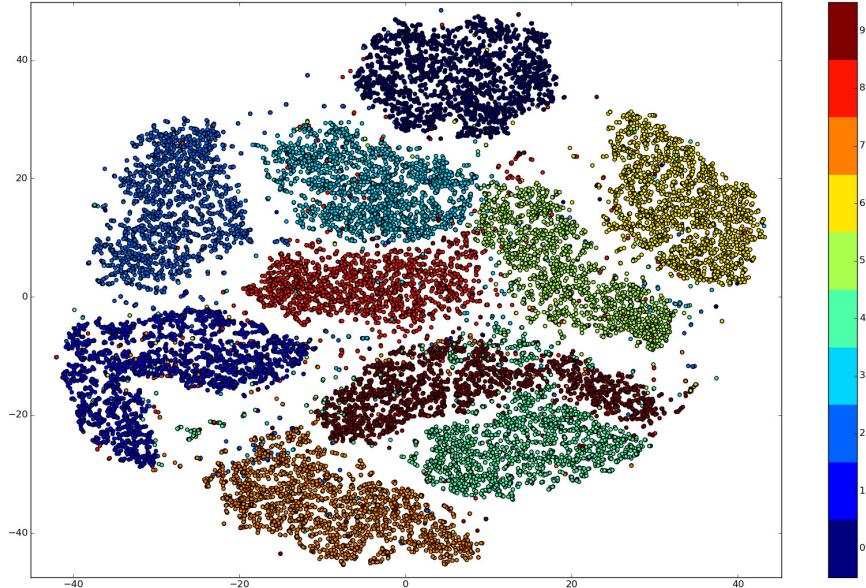


Figura 1.4: Esempio di grafico di dispersione

reti neurali per estrarre conoscenza partendo da una collezione di dati. Questo è il Word Embedding, nome di una serie di tecniche per il language modeling e per il Feature Learning nel campo del Natural Language Processing [3], in cui ad ogni parola viene associato un vettore chiamato *Feature Vector*.

Il Word Embedding è una funzione parametrizzata

$$W : \text{words} \rightarrow \mathbb{R}^k \quad (1.4)$$

che trasforma le parole di un dato linguaggio in un vettore multidimensionale. Per esempio:

$$W(\text{"mat"}) = (0.0, 0.6, -0.1, \dots) \quad (1.5)$$

Partendo da un documento, è possibile trasformare i termini in vettori, formando un vero e proprio spazio vettoriale, chiamato anche Word Embeddings Space.

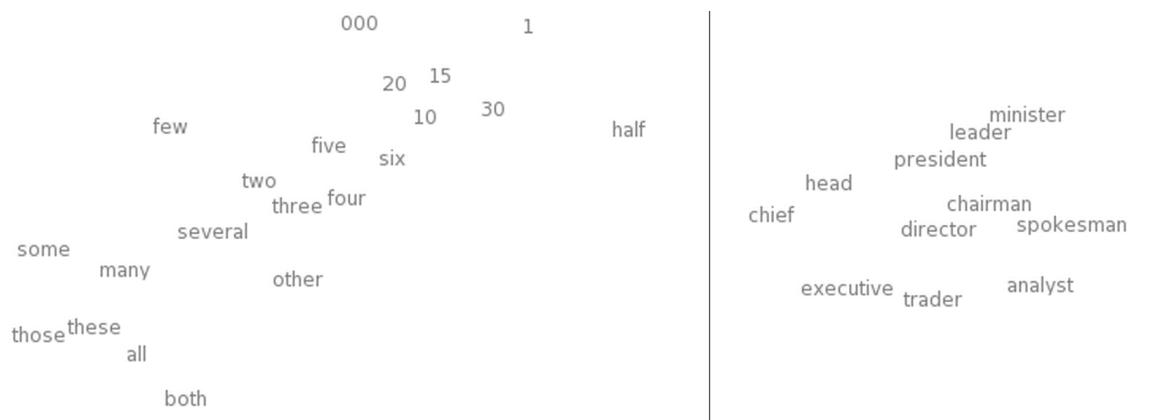


Figura 1.5: Esempio di spazio degli embeddings

In Figura 1.5 è possibile notare come parole simili si trovano vicine tra loro: la parola *three* è molto vicina alle parole *two* e *four*. Questo è dovuto al fatto che tali parole hanno vettori simili. Infatti, se si usa un sinonimo, la validità della frase non cambia (per esempio: "poche persone cantano bene" → "un paio di persone cantano bene"), perché le parole "poche" e "paio" sono vicine tra loro ed hanno vettori simili.

Un'altra proprietà interessante della funzione di Word Embedding è l'analogia tra le parole, nascosta nella differenza dei loro vettori:

$$W("woman") - W("man") \simeq W("queen") - W("king") \quad (1.6)$$

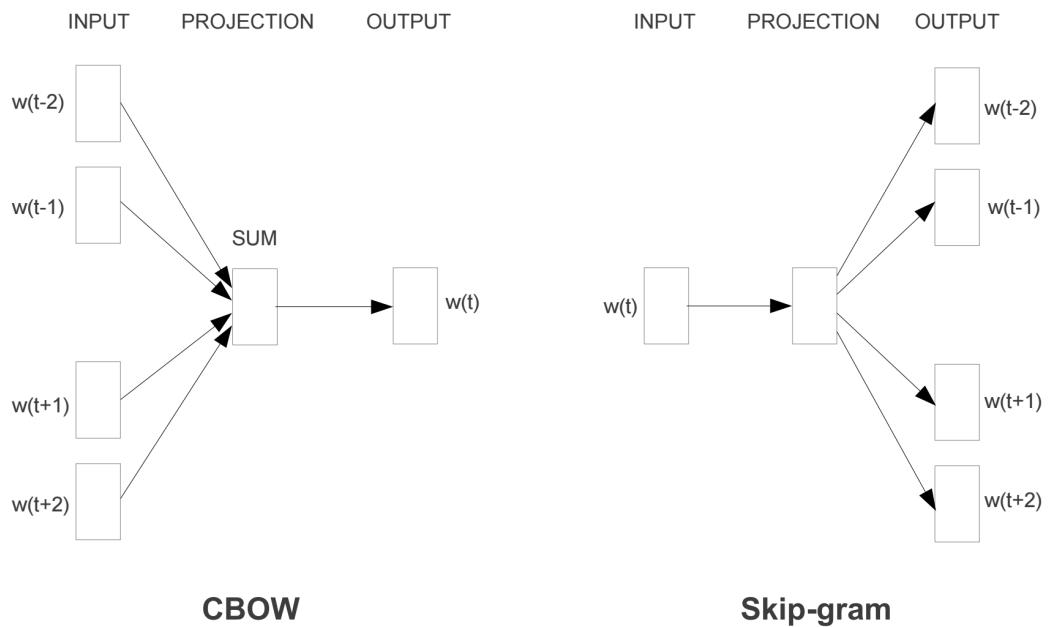
Da questo si evince che c'è una correlazione tra parole che hanno genere opposto, in quanto appariranno in contesti simili, differenti solo per alcuni dettagli come pronomi o articoli. Lo stesso principio vale per parole singolari e plurali [24].

Apprendere dei termini e trasformarli in Feature Vectors rappresenta una base per effettuare operazioni di Data Mining, come per esempio il raggruppamento dei termini appresi in gruppi attraverso il Clustering, usando qualche funzione di similarità. Si approfondiranno tali funzioni nella Sezione 1.3.1.

1.2.2 Word2Vec

Word2Vec è un algoritmo di Word Embedding ed è una rete neurale a due livelli che apprende le parole da un testo in input, le quali vengono trasformate in vettori chiamati Feature Vectors. Viene considerato erroneamente come un deep-learning (apprendimento approfondito): in realtà si tratta di un apprendimento di tipo superficiale (shallow-learning).

L'output di questa rete neurale è un vocabolario in cui ogni termine ha un vettore, che può essere compreso da una rete di deep-learning o semplicemente interrogato per rilevare delle relazioni tra i termini.



CBOW

Skip-gram

Figura 1.6: I modelli di apprendimento di Word2Vec

Word2Vec è composto da due modelli di apprendimento: **CBOW** (continuous bag of words) e **Skip-Gram**.

CBOW consiste nel predire una determinata parola a partire dal suo contesto, che è composto dal numero di parole che vengono prese in considerazione durante l'apprendimento. Questo modello di apprendimento tratta l'intero

contesto come una sola osservazione. Generalmente, CBOW restituisce risultati più accurati con piccole collezioni di dati [23].

Skip-Gram, invece, è l'inverso di CBOW: predice il contesto a partire da una parola.

Questo modello di apprendimento tratta ogni coppia contesto-obiettivo come una nuova osservazione, rendendo i risultati più accurati quando si hanno grandi collezioni di dati [23].

L'obiettivo dell'apprendimento di questo modello è quello di trovare le rappresentazioni vettoriali delle parole utili per predire quelle circostanti in una frase o in un documento. Formalmente, data una sequenza di parole w_1, w_2, \dots, w_T viene costruito un vocabolario, i cui termini hanno un vettore con n dimensione generato casualmente. Skip-Gram deve massimizzare la probabilità media logaritmica:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \quad (1.7)$$

dove c è la dimensione della finestra di contesto, w_t è la parola in input e w_{t+j} è la parola in analisi del contesto.

Per capire meglio questo tipo di modello di apprendimento, analizziamo questa frase:

The quick brown fox jumped over the lazy dog.

Inizialmente, si crea il set di dati formato da coppie (contesto, parola), di cui il primo è una sequenza di parole che dipende dalla dimensione della finestra, mentre la parola è il termine che si sta esaminando. Quindi, se si ha una finestra di contesto di dimensione 1, il set di dati sarà:

([quick], the), ([the, brown], quick), ([quick, fox], brown), [...] → (the, quick), (quick, the), (quick, brown), (brown, quick), [...]

Supponiamo di trovarci al passo t , dove si trova il primo caso di apprendimento (the, quick). L'obiettivo è quello di predire *quick* da *the*, applicando

la formula 1.7 e massimizzando la probabilità media logaritmica per apprendere il vettore ottimale di *quick*. Così facendo le parole simili avranno vettori simili. Ad ogni passo vengono aggiornati i vettori delle parole precedentemente apprese. Questo processo di apprendimento viene ripetuto sull'intera collezione di dati.

1.2.3 Doc2Vec

Doc2Vec, chiamato anche Paragraph2Vec, è un'estensione di Word2Vec che apprende correlando etichette e parole, invece che parole con altre parole. Differentemente da Word2Vec, che converte una parola in un vettore, Doc2Vec aggrega tutte le parole di un paragrafo in un vettore.

Quindi, data una collezione di testi che possa essere divisa in n documenti, o paragrafi, ad ogni paragrafo è assegnato un vettore. Il processo di apprendimento è caratterizzato dallo spostamento della finestra delle parole di contesto attraverso ogni parola di ogni paragrafo, per ogni paragrafo [17].

L'idea di base è quella di apprendere vettori di paragrafi in maniera simile all'apprendimento dei vettori delle parole. Vengono create due matrici: una formata dai vettori dei paragrafi appresi e l'altra da quelli delle parole. Il vettore del paragrafo è condiviso per tutte le parole che si trovano nello stesso, ma non per gli altri paragrafi. Gli embedding dei vettori e dei paragrafi sono combinati durante l'apprendimento ed aggiornati mediante la concatenazione o la media. I vettori vengono appresi utilizzando coppie, composte dalla parola da predire e dal contesto del campione, contrassegnato da un identificativo del paragrafo [11].

1.2.4 LINE

LINE è un nuovo modello di apprendimento, capace di produrre rappresentazioni vettoriali (embedding) di vertici in una rete o grafo. Un grafo è una coppia ordinata $G = (V, E)$ di insiemi, con V insieme dei nodi ed E insieme degli archi.

Questo modello di apprendimento lavora bene soprattutto con grafi orientati, pesati e non-pesati. Viene richiesto come input un file contenente gli archi che costituiscono il grafo da apprendere. L'obiettivo è quello di rappresentare ogni vertice $v \in V$ in un vettore R^d applicando una funzione $f_G : V \rightarrow R^d$, dove $d << |V|$ [36]. Viene prodotto, infine, uno spazio vettoriale formato da rappresentazioni vettoriali dei singoli nodi del grafo appreso.

LINE preserva sia la prossimità di primo che di secondo ordine. Nel primo caso ottimizza la seguente funzione di perdita:

$$-\sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j) \quad (1.8)$$

Trovando un insieme $\{\vec{u}_i\}_{i=1 \dots |V|}$ che minimizza la funzione obiettivo 1.8, è possibile rappresentare ogni vertice in uno spazio d -dimensionale. Anche nel secondo caso, si cerca di ottimizzare la seguente funzione di perdita:

$$-\sum_{(i,j) \in E} w_{ij} \log p_2(v_j | v_i) \quad (1.9)$$

Trovando due insiemi $\{\vec{u}_i\}_{i=1 \dots |V|}$ e $\{\vec{u}'_i\}_{i=1 \dots |V|}$ che minimizzano la funzione obiettivo 1.9, è possibile rappresentare ogni vertice v_i con un vettore d -dimensionale \vec{u}_i .

Per spiegare la prossimità di primo e secondo ordine, analizziamo la Figura 1.7. I vertici 6 e 7 sono collegati da un arco avente un determinato peso: tale peso indica la prossimità di prim'ordine. Nel caso dei vertici 6 e 8, dato che non esiste un arco tra questi due, la prossimità di primo ordine è 0.

I vertici 5 e 6, invece, condividono molti vertici vicini: hanno un'alta prossimità di secondo ordine.

In altre parole, la prossimità di primo ordine implica la somiglianza di due nodi. Per esempio, persone che sono amici gli uni con gli altri in un social network tendono a condividere simili interessi; pagine che sono collegate le une alle altre nel World Wide Web tendono a riferirsi a simili argomenti. La tipologia di secondo ordine, invece, sono persone, in un social network, che condividono simili amici che tendono ad avere simili interessi; parole che co-

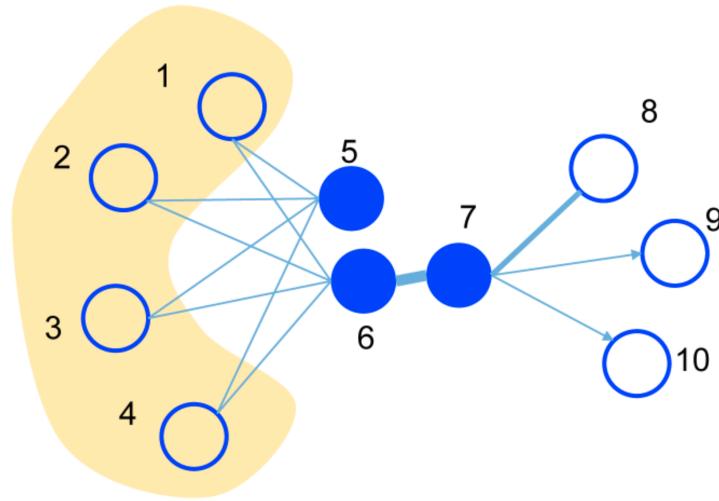


Figura 1.7: Un esempio di grafo/network di informazioni [36]

occorrono sempre con lo stesso insieme di termini tendono ad avere significati simili.

Può capitare che una coppia di nodi di un grafo possa non avere un arco, avendo come prossimità di primo ordine 0, anche essendo simili. Per questo la prossimità di primo ordine non è sufficiente per preservare le strutture del network, ed è importante considerare una nozione alternativa di prossimità che permetta di considerare i vertici come simili, se condividono simili nodi vicini (prossimità di secondo ordine). LINE, quindi, permette di apprendere sia le strutture locali (i.e. i diretti vicini) sia quelli globali (i.e. i vicini dei vicini) del network.

La situazione sopra descritta è riscontrabile anche in Word2Vec. Analizzare solo le parole immediatamente vicine al termine target non permette di considerare quelle simili che non si trovano vicine le une alle altre. Per questo è possibile modificare la dimensione della finestra di contesto per apprendere parole che non si trovano nell'immediato vicinato del termine target.

Rispetto a LINE, che permette al più di analizzare i vicini dei vicini di un dato vertice (quindi relazioni avente profondità 2), Word2Vec permette di analizzare relazioni più ampie, aventi profondità maggiori di 2.

1.3 Clustering

Le classi, insiemi di oggetti che condividono determinate caratteristiche, hanno un ruolo importante sia nell'analisi effettuata da persone, sia nella descrizione del mondo. Infatti, gli esseri umani hanno l'abilità di dividere gli oggetti in gruppi e di assegnarli a questi insiemi [35].

Questo processo di divisione prende il nome di Clustering ed ha come scopo quello di selezionare e raggruppare, da una collezione di dati, elementi omogenei, avendo come base la somiglianza tra gli stessi.

L'attività di raggruppamento può essere applicata in molti campi [35]:

- **Biologia.** Tecniche di Clustering sono state applicate per analizzare informazioni genetiche e creare una tassonomia di tutte le specie viventi.
- **Clima.** Il Clustering è stato utilizzato per trovare degli schemi nel clima della Terra.
- **Psicologia e Medicina.** Le operazioni di Clustering hanno identificato variazioni di malattie e depressioni.
- **Business.** Il Clustering viene usato per dividere i consumatori in gruppi per successive attività di analisi e di marketing.
- **Information Retrieval.** Il World Wide Web è un enorme contenitore di pagine Web, ed una semplice ricerca può dare come risultato milioni di pagine. Le tecniche di Clustering vengono usate per dividerle in gruppi, ognuno dei quali cattura un aspetto particolare dell'interrogazione. Quindi, se si cerca, con un motore di ricerca, la parola *film*, verranno restituite pagine Web raggruppate in categorie come recensioni, trailer, celebrità, teatri e cinema. Ogni categoria (cluster) può essere diviso in sottogruppi (sottocluster) e si produrrà una struttura gerarchica che servirà all'utente per effettuare altre ricerche.

Quindi, l'operazione di Clustering è essenzialmente la creazione di un insieme di Cluster (i.e. un insieme di insiemi), che generalmente contengono tutti gli

elementi iniziali.

In questa tesi sono stati utilizzati algoritmi di Clustering che si basano sulla rappresentazione vettoriale degli elementi (i.e. pagine di un sito Web) per poter raggrupparli in insiemi.

Esistono vari approcci di Clustering per raggruppare elementi in insiemi. Alcuni di questi sono:

- Hard Clustering o Soft Clustering
- Partizionali o gerarchici

Hard Clustering e Soft Clustering Questi algoritmi attuano un approccio secondo cui un elemento può essere assegnato ad un solo Cluster o a più Cluster. Con Hard Clustering intendiamo che l'algoritmo assegna un elemento ad uno ed un solo Cluster; con Soft Clustering, invece, l'elemento può essere assegnato a più Cluster con gradi di appartenenza diversi.

Clustering partizionale Gli algoritmi di Clustering partizionali creano una divisione delle osservazioni minimizzando una certa funzione di costo:

$$\sum_{j=1}^k E(C_j) \quad (1.10)$$

dove k è il numero desiderato di Cluster, C_j è il j -esimo Cluster ed $E : C \rightarrow \mathbb{R}^+$ è la funzione di costo associata al singolo Cluster. L'algoritmo più famoso che fa parte di questa categoria è K-Means.

Clustering gerarchico Gli algoritmi che fanno parte di questa categoria non suddividono lo spazio, bensì costruiscono una gerarchia di Cluster. In questa strategia rientrano due sottotipi:

- **Aggregativo:** tale approccio considera n Cluster per n elementi, cioè ogni elemento viene considerato un Cluster a sé. Successivamente, l'algoritmo unisce tutti i Cluster più vicini. Viene anche chiamato bottom-up.

- **Divisivo:** tale approccio opera in maniera opposta rispetto al precedente, poichè tutti gli elementi vengono considerati come un unico Cluster e l'algoritmo deve dividere il Cluster in insiemi aventi dimensioni inferiori. Questa metodologia viene anche chiamata top-down.

Durante l'aggregazione degli elementi è necessario usare una funzione che permette di calcolare la similarità (o meglio la distanza) tra due Cluster: questo permette all'algoritmo di unire i Cluster simili.

1.3.1 Funzioni (o misure) di distanza

A seconda dell'approccio utilizzato, vi sono delle funzioni (o misure) che permettono di calcolare la distanza tra due Cluster. Viene molto usato dagli algoritmi di Clustering gerarchico per calcolare la similarità tra i Cluster e per unire, eventualmente, quelli simili.

Le funzioni di distanza usate da questo tipo di Clustering sono:

- **Single-link proximity.** Questa funzione calcola la distanza tra due Cluster come la distanza minima tra elementi appartenenti a Cluster differenti.

$$D(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \quad (1.11)$$

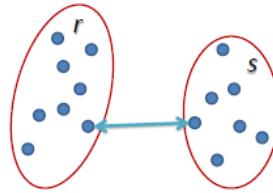


Figura 1.8: Prossimità di tipo Single-link

- **Average-link proximity.** Questa funzione calcola la distanza tra due

Cluster come la media delle distanze tra i singoli elementi.

$$D(C_i, C_j) = \frac{1}{(|C_i||C_j|)} \sum_{x \in C_i, y \in C_j} d(x, y) \quad (1.12)$$

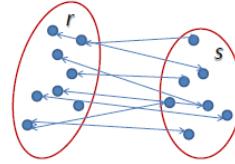


Figura 1.9: Prossimità di tipo Average-link

- **Complete-link proximity.** Questa funzione calcola la distanza tra i due Cluster, considerando la distanza massima tra gli elementi appartenenti ai due Cluster.

$$D(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \quad (1.13)$$

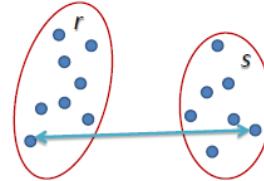


Figura 1.10: Prossimità di tipo Complete-link

- **Distanza tra centroidi.** Questa, invece, è la distanza tra i due Cluster prendendo in considerazione i centroidi degli stessi. Un centroide è un punto rappresentativo, prodotto dalla posizione media aritmetica della distanza tra tutti i punti.

$$D(C_i, C_j) = d(\hat{c}_i, \hat{c}_j) \quad (1.14)$$

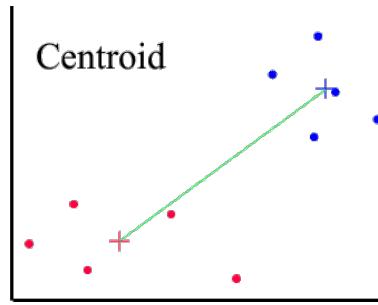


Figura 1.11: Distanza tra centroidi

Nei casi precedenti, $d(x, y)$ indica una qualsiasi funzione distanza, su uno spazio metrico. Le più importanti sono:

- **Distanza euclidea:** chiamata anche norma 2, è la distanza calcolata tra due punti, che può essere misurata su uno spazio multidimensionale. Siano $P = (p_1, p_2, \dots, p_n)$ e $Q = (q_1, q_2, \dots, q_n)$ due punti, la distanza sarà:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^n (p_k - q_k)^2} \quad (1.15)$$

- **Coseno di similarità:** tecnica euristica usata per misurare la distanza tra due vettori, che viene effettuata calcolando il coseno dell'angolo ivi compreso, che hanno l'origine coincidente con quello del sistema di assi e passano per i rispettivi elementi. Il valore risultante più sarà vicino ad 1, più i due elementi saranno simili tra loro. Siano A e B due vettori di attributi numerici, allora il coseno di similarità sarà calcolato mediante la formula:

$$\cos(\theta) = \frac{AB}{\|A\|\|B\|} \quad (1.16)$$

1.3.2 Algoritmi usati

In questa sezione vengono descritti gli algoritmi di Clustering usati in questa tesi, che sono stati applicati per raggruppare pagine di un sito Web. Questi

algoritmi si basano sulla rappresentazione vettoriale delle pagine (Sezione 1.2) per assegnare gli elementi ad un Cluster.

K-Means K-Means è un algoritmo di Clustering di tipo partizionale, in cui ogni Cluster viene identificato mediante un centroide.

Si basa sull'algoritmo di Lloyd e consiste in 3 step. Il primo step consiste nella scelta dei centroidi iniziali che saranno K elementi, casuali o usando informazioni euristiche, scelti dal dataset. Successivamente, l'algoritmo assegna per ogni elemento il centroide più vicino e ne crea di nuovi dalla media di tutti i campioni, assegnati ai centroidi precedenti. Si ripete questa fase finché l'algoritmo non converge.

Il pregio principale di questo algoritmo è la rapidità di convergenza: si è analizzato, infatti, che il numero di iterazioni che l'algoritmo esegue è minore del numero di elementi del dataset.

K-means, però, può essere molto lento nel caso peggiore e non garantisce il raggiungimento dell'ottimo globale: la bontà della soluzione dipende dal set di Cluster iniziale. Inoltre, un altro svantaggio è che l'algoritmo richiede, in input, il numero dei Cluster.

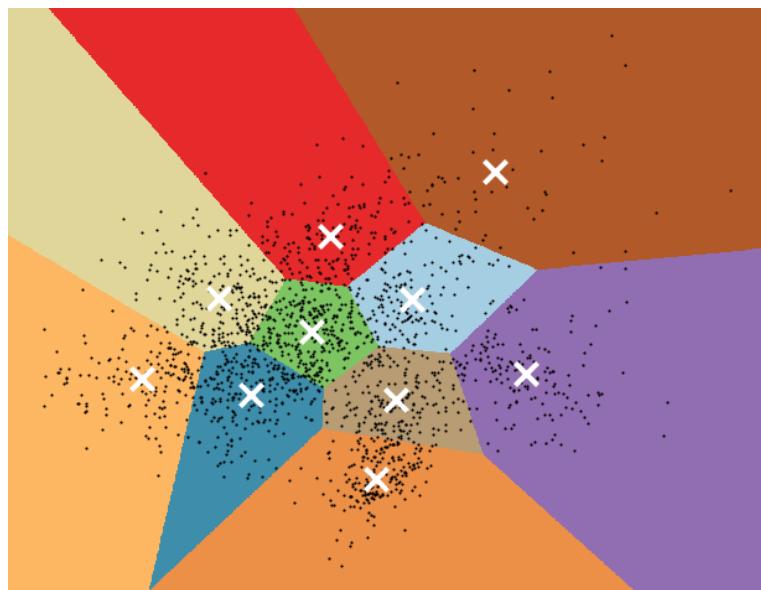


Figura 1.12: Esempio di Clustering utilizzando K-Means

HDBScan HDBScan è un algoritmo di Clustering che estende DBScan, rendendolo di tipo gerarchico.

DBScan necessita di due parametri: ϵ e del numero minimo di elementi richiesti per formare un Cluster (minPts). Si comincia con un punto casuale che non è stato ancora visitato. Viene calcolato il suo ϵ -vicinato, e se contiene un numero sufficiente di punti viene creato un nuovo raggruppamento. Se ciò non avviene, il punto viene etichettato come rumore, ovvero non viene associato ad alcun Cluster. È possibile che tale elemento rumore venga ritrovato in un ϵ -vicinato sufficientemente grande e, quindi, inserito in un altro Cluster. Se un punto è associato ad un raggruppamento, allora anche tutti quelli nel suo ϵ -vicinato sono parte del Cluster. Di conseguenza, tutti i punti trovati all'interno del suo ϵ -vicinato sono aggiunti al Cluster, così come i loro ϵ -vicini. Questo processo continua fino a quando il Cluster viene completato. Allora, un nuovo punto non visitato viene estratto e processato.

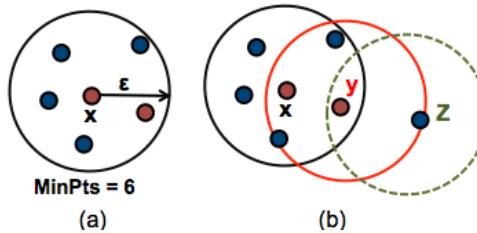


Figura 1.13: Principio su cui si basa DBScan

In HDBScan, invece, si parte in maniera simile a DBScan: lo spazio viene trasformato a seconda della densità e viene effettuato su di esso una prossimità a single-link. Invece di richiedere come input il parametro ϵ , che viene usato da DBScan per considerare gli elementi del vicinato appartenenti al Cluster, viene creato un albero, il quale viene usato per selezionare i Cluster più stabili e persistenti. Viene solo richiesta la dimensione minima dei Cluster per determinare quali gruppi non devono essere considerati come Cluster, oppure per dividerli e formare nuovi Cluster.

Questo algoritmo è molto efficace ed è più veloce sia di DBScan che di K-Means.

1.4 Obiettivi della tesi

Questa tesi ha due obiettivi:

1. Capire se, combinando diverse rappresentazioni di cui una pagina Web si compone in uno spazio vettoriale, al fine di applicare i tradizionali algoritmi di Clustering, vi è un miglioramento della qualità dei Cluster prodotti.
2. Verificare che, analizzando la topologia del grafo del sito Web e dando più importanza a pagine vicino alla homepage, si ottengono risultati migliori rispetto all'approccio che analizza tutte le pagine come ugualmente importanti. La modifica applicata verrà discussa nella Sezione 3.2.3.

Capitolo 2

Stato dell'Arte

L'applicazione delle tecniche di Clustering su pagine Web non è un nuovo campo di ricerca. In letteratura, molte metodologie sono state usate per raggruppare le pagine Web.

Tuttavia, queste ricerche sono state indirizzate sul Clustering di pagine provenienti da diversi siti Web, trascurando quelle di uno specifico sito. Gli hyperlink, infatti, hanno significati diversi in base al dominio di destinazione: se la pagina puntata si trova nello stesso sito Web, allora il collegamento avrà funzione di organizzazione dei contenuti. Altrimenti, se la pagina di destinazione è esterna, avrà la funzionalità di riferirsi a pagine che, probabilmente, avranno contenuti simili.

Gli algoritmi di Clustering esistenti si classificano in quattro categorie, in base alle informazioni usate per raggruppare le pagine Web:

- **Algoritmi di Clustering basati sul contenuto testuale.** Questa tipologia di algoritmi considerano le pagine come dei documenti testuali. Questo è il caso di [39, 5, 15, 1], dove la distribuzione delle parole è usata per scoprire insiemi appropriati di pagine Web correlate. Il vantaggio di questo approccio è che molti strumenti di Clustering, basati sul modello dello spazio vettoriale, possono essere direttamente applicabili. Lo svantaggio è che questi algoritmi falliscono quando devono

essere appresi modelli accurati, a causa della natura non controllata ed eterogenea dei contenuti delle pagine Web.

I tradizionali algoritmi di Clustering si basano sull'assunto che i documenti testuali condividono stili di scrittura consistenti, dando abbastanza informazioni contestuali, sono chiari e completamente non strutturati, sono indipendenti e identicamente distribuiti. Queste limitazioni sono più marcate per il Clustering di pagine Web di differenti siti. Infatti, le pagine aventi stesso argomento potrebbero essere contestualmente differenti: potrebbero avere un contenuto informativo simile inserito in elementi Web aventi diverse regole semantiche (i.e. tabelle o menu di navigazione) e differenti funzionalità (i.e. link, pulsanti, immagini).

- **Algoritmi di Clustering basati sui Web log.** In questa tipologia si trovano algoritmi che analizzano ed estraggono informazioni a partire dai Web Log, che vengono usati per raggruppare pagine Web in funzione a schemi di comportamento degli utenti durante la navigazione in un determinato sito. In [32] l'autore considera la cronologia di navigazione ed il tempo di visualizzazione di ogni pagina come informazioni per raggruppare i profili degli utenti. I Cluster estratti possono essere usati per migliorare l'esperienza di navigazione degli utenti [7]: così facendo, per esempio, è possibile migliorare la navigazione del sito Web di un dipartimento, differenziandola in base al profilo utente. Questa operazione di Clustering, tuttavia, può portare a difficoltà: ad ogni profilo utente potrebbero corrispondere Cluster differenti di pagine Web.
- **Algoritmi di Clustering basati sulla struttura HTML.** Le pagine Web, a differenza dei documenti testuali, sono caratterizzate da proprietà strutturali come i tag HTML, che permettono di definire la loro rappresentazione strutturale. E' stato provato da [8, 4, 22, 41] che l'informazione strutturale fornisce una differente e complementare rappresentazione rispetto a quella testuale. Questi tipi di algoritmi di Clustering hanno il vantaggio di considerare l'informazione strutturale e visuale inserita nei tag HTML, che viene ignorata dall'approccio te-

stuale. I tag HTML sono i responsabili della visualizzazione dei dati all'interno di una pagina Web. Per questo è possibile avere pagine aventi lo stesso tipo di semantica (e.g. pagine di professori) ma codificate e visualizzate in maniera differente. Per esempio, i dati strutturati inseriti in tabelle (aventi come tag HTML `<table>`) oppure in liste (aventi come tag HTML ``) avranno una simile visualizzazione. Questo abbassa la qualità dei Cluster generati.

Per risolvere questo limite, gli autori di [8, 4] propongono di usare l'informazione associata ai tag HTML. In [4] viene effettuato un processo di Clustering basandosi solo sulle proprietà visuali delle pagine Web. L'obiettivo di questo approccio è quello di raggruppare le pagine che hanno una visualizzazione simile, trascurando il contenuto e la struttura HTML. In [8] il layout e le proprietà visuali associati ai tag HTML vengono usati per caratterizzare la struttura dell'intera pagina Web, e le collezioni di hyperlink vengono considerate per trovare pagine aventi una rappresentazione strutturale simile.

- **Algoritmi di Clustering basati sulla struttura ad hyperlink.**
Anche la struttura ad hyperlink, che interconnettono le pagine Web, caratterizzano le loro proprietà strutturali. Gli algoritmi basati sulla struttura ad hyperlink lavorano su una interconnessa collezione di pagine Web. L'idea di base è che quando due pagine web sono connesse tramite un link, esiste una relazione tra le due. Da qui sarà possibile effettuare il Clustering. In generale, questi metodi [9] usano solo link diretti tra le pagine e, da questi, vengono usate/definite alcune misure di similarità, come la somiglianza bibliografica [19] e le co-citazioni [33].

In [12], l'autore spiega come questa tipologia di algoritmi di Clustering lavorano bene quando la struttura ad hyperlink è densa e senza link rumore. Essi restituiscono risultati di bassa qualità per pagine Web aventi un numero insufficiente di link, sia che portino a pagine appartenenti allo stesso sito Web, sia che puntino a pagine di siti differenti. Inoltre, non tutti i link hanno stessa importanza nel processo

di Clustering: le pagine sono spesso arricchite di link rumore come gli short-cut hyperlink (i link scorciatoia). Per superare questo problema, molti algoritmi combinano le informazioni ricavate dal contenuto con quelle dei link [22, 16, 25, 38, 10, 2].

Inoltre, in [16] l'autore ha risolto il problema dei link rumore considerando solo gli hyperlink tra pagine Web di argomento simile e co-citate. Successivamente, è stato applicato un tradizionale algoritmo di Clustering basato sul partizionamento del grafo. In particolare, è stato assegnato un peso che combina la similarità di contenuto e la co-citazione per ogni arco (i, j) , dove i e j sono le pagine collegate del grafo del sito Web. Il metodo ha due principali limitazioni: *i*) le informazioni testuali vengono usate solo per definire i pesi dei link, di conseguenza due pagine Web condivideranno le stesse proprietà distribuzionali, ma, avendo una similarità testuale bassa, non potranno essere inserite nello stesso cluster; *ii*) l'algoritmo di Clustering del grafo è NP-hard, ovvero è altamente complesso in termini di computazione.

In [22], l'autore propone una misura di similarità ottenuta combinando la somiglianza di tipo testuale, di co-citazione, di bibliografia e della struttura ad hyperlink che interconnettono pagine dello stesso sito Web. In questo modo, due pagine che hanno una similarità di struttura ad hyperlink appariranno più volte all'interno della collezione dei link. Le combinazioni delle varie similarità, ancora oggi, sono ancora un problema aperto.

2.1 Altre metodologie di Clustering

Gli algoritmi di Clustering basati sulla struttura ad hyperlink considerano solo relazioni dirette tra i vicini, senza analizzare la struttura globale del grafo del sito Web. In [14, 40, 36, 29] gli autori sostengono che la similarità tra i nodi di due grafi può essere rappresentata in termini di somiglianza dei loro rispettivi contesti. In altri termini, la similarità tra i nodi si basa su come questi condividono quelli circostanti, i quali non sono necessariamente

vicini immediati.

In [40] viene proposto un algoritmo di Clustering che si concentra sulla struttura topologica di un grafo e sulle proprietà dei nodi, che possono essere testuali o relazionali. Un insieme di nodi ed archi attributi vengono aggiunti al grafo originale. Così facendo, la similarità degli attributi è trasformata in base alla vicinanza dei vertici nei grafi: due vertici che condividono un attributo sono collegati da uno di tipo attributo. Nonostante l'algoritmo possa combinare sia le informazioni strutturali che di contenuto, usando una comune rappresentazione, non può essere utilizzato a dati che hanno valori numerici (e.g. tf-idf) o attributi categorici aventi un gran numero di valori distinti.

In [29, 36] vengono proposti due metodi di embedding, rispettivamente DeepWalk e LINE. Questi sfruttano le reti neurali per generare una rappresentazione vettoriale dei nodi del grafo. DeepWalk [29] applica il modello di apprendimento Skip-Gram sui Random Walks (Sezione 3.2.1) troncati per codificare le relazioni tra i nodi del grafo. Tale approccio non è capace di catturare la struttura locale del grafo (i.e. i nodi che possono essere considerati simili perché sono fortemente connessi). LINE [36] ottimizza la funzione obiettivo che incorpora sia le strutture locali che quelle globali della rete. Mentre DeepWalk è capace di considerare relazioni più ampie, LINE può apprendere solo relazioni aventi profondità 2 (i.e. vicini dei vicini). Una limitazione di entrambi i metodi è quella di ignorare i nodi attributo (e.g. contenuto testuale). Di conseguenza, l'operazione di Clustering basata su embedding può essere difficoltosa in grafi senza sufficienti hyperlink interni ed esterni, ma caratterizzati da ricchi contenuti testuali.

2.1.1 Combinazione di embeddings

Recenti sperimentazioni hanno esaminato vari approcci di generazione di embedding per capire i punti di forza e le debolezze di ognuno. Altre aree di ricerca del Machine Learning hanno scoperto che, combinando varie tecniche, è possibile ottenere risultati eccellenti.

In questo contesto, gli autori di [13] hanno esplorato varie tipologie di composizioni di embedding, dimostrando che i vari metodi di combinazione dei vettori degli embedding possono produrre spazi vettoriali ibridi che forniscano risultati aventi una bontà significativa. Nello specifico, hanno cercato di capire se si ottengono risultati migliori effettuando una semplice addizione o una concatenazione tra i vettori. Gli spazi vettoriali sono stati prodotti da due algoritmi differenti di Machine Learning:

- **DVRS.** E' un metodo per generare rappresentazioni vettoriali semantiche. Ogni parola è rappresentata da due vettori: un vettore ambientale fisso $e(i)$ viene generato casualmente assegnando ad ogni elemento del vettore un valore compreso tra $[-1, 1]$; il vettore lessicale $l(i)$ cattura, invece, il significato della parola, che viene aggiornato durante l'apprendimento. Una volta che il documento viene processato, il vettore lessicale di ogni parola viene aggiornato in base sia al contesto di paragrafo $c(k)$, sia al contesto di ordine della frase $o(k)$. Rispettivamente:

$$c(k) = \sum_{i=1}^n e(i), i \neq k \quad (2.1)$$

$$o(k) = \sum_{j=-4}^4 s(j) * e(k+j) \quad (2.2)$$

dove $j \neq 0$ e $0 < (k+j) \leq n$.

- **Word2Vec.** E' un algoritmo di Word Embedding che trasforma parole in Feature Vectors. Per approfondimenti, vedere la Sezione 1.2.2.

I risultati riportati da [13] mostrano come la concatenazione dei vettori di DVRS e di Word2Vec porta ad un incremento della bontà dei risultati.

Capitolo 3

Metodologia

Il Clustering delle pagine di un sito Web è un processo fondamentale nel Web Mining utile a valutare l'interazione tra le pagine, organizzare i contenuti del sito e capire come questo sia stato strutturato.

Una pagina Web è composta da varie rappresentazioni (Sezione 1.2) che la caratterizzano e la diversificano dalle altre. Attualmente, queste proprietà vengono sfruttate dagli algoritmi di Clustering in maniera indipendente.

Gli obiettivi di questa tesi, descritti nella Sezione 1.4, sono stati raggiunti mediante l'utilizzo di una metodologia composta da 3 passi:

- *Crawling del sito Web*
- *Costruzione del Dataset*
- *Clustering delle pagine Web*

Di seguito, si descrivono in dettaglio le fasi della metodologia.

3.1 Web Crawling

Le proprietà che caratterizzano le pagine Web rendono complicato il processo di estrazione di informazioni, soprattutto nel caso in cui i contenuti vengono generati dinamicamente. Per analizzare i contenuti della rete e delle pagine

Web si utilizza un software automatizzato chiamato Web Crawler. Tipicamente, questo programma viene usato per molti altri scopi, come l'indicizzazione di pagine Web. I motori di ricerca sfruttano tali indici per aumentare l'efficienza delle interrogazioni che gli utenti effettuano durante la navigazione.

La ragione principale dell'uso dei Crawler è che il World Wide Web non è un contenitore centralizzato: può essere visto, infatti, come un insieme di siti Web che forniscono differenti servizi.

L'algoritmo di Crawling è relativamente semplice: dato un insieme di URL di pagine Web, vengono scaricate tutte le pagine associate all'indirizzo URL, estratti gli hyperlinks e, iterativamente, scaricate le pagine associate a questi link. Il loro contenuto verrà analizzato e salvato per essere successivamente indicizzato. Nonostante l'apparente semplicità di questo algoritmo, l'attività di Web Crawling è caratterizzata da obiettivi inerenti [26]:

- **Scalabilità.** Il Web è molto grande e in continua evoluzione. I Crawler che cercano una copertura ampia ed aggiornata devono raggiungere throughput (prestazioni) estremamente alti. Per riuscirci, bisogna risolvere problematiche ingegneristiche complesse. I moderni motori di ricerca impiegano migliaia di computer e decine di collegamenti di rete ad alta velocità.
- **Compromessi per la selezione dei contenuti.** Persino i Crawler con un alto throughput non sono in grado di scandire l'intero Web o tenere traccia di tutti i cambiamenti. Per questo, il processo di Crawling viene effettuato in maniera selettiva con un attento e controllato ordine. Gli obiettivi sono di acquisire velocemente i contenuti aventi un alto valore, di assicurare la copertura di tutti i contenuti scelti, di ignorare quelli a bassa qualità, irrilevanti, ridondanti e dannosi e di mantenerli aggiornati. Il Crawler deve rispettare alcuni vincoli come il numero massimo di visite per sito e le pagine da scartare durante l'analisi.
- **Obblighi sociali.** I Crawler dovrebbero essere dei "buoni cittadini"

del Web: non devono sovraccaricare i siti che scandiscono. Infatti, senza i giusti meccanismi, un throughput alto può inavvertitamente provocare un attacco Denial of Service (DOS), sovraccaricando il server Web e rendendolo inaccessibile.

- **Avversari.** Alcuni siti Web cercano di iniettare contenuti inutili o fuorvianti nel corpus assemblato dai Crawler. Questo comportamento è spesso motivato da incentivi finanziari, per esempio per indirizzare male il traffico verso siti commerciali.

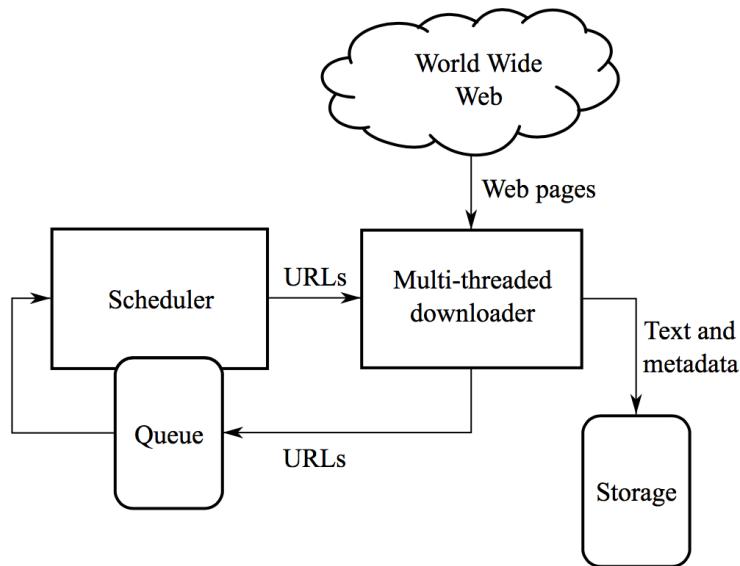


Figura 3.1: Architettura di un web Crawler

3.1.1 Crawling delle pagine di un sito Web

Ai fini sperimentali è stato utilizzato un Web Crawler che estrae gli hyperlink di un sito Web per effettuare successivamente l'operazione di Clustering delle pagine associate a tale sito. Un sito Web può essere formalmente descritto come un grafo orientato $G = (V, E)$, dove V è l'insieme delle pagine appartenenti al sito ed E è l'insieme degli hyperlink. In molti casi, la homepage h di un sito rappresenta il punto di inizio, tecnicamente espresso come un grafo orientato radicato (albero), che consente l'esplorazione da parte degli utenti.

Il grafo del sito Web può essere ricco di link rumore (e.g. hyperlink scorciatoia) che non sono rilevanti nel processo di Clustering [8] e vengono esclusi dal Crawling. In più, la struttura del sito Web è codificata in sistemi di navigazione che offrono una visuale della sua organizzazione. Questi sistemi vengono implementati come collezioni di link che hanno lo stesso dominio e condividono il layout e le proprietà visuali.

Algorithm 1 crawlingWebsite(homepage)

```

Input: URL homepage
Output: Set<(URL, URL)> E; Set<(URL, String)> V
frontier ← Set()
Q ← Queue(homepage)
repeat
  currentPage ← Q.dequeue()
  text ← currentPage.getText()
  V.add((currentPage, text))
  webLists ← extractWebLists(currentPage)
  for each a ∈ b do
    pagesToAnalyze ← list.filterNot(page →
      frontier.contains(page))
    Q.enqueue(pagesToAnalyze)
    frontier.add(pagesToAnalyze)
    for each u ∈ pagesToAnalyze do
      E.add((currentPage, u))
    end for
  end for
  until !Q.empty()
return (V, E)
  
```

La soluzione utilizzata per indicizzare un sito Web è stata quella di sfruttare il concetto di **lista Web**.

Per definizione, una lista Web è una collezione di due o più elementi web che hanno struttura HTML simile, visualmente adiacenti ed allineati sulla pagina renderizzata. Questo allineamento può essere visto sia lungo l'asse x (i.e. una lista verticale), sia lungo l'asse y (i.e. una lista orizzontale), o ancora in maniera mista (i.e. griglia).

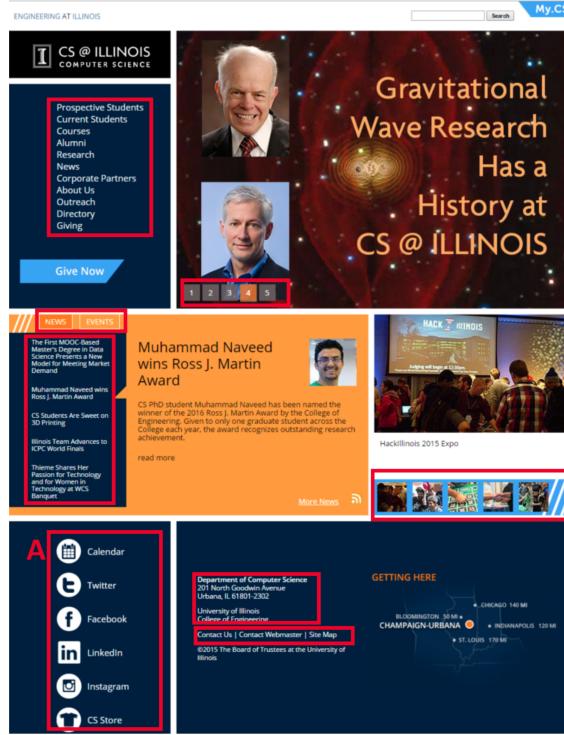


Figura 3.2: Esempio di liste Web

In Figura 3.2 vengono mostrate le liste Web, usate per guidare il processo di Crawling. I link inseriti nel box A sono stati esclusi perché il loro dominio è differente da quello della homepage. Il risultato del Crawler è un sottografo $G' = (V', E')$, dove $V' \subseteq V$ ed $E' \subseteq E$.

3.1.2 Normalizzazione degli URL

Una volta effettuato il processo di Crawling ed estratti gli URL, si procede con la normalizzazione degli stessi. Questo è un processo in cui gli URL vengono modificati e standardizzati in maniera consistente. Il suo obiettivo è poter determinare se due URL, che sono sintatticamente differenti, possono essere equivalenti.

I Crawler effettuano un qualche tipo di normalizzazione degli URL in modo da evitare che il processo di Crawling non vada ad analizzarli più volte.

`http://www.facebook.com/
facebook.com/`

`http://208.77.188.166/
http://www.example.com/`

La normalizzazione di URL, come si evince dagli esempi sopra riportati, può comprendere sia la rimozione del protocollo ("http://") e della stringa "www", oppure la sostituzione dell'indirizzo IP con il nome del dominio. E' bene sottolineare come le due coppie in analisi puntino a due siti Web, rispettivamente *Facebook* ed *example*.

Ci sono diverse modalità di normalizzazione che possono essere effettuate, fra cui la conversione degli URL in minuscolo e la rimozione del "." e ".." per portare gli URL da assoluti a relativi, aggiungere slash finali al componente di percorso non vuoto.

Per la sperimentazione si è scelto di normalizzare gli URL eliminando la dicitura del protocollo ("http://" o "https://"), del "www" e dello slash finale.

3.2 Costruzione del dataset

Il Crawler, a seguito dell'analisi del sito, produce un grafo delle pagine Web ed il contenuto testuale di ogni pagina esplorata. Il grafo del sito Web servirà per generare le sequenze attraverso i Random Walks.

Di seguito verrà spiegato il concetto di Random Walk e come questi sono stati utilizzati ai fini della sperimentazione.

3.2.1 Random Walks

Un Random Walk, o passeggiata aleatoria, è la formalizzazione dell'idea di effettuare passi successivi in direzioni casuali. Dal punto di vista matematico

è il processo stocastico più semplice, ovvero quello markoviano, nel quale la probabilità che determina il passaggio ad uno stato, dipende solo dallo stato immediatamente precedente, e non dal come si è giunti a tale stato.

In una passeggiata aleatoria monodimensionale si studia il moto di una particella puntiforme vincolata a muoversi lungo una retta nelle due direzioni consentite. Ad ogni movimento, questa si sposta a caso o a destra, con una probabilità fissata p , oppure a sinistra, con una probabilità $1 - p$, ed ogni passo è di lunghezza uguale e indipendente dagli altri.

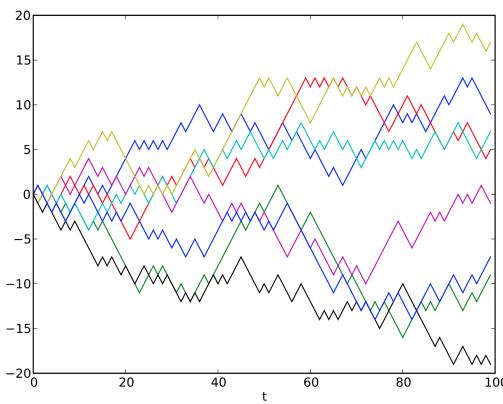


Figura 3.3: Esempio di otto Random Walks in una dimensione

In una passeggiata aleatoria bidimensionale si studia il moto di una particella vincolata a muoversi sul piano spostandosi casualmente ad ogni passo a destra, a sinistra, in alto o in basso con probabilità $1/2$. In particolare, ad ogni passo, la particella può compiere un movimento lungo una delle quattro diagonali con probabilità $1/4$. Ma qual è la probabilità che la particella torni al punto di partenza? In questo caso, la particella, che è libera di camminare casualmente con uguale probabilità nelle quattro direzioni, tornerà infinite volte al punto di partenza.

In una passeggiata aleatoria tridimensionale si studia il moto di una particella vincolata a muoversi nello spazio spostandosi casualmente ad ogni passo a destra, a sinistra, in alto, in basso, in su o in giù con probabilità $1/2$. In pratica, ad ogni passo può compiere un movimento lungo una delle otto diagonali con probabilità $1/8$. Come nel caso precedente, è stata calcolata

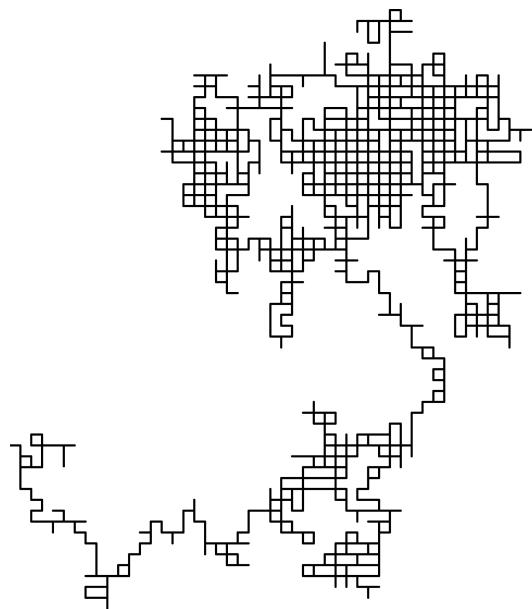


Figura 3.4: Esempio di Random Walks in due dimensioni

la probabilità che la particella torni prima o poi al punto di partenza ed è pari a **0,239**.

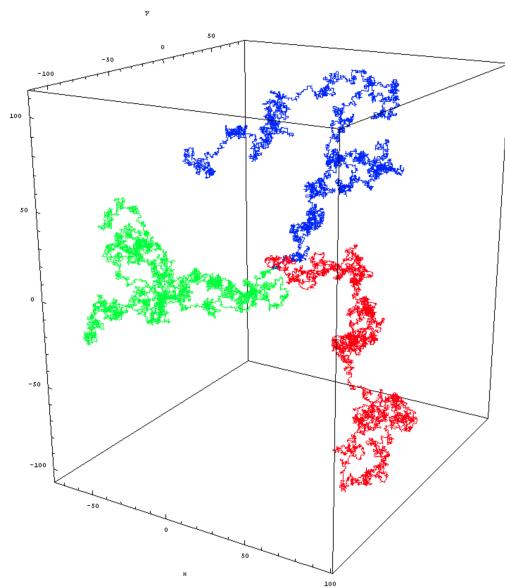


Figura 3.5: Esempio di Random Walks in tre dimensioni

Il concetto di Random Walk è stato applicato nei campi più disparati, alcuni

dei quali sono:

- **Economia finanziaria.** I Random Walk sono stati usati per modellare i prezzi delle azioni sui mercati azionari, tassi di cambio di moneta e materie prime.
- **Genetica.** Descrivono le proprietà statistiche della deriva generica, ovvero una componente dell’evoluzione di una specie dovuta a fattori casuali, che può essere studiata con metodi statistici.
- **Fisica.** Usati come modelli semplificati per studiare il moto browniano, ovvero il moto delle particelle presenti in fluidi che è osservabile al microscopio.
- **Ecologia matematica.** Usati per descrivere i movimenti dei singoli animali, i processi di diffusione della materia, o ancora per la dinamica della popolazione. Quest’ultimo studia i cambiamenti del numero di individui, della densità e della struttura di una o diverse popolazioni. Inoltre analizza i processi biologici e ambientali che influenzano queste trasformazioni.
- **Informatica.** Usati per stimare la dimensione del Web, ovvero il numero di pagine e di siti che fanno parte del World Wide Web.

3.2.2 Generazione delle sequenze

Una passeggiata aleatoria sulla struttura ad hyperlink di un sito Web si basa sull’idea che le connessioni tra nodi (i.e. le pagine del sito) presentano delle informazioni latenti circa la loro correlazione. Per catturarle si è deciso di utilizzare un Random Walker, una componente incaricata di effettuare le passeggiate aleatorie sul grafo di un sito Web. Questa scelta è motivata dal fatto che le metodologie selettive per esplorare un sito Web, che derivano dalla teoria dei grafi, prevedono l’esplorazione di tutte le possibili opzioni per arrivare alla soluzione. Ma tali tecniche sono difficilmente computabili in quanto ricadono nella classe di complessità NP-completa.

Il Random Walker, anche se permette di avere buone approssimazioni nell'esplorazione del grafo del sito, presenta una problematica: potrebbe capitare che la pagina che sta analizzando non presenta link al suo interno. La soluzione più diffusa è quella di effettuare un "salto" verso una qualsiasi altra pagina. Nel nostro caso, invece, dato che le sequenze da generare devono avere una lunghezza massima fissata prima della generazione, se l'attraversatore casuale incontra una pagina priva di hyperlink, allora si blocca semplicemente. La sequenza finale quindi, risulterà più piccola. Questa scelta è stata presa poiché l'informazione cercata nasce da percorsi reali di navigazione. Inoltre non vi è la necessità di una lunghezza obbligatoria da rispettare, in quanto le sequenze possono essere viste come frasi di un testo, dove le parole sono gli URL.

Algorithm 2 rwrGeneration(rwrLength, dbLength, G, α)

```

Input: int rwrLength //numero di passi massimo
Input: int dbLength //numero di frasi
Input: Graph G //il grafo del sito Web
Input: float  $\alpha$ 
Output: List<List<URL>> randomWalks
for each  $i \in Range(0, dbLength)$  do
     $w \leftarrow List()$ 
     $w[0] \leftarrow G.getRandomVertex()$ 
    for each  $j \in Range(1, rwrLength)$  do
         $\lambda \leftarrow Math.random()$ 
        if  $\lambda > \alpha$  then
             $w[j] \leftarrow G.getRandomOutlink(w[j - 1])$ 
        else
             $w[j] \leftarrow w[0]$ 
        end if
    end for
     $randomWalks.add(w)$ 
end for
return  $randomWalks$ 
  
```

Per motivi di sperimentazione sono stati implementati tre tipi diversi di Random Walk, utilizzabili modificando i parametri di esecuzione dell'Algoritmo 2.

- **Random Walk standard.** Il caso standard prevede che si parta da un nodo casuale del grafo e si segua ogni volta un arco a caso fra quelli disponibili, fino al raggiungimento della lunghezza prefissata o all'impossibilità di proseguire.
- **Random Walk con partenza da homepage.** Con quest'altra modalità si ha una partenza fissata. Si parte, quindi, da un nodo prefissato del grafo, generalmente la homepage del sito Web in analisi, in modo da esplorare più percorsi possibili.
- **Random Walk attraverso le Liste.** Questo è il caso in cui si può eseguire uno delle due tipologie di Random Walks viste in precedenza, ma avendo il vincolo delle liste: l'algoritmo, quindi, opererà solo su un sottoinsieme di quello prodotto dalla metodologia scelta.

Quindi, i Random Walks prodotti sono stati trattati come frasi, in cui le parole sono i codici univoci degli URL. L'utilizzo di tali codici ha permesso di ridurre lo spazio in memoria e i tempi di elaborazione.

Le frasi generate dal Random Walk standard sono state sfruttate da Word2Vec per apprendere la struttura del grafo del sito Web. Inoltre, per raggiungere uno degli obiettivi di questa tesi, è stata modificata l'implementazione di Word2Vec e successivamente applicata sui Random Walk con partenza fissa (dalla homepage). La stessa cosa è stata effettuata utilizzando le Liste Web. Nella sezione successiva viene spiegata, in maniera approfondita, la modifica di Word2Vec.

3.2.3 Modifica dell'implementazione di Word2Vec

Un aspetto particolare preso in considerazione in questa tesi è stato quello di chiedersi se, modificando in maniera opportuna l'algoritmo di Word Embedding Word2Vec, si potesse avere un miglioramento nel processo di Clustering delle pagine del grafo del sito Web in analisi.

Per rispondere a questa domanda, si è deciso di analizzare e modificare il modello di apprendimento di Word2Vec Skip-Gram della libreria `deeplearning4j`.

Questo modello consiste nel predire il contesto a partire da una parola. Per una argomentazione più approfondita si veda la Sezione 1.2.2.

La modifica di Skip-Gram consiste nel limitare l'apprendimento dell'algoritmo in maniera tale che venga analizzato solo il contesto sinistro, data una parola. E' importante sottolineare, inoltre, come questo modello sia stato ottimizzato con un valore chiamato **b**, che permette di aumentare o diminuire la finestra di contesto, dando più importanza alle parole più vicine a quella in analisi.

Per la sperimentazione sono stati prodotti, quindi, due tipologie di modelli di Skip-Gram che analizzano solo il contesto sinistro: uno che utilizza il valore b ed un'altro che non lo usa.

3.2.4 Scaling degli embeddings

Una volta prodotti gli spazi vettoriali delle parole, sono state analizzate metodologie di Feature Scaling per capire la migliore da usare per la sperimentazione.

Il Feature Scaling è un metodo usato per standardizzare un intervallo di variabili indipendenti o features di dati. Viene anche chiamato normalizzazione di dati ed è generalmente effettuato nei passi di preprocessing di dati. Non solo, normalizzare i dati significa anche ridurre l'effetto degli Outlier. Il termine Outlier è usato in statistica per definire, in un insieme di osservazioni, un valore anomalo e aberrante (i.e. un valore chiaramente distante dalle altre osservazioni disponibili). Un numero consistente di Outlier nel campione in analisi può portare a risultati fuorvianti. Per esempio, se misurassimo la temperatura di 10 oggetti presenti in una stanza, la maggior parte dei quali risultasse avere una temperatura compresa fra i 20 e 25 gradi Celsius, allora il forno acceso, avente una temperatura di 350 gradi, sarebbe un dato aberrante.

Le tecniche di Scaling più utilizzate sono:

- **Z-score.** Con questa metodologia, i vettori vengono standardizzati

ed avranno le proprietà di una distribuzione normale standardizzata, particolarmente utile nelle operazioni di stima statistica. Sono curve simmetriche con valori più concentrati verso il centro e meno nelle estremità laterali. Un esempio è la curva di Gauss.

Si avranno vettori aventi $\mu = 0$ e $\sigma = 1$, dove μ è la media e σ è la deviazione standard dalla media. Gli Z-score sono calcolati come segue:

$$z = \frac{x - \mu}{\sigma} \quad (3.1)$$

- **Min-Max.** E' un approccio alternativo allo Z-score visto in precedenza. I dati vengono normalizzati usando un intervallo fissato, generalmente tra 0 ed 1, dove 0 è il valore minimo ed 1 quello massimo. Viene applicata la formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.2)$$

- **L2.** Viene chiamato anche Normalizzazione Euclidea. Questo metodo è una normalizzazione vettoriale. Dato un vettore $x = [x_1, x_2, \dots, x_n]$, è possibile calcolare il valore della norma L2 con la seguente formula:

$$|x| = \sqrt{\sum_{k=1}^n x_k^2} \quad (3.3)$$

Infine, per normalizzarlo, bisogna dividere ogni valore di x con quello ottenuto dalla formula di norma L2.

In uno spazio vettoriale occorre calcolare il valore di norma L2 per ogni riga o colonna della matrice, a seconda della scelta e dividere ogni elemento della riga o della colonna per quel valore.

Ai fini della sperimentazione, alla luce di quanto dichiarato dagli autori di [21], si è deciso di utilizzare come tecnica la L2 per ogni riga. Infatti gli stessi autori hanno affermato che, dopo una serie di tentativi, la norma L2 fornisce risultati di normalizzazione superiori se applicata per ogni riga della

matrice.

3.3 Web page Clustering

L'operazione di Clustering delle pagine di un sito Web, generalmente, può avvenire sfruttando la struttura connessa del sito o trattando le singole pagine come documenti. Nel primo caso, vengono applicate tecniche e metodologie derivanti dalla teoria dei grafi per partizionare il grafo e raggruppare le pagine simili. Nel secondo caso, invece, viene analizzato il contenuto testuale visivo della pagina, ovvero tutto quello che l'utente può percepire durante la navigazione sulle pagine di un sito Web.

Gli hyperlink tra le pagine vengono usati per organizzare i contenuti, puntando a pagine differenti. Il contenuto testuale, data l'ambiguità del linguaggio naturale, può fornire indizi sbagliati e considerare diverse pagine correlate solo per una differente distribuzione dei termini.

In questa tesi si è scelto di utilizzare algoritmi di Clustering per raggruppare le pagine di un sito Web in base alla loro rappresentazione vettoriale. I vettori utilizzati per l'operazione di Clustering sono:

- **Vettori dei link.** Sono state utilizzate tre tipologie di Word2Vec: Skip-Gram che considera solo il contesto sinistro ottimizzato (utilizzando il valore della b) che apprende da sequenze generate da Random Walk partendo dalla homepage; Skip-Gram che considera solo il contesto sinistro non ottimizzato (ignorando la b) che apprende da sequenze generate da Random Walk partendo dalla homepage; Skip-Gram che considera il contesto sia destro sia sinistro che apprende da sequenze generate da Random Walk standard.

E' stato inoltre applicato sul grafo del sito Web LINE (Sezione 1.2.4), un altro algoritmo che produce gli embedding dei nodi in base alla loro prossimità con gli altri. Questo algoritmo è stato applicato nelle sue

due varianti: prossimità di primo e secondo ordine.

I vettori dei link sono stati normalizzati utilizzando L2 per ogni riga.

- **Vettori dei contenuti.** Le pagine del sito Web sono state opportunamente preprocessate per essere trattate come documenti. Sono stati rimossi i tag HTML, i caratteri di escape e non alfanumerici e le parole troppo frequenti ($> 90\%$) e poco frequenti ($< 5\%$). Successivamente, sono stati applicati Doc2Vec (in Sezione 1.2.3) e TF-IDF (in Sezione 1.2.1) sulla pagina Web preprocessata. I vettori dei contenuti sono stati normalizzati utilizzando L2 per ogni riga.
- **Vettori dei link-contenuti.** Questa tipologia di vettori è stata prodotta andando a concatenare i vettori dei link e quelli dei contenuti aventi stesso codice identificativo dell'URL. Il risultato sarà un nuovo spazio vettoriale i cui vettori avranno dimensione $n + m$, dove n è la lunghezza del vettore dei link ed m è la lunghezza del vettore dei contenuti. Prima della concatenazione, i vettori di link e di contenuto sono stati normalizzati utilizzando L2 per ogni riga.

3.4 Esempio di dataset

Questa sezione ha il compito di spiegare come è stato strutturato il dataset per la sperimentazione. I file sono stati generati dal Web Crawling e dai Random Walks, usati per effettuare la sperimentazione descritta in questa tesi.

Per spiegare il dataset è stato utilizzato come esempio il sito del dipartimento di informatica di Stanford, CA: `cs.stanford.edu`.

seedsMap.txt Questo file contiene le associazioni tra gli URL e il codice identificativo, usato per ridurre i tempi di elaborazione e spazio di archiviazione.

`cs.stanford.edu/csdcf/policies 73`

```
cs.stanford.edu/admissions/reapplying 52  
cs.stanford.edu/people/eaf/wordpress/videos 247  
...
```

vertex.txt Contiene il contenuto testuale di ogni pagina esplorata. Ogni riga è formata dal codice identificativo di un URL ed il relativo contenuto.

```
3 skip to skip to content skip to navigation webauth login ...  
121 webauth error webauth error an error has occurred error ...  
90 skip to content skip to navigation webauth login sunetid ...  
...
```

edges.txt Contiene gli archi tra i nodi del grafo del sito web, i quali non sono altro che i collegamenti tra le pagine. Questo file viene usato per la generazione delle sequenze dall'algoritmo dei Random Walks.

```
3 69  
3 101  
3 88  
...
```

sequenceIDs.txt Contiene le sequenze generate da un Random Walker. Nel file vengono riportati i passi generati dall'algoritmo di Random Walk partendo da un nodo casuale del grafo del sito Web.

```
109 33 106 89 10 108 57 91 8 51  
80 42 17 95 66 109 109 78 44 22  
206 280  
...
```

sequenceIDsFromHomepage.txt Contiene le sequenze generate da un Random Walker. Nel file vengono riportati i passi generati dall'algoritmo di

Random Walk specificando il nodo di partenza (i.e. la homepage) di ogni sequenza.

```
3 37 72  
3 12 48 47 22 64 74 48 36 8  
3 6 26 12 57 63 95 109 87 71  
...
```

embeddings_with_b.txt, embeddings_no_b.txt, embeddings_normal.txt,
embeddings_line_first.txt, embeddings_line_second.txt, embeddings_doc2vec.txt
gli embedding degli URL appresi da sequenze generate da Random Walk con partenza da homepage, che sono stati generati dal modello di apprendimento Skip-Gram, rispettivamente ottimizzato e non, che considera solo il contesto sinistro, data una parola.

embeddings_normal.txt contiene gli embedding degli URL appresi da sequenze prodotte da Random Walk standard, che sono stati generati dal modello di apprendimento Skip-Gram che considera il contesto destro e sinistro, data una parola.

embeddings_line_first.txt ed **embeddings_line_second.txt** contengono gli embedding degli URL appresi dal file degli archi del grafo del sito Web. E' stato utilizzato LINE rispettivamente con prossimità di primo e secondo ordine.

embeddings_doc2vec.txt contiene gli embedding del contenuto delle pagine del grafo del sito Web, dove ognuno dei vettori è associato ad un codice identificativo dell'URL.

```
80 0.030692825093865395 0.09835819154977798 ...  
44 -0.06424273550510406 -0.0433584563434124 ...  
69 -0.1409958302974701 -0.013164487667381763 ...  
...
```

groundTruth.csv Contiene la tavola di verità, in cui ad ogni URL è associata una etichetta che indica il cluster di appartenenza. Questo file viene utilizzato per misurare la bontà dell'algoritmo di Clustering ed è stato usato nella sperimentazione. Nella tavola di verità è presente come etichetta -1: se un URL presenta questo valore, significa che la pagina non è stata assegnata ad alcun raggruppamento.

```
cs.stanford.edu/ip -1  
cs.stanford.edu/about/contact-us 1  
cs.stanford.edu/academics 2  
cs.stanford.edu/academics/phd 2  
cs.stanford.edu/admissions 3  
cs.stanford.edu/computing-guide 4  
...  
...
```

Capitolo 4

Sperimentazione

In questo capitolo verranno descritte le modalità di esecuzione della sperimentazione.

L’obiettivo è quello di valutare l’efficacia della modifica del modello di Word2Vec Skip-Gram al fine di migliorare il processo di Clustering delle pagine di un sito Web.

Si è cercato di capire, inoltre, se e come i dati strutturati, di cui una pagina Web si compone, possano essere combinati con dati non strutturati, come quelli testuali, al fine di migliorare il processo di Clustering, valutando in dettaglio le cause di un eventuale successo o insuccesso delle tecniche utilizzate. A tal fine, verranno confrontate le performance degli algoritmi di Clustering basati su rappresentazione vettoriale, ovvero K-Means e HDBScan.

Si descrivono in seguito i dataset e le configurazioni degli algoritmi di Clustering su cui sono realizzate le sperimentazioni e le metriche utilizzate per valutare la qualità dei Cluster estratti.

4.1 Dataset

La sperimentazione è stata effettuata sfruttando i dati provenienti da siti Web appartenenti ad importanti dipartimenti di Computer Science: **Illinois** (cs.illinois.edu), **Oxford** (cs.ox.ac.uk), **Priceton** (cs.princeton.edu) e **Stanford** (cs.stanford.edu). La motivazione di questa scelta è legata al fatto che le nostre competenze, per assegnare ad ogni pagina del sito una etichetta, appartengono a questo dominio applicativo. Questa decisione è necessaria per creare una ground truth per la valutazione dei risultati del Clustering.

Per ogni sito Web è stato lanciato il processo di Crawling e sono stati estratti due differenti grafi:

- **NoConstraint**, prodotto dal Crawler tradizionale. Il grafo Web $G_{nc} = (V, E)$ estratto rappresenta fedelmente il sito. In particolare V rappresenta l'insieme delle pagine appartenenti al sito ed E è l'insieme di tutte le coppie (i, j) , in cui la pagina con URL i contiene un hyperlink alla pagina con URL j .
- **ListConstraint**, prodotto dal Crawler che sfrutta le liste Web. Il grafo Web $G_{lc} = (V, E)$ estratto è filtrato, eliminando tutti gli archi $(i, j) \in E$, in cui l'URL j non è contenuto in nessuna lista Web nella pagina con URL i .

Successivamente, i grafi NoConstraint e ListConstraint sono stati utilizzati per generare sequenze di URL di 100.000, 500.000 e 1.000.000 con profondità di 10, 15, 20. Una volta prodotte, sono state apprese da Skip-Gram ottimizzato che considera solo il contesto sinistro (Random Walk con partenza fissa), Skip-Gram non ottimizzato che considera solo il contesto sinistro (Random Walk con partenza fissa) e Skip-Gram che considera contesto destro e sinistro (Random Walk standard) con dimensione della finestra di apprendimento di 2, 3, 5, 7.

In Tabella 4.1 vengono riportate le dimensioni di ogni sito al termine del

Tabella 4.1: Descrizione dei siti Web

Sito	# pagine	# archi	# archi con Liste Web	# Cluster
Illinois	563	9415	5330	10
Oxford	3480	44526	35148	19
Priceton	3132	122493	104585	16
Stanford	167	30087	12372	10

Crawling. In particolare, per analizzare correttamente il contributo fornito dall'applicazione delle Liste Web nel processo di Clustering, sono state confrontate le pagine estratte sia dal Crawler che usa le Liste Web, sia da quello tradizionale (prima colonna). Inoltre, è stata riportata la dimensione della collezione degli archi ottenuta dal Crawler tradizionale (seconda colonna) e da quello che usa le Liste Web (terza colonna). Nell'ultima colonna sono stati inseriti il numero dei Cluster identificati manualmente dagli esperti durante la generazione della ground truth.

4.2 Metriche

Valutare le prestazioni di un algoritmo di Clustering non è semplice. Queste non dovrebbero considerare gli specifici valori delle etichette assegnate dall'algoritmo, piuttosto dovrebbero verificare se il raggruppamento generato dall'algoritmo definisce una separazione dei dati simile a quello fornito nella ground truth, ovvero il vero valore delle etichette. Un'altra modalità di valutazione consiste nel basarsi su funzioni di similarità, come per esempio che elementi dello stesso raggruppamento siano più simili rispetto a quelli di Cluster differenti.

Per la sperimentazione si è scelto di utilizzare le metriche elencate in seguito per valutare l'efficacia dell'approccio proposto.

- **Omogeneità** [30]. Ogni Cluster dovrebbe contenere elementi appartenenti alla stessa classe. Questa metrica viene calcolata dall'entropia condizionata della distribuzione di classe, dato il Cluster. Formalmente,

l'omogeneità viene definita come:

$$h = 1 - \frac{H(C|K)}{H(C)} \quad (4.1)$$

dove $H(C|K)$ è l'entropia condizionata delle classi date le assegnazioni dei Cluster e $H(C)$ è l'entropia delle classi, ossia:

$$H(C|K) = - \sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{n_{ck}}{N} \log \frac{n_{ck}}{n_k} \quad (4.2)$$

$$H(C) = - \sum_{c=1}^{|C|} \frac{n_c}{N} \log \frac{n_c}{N} \quad (4.3)$$

con N il numero totale delle pagine, n_c e n_k il numero delle pagine appartenenti alla classe c o al Cluster k , n_{ck} il numero delle pagine della classe c assegnate al cluster k .

Il valore di questa metrica ha come intervallo $[0, 1]$. In una situazione ideale, l'Omogeneità assume valore 1.

- **Completezza** [30]. Tutti gli elementi di una stessa classe dovrebbero appartenere ad uno stesso Cluster. Questa metrica è simmetrica all'omogeneità ed è calcolata dall'entropia condizionata della distribuzione degli assegnamenti di ogni classe ad un dato Cluster. Formalmente, la Completezza è definita come segue:

$$c = 1 - \frac{H(C|K)}{H(K)} \quad (4.4)$$

dove $H(C|K)$ è l'entropia condizionata delle classi date le assegnazioni dei Cluster. Questa è stata definita nell'equazione 4.2. $H(K)$ è l'entropia dei cluster ed è così formulata:

$$H(K) = - \sum_{k=1}^{|K|} \frac{n_k}{N} \log \frac{n_k}{N} \quad (4.5)$$

con N è il numero totale delle pagine ed n_k è il numero delle pagine

appartenenti al Cluster K.

Come l'Omogeneità, anche questa metrica ha come intervallo [0, 1]. In una soluzione perfetta, ogni distribuzione di classe dovrebbe convergere in un Cluster, avendo come Completezza 1.

- **V-Measure** [30]. E' la media armonica tra l'Omogeneità e la Completezza ed è formalmente descritta come segue:

$$v = 2 \cdot \frac{h \cdot c}{h + c} \quad (4.6)$$

dove h è il valore dell'Omogeneità e c quello della Completezza.

- **Adjusted Mutual Information (AMI)**. Questa metrica è una variazione della *Mutual Information* (MI), ovvero una funzione che misura la corrispondenza delle due informazioni, ignorando le permutazioni. La MI viene definita come:

$$MI = \sum_{i \in K} \sum_{j \in C} \log \frac{P(i, j)}{P(i)P(j)} \quad (4.7)$$

dove C è l'insieme delle classi reali, K è l'insieme dei Cluster appresi, $P(i, j)$ indica la probabilità di un elemento appartenente sia alla classe reale i che a quella appresa j , $P(i)$ è la probabilità a priori che un elemento appartenga alla classe i .

La MI è generalmente più alta per due Clustering aventi un più grande numero di Cluster, non considerando il fatto che in realtà ci sono più informazioni condivise. La AMI rappresenta un adeguamento della MI per superare questa limitazione.

Il valore di questa metrica ha come intervallo [0, 1]. In una situazione ideale, l'AMI assume valore 1.

- **Adjusted Random Index (ARI)** [18]. Utilizzando la ground truth e gli assegnamenti restituiti da un algoritmo di Clustering, questa metrica calcola la similarità considerando tutte le coppie dei campioni e contando quelle che sono state assegnate nello stesso o in differenti Cluster, sia dalla tabella di verità che dal processo di Clustering reale. In

parole povere, l'ARI misura l'accuratezza del processo di Clustering, ossia la percentuale di coppie di oggetti per i quali la ground truth e l'algoritmo concordano sull'assegnazione. Come la AMI, anche qui si ignorano le permutazioni tra i due insiemi.

Questa metrica dipende fortemente dal *Random Index* (RI), che misura anch'essa la similarità tra due processi di Clustering. Formalmente viene definita come:

$$RI = \frac{a + b}{\binom{n}{2}} \quad (4.8)$$

dove a è il numero di coppie di elementi che si trovano nella stessa classe e nell'insieme delle etichette predette, b è il numero di coppie di elementi che appartengono a differenti classi e nell'insieme delle etichette predette, $\binom{n}{2}$ è il numero totale di tutte le possibili coppie nel dataset. Si è preferito utilizzare l'ARI poiché il RI, nel caso di Cluster computati in maniera casuale, non assume valori costanti (per esempio 0). In questo modo si garantisce che i Cluster generati in modo casuale abbiano un valore di ARI prossimo allo 0.

Il valore di questa metrica ha come intervallo $[0, 1]$. In una situazione ideale, l'AMI assume valore 1, ovvero quando le etichette che si trovano nella ground truth e quelle predette sono identiche.

- **Silhouette.** Misura la forma di ogni Cluster. Misura quanto simile è un elemento al Cluster a cui è assegnato (coesione) comparato con gli altri insiemi (separazione). Formalmente, questa metrica viene definita come:

$$s = \frac{b - a}{\max(a, b)} \quad (4.9)$$

dove a è la distanza media tra un elemento e tutti gli altri della stessa classe, b è la distanza media tra un elemento e tutti gli altri nella classe più vicina.

Il valore di questa metrica, a differenza delle precedenti, ha come intervallo $[-1, 1]$. Un valore intorno allo 0 indica Cluster sovrapposti; -1 cluster non definiti; 1 elementi che sono altamente coesi con gli altri dello stesso raggruppamento ma bassamente coesi per quelli appartenenti

a Cluster vicini.

4.3 Configurazioni

Si descrivono di seguito le configurazioni utilizzate per la sperimentazione in questa tesi.

4.3.1 Testo

Sono state applicate tecniche di Text Mining per il Clustering basato sul contenuto testuale. Con questa metodologia viene considerata solo l'informazione estratta dal testo, assumendo che i termini all'interno del sito Web siano indipendenti l'uno dall'altro, così come i documenti. Vengono ignorate le relazioni interdipendenti tra questi. Il Web si discosta dall'analisi classica di testi per l'esistenza di relazioni tra le pagine. Tuttavia l'analisi testuale rimane una tecnica importante.

Nella fase di sperimentazione sono stati utilizzati due algoritmi per estrarre informazioni dal testo: *Doc2Vec* (esaminato in Sezione 1.2.3) e *tf-idf* (esaminato in Sezione 1.2.1).

I parametri utilizzati in Doc2Vec sono stati:

- **minWordFrequency**: questo valore rappresenta il numero minimo di occorrenze per considerare una determinata parola. E' stato impostato ad 1.
- **layerSize**: indica la dimensione dei vettori in output. E' stato impostato a 100.
- **windowSize**: E' la dimensione della finestra di contesto, utile per svolgere il training sulle parole del testo. E' stata impostata a 5.

I parametri, invece, per creare la matrice con tf-idf sono stati:

- **max-df**: questo valore rappresenta la massima frequenza, all'interno dei documenti, che un termine può avere per essere utilizzato nella matrice tf-idf. Se un termine appare molte volte nel corpus, molto probabilmente avrà poco significato. E' stato impostato a 0.9.
- **min-df**: indica il numero minimo di documenti in cui un termine dovrà apparire per essere considerato. E' stato impostato a 0.05.
- **ngram-range**: vengono presi in considerazioni gli n-grammi di lunghezza compresa nell'intervallo specificato in questo parametro. Nello specifico, questo parametro è una tupla (`min_n`, `max_n`) che definisce il confine minimo e massimo dell'intervallo di valori n-esimi per differenti n-gram per essere estratti. Tutti i valori di n che si trovano nell'intervallo $min_n \leq n \leq max_n$ vengono considerati. Un n-gramma è una sottosequenza di n elementi di un'altra. E' stato impostato a (1, 2).

Successivamente, la dimensione dei vettori della matrice tf-idf è stata ridotta a 100.

4.3.2 Random Walk

Considerando i Random Walk come frasi, è possibile applicare gli algoritmi di Word Embedding per raggruppare pagine di un sito Web sulla base del contesto in cui appaiono, ovvero le pagine che più verosimilmente appariranno insieme nelle sequenze.

Queste frasi sono state generate da grafi di siti Web a liste di costrizione e senza costrizione (esaminate in sezione 4.1). Sono stati prodotti Random Walk che partono dalla homepage e quelli standard, che sono stati rispettivamente appresi dalla versione modificata di Skip-Gram (ottimizzato e non, che considera solo il contesto sinistro) e da Skip-Gram tradizionale. Per entrambi i casi, sono state create un numero di frasi di 100.000, 500.000 e 1.000.000 e la loro lunghezza massima di 10, 15, 20.

Per tutte le versioni di Word2Vec Skip-Gram, sono stati utilizzati i seguenti paragrafi:

- **minWordFrequency**: questo valore rappresenta il numero minimo di occorrenze per considerare una determinata parola. E' stato impostato ad 1.
- **layerSize**: indica la dimensione dei vettori in output. E' stato impostato a 100.
- **windowSize**: E' la dimensione della finestra di contesto, utile per svolgere il training sulle parole del testo. E' stata impostata a 2, 3, 5, 7.
- **iterate**: numero di iterazioni sulle frasi da apprendere. E' stato impostato a 50 per Random Walk di 100.000 frasi; 10 per Random Walk di 500.000 frasi; 3 per Random Walk di 1.000.000 di frasi.

I valori dei parametri sopra citati sono stati utilizzati per tutti i dataset della sperimentazione.

Inoltre, si è voluto confrontare i modelli Skip-Gram tradizionale e quello modificato con un nuovo algoritmo di apprendimento, chiamato LINE (esaminato in Sezione 1.2.4). A differenza di Skip-Gram, LINE considera solo relazioni dirette (prossimità di primo ordine) o al più relazioni tra nodi fratelli (prossimità di secondo ordine). Richiede, per questo, il file contenente gli archi del sito Web, che viene generato dal Crawler.

I parametri per utilizzare LINE di primo e secondo ordine sono:

- **-train**: path del file contenente gli archi del grafo Web, ovvero `edges.txt`.
- **-output**: path del file contenente gli embeddings prodotti dall'algoritmo.
- **-order**: tipo di prossimità per produrre gli embeddings, ovvero 1 per primo ordine e 2 per secondo ordine.

4.3.3 Combinato

Sono state combinate le informazioni riguardanti la struttura del sito Web e il contenuto testuale delle pagine. Le combinazioni prodotte per la sperimentazione sono di due tipi:

- **Combined List Constraint.** Sono stati combinati i vettori prodotti dalla struttura ad hyperlink del sito, usando il Crawler che sfrutta le Liste Web, e quelli prodotti dal contenuto delle pagine. Sono stati prodotti vettori aventi come dimensione 200.
- **Combined No Constraint.** Come nella configurazione precedente, sono stati combinati testo e struttura ad hyperlink del sito Web in un singolo spazio vettoriale, avente dimensione 200. In questo caso, è stato utilizzato il Crawler tradizionale.

4.4 Parametri degli algoritmi di Clustering

Una volta prodotti gli embeddings dai rispettivi algoritmi, sono stati normalizzati secondo la norma L2 (esaminata in Sezione 3.2.4) ed è stato effettuato il Clustering delle pagine del sito Web. Da questo processo sono state escluse tutte le pagine che presentavano nella ground truth etichette pari a -1, poiché tale valore significa che l'esperto che ha creato la tabella di verità non è riuscito ad assegnare a nessun raggruppamento la pagina in questione.

In K-Means (approfondito in Sezione 1.3.2), il parametro utilizzato nella sperimentazione è stato **n_clusters**, ovvero il numero di Cluster totale da generare. Per rendere il sistema dinamico, si è scelto di settare questo parametro con il numero delle etichette distinte presente nella ground truth.

In HDBScan (approfondito in Sezione 1.3.2), il parametro utilizzato nella sperimentazione è stato **min_cluster_size**, ovvero la dimensione minima di

un raggruppamento per essere considerato Cluster. Questo valore è stato impostato a 5.

4.5 Analisi dei risultati

Di seguito si confrontano i risultati degli algoritmi di Clustering basati sulle configurazioni sopra citate. Sono stati riportati solo i risultati di Skip-Gram ottimizzato che considera solo il contesto sinistro (*leftSkipgram*) e Skip-Gram tradizionale (*normalSkipgram*) poiché, nella maggior parte dei casi, *leftSkipgram* ha fornito risultati migliori rispetto alla versione modificata non ottimizzata.

Per motivi di praticità, nella configurazione dei Random Walk è stata utilizzata la dicitura $[db, window, depth]$, dove db rappresenta il numero di frasi da far apprendere all'algoritmo, $window$ la lunghezza della finestra di contesto e $depth$ la lunghezza massima della singola frase.

Inoltre, per motivi di spazio, si è scelta la dicitura *LINE-1* e *LINE-2* per riferirsi rispettivamente a LINE con primo e secondo ordine di prossimità.

4.5.1 cs.illinois.edu

Random Walk

Nelle Tabelle 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 sono stati riportati i valori delle metriche di *leftSkipgram* e *normalSkipgram* utilizzati con gli algoritmi di Clustering KMeans e HDBScan. Queste misurazioni sono risultate particolarmente utili per capire quale fosse la configurazione migliore per numero di frasi, per dimensione della finestra di apprendimento e per dimensione massima delle frasi per ogni algoritmo di apprendimento e di Clustering, applicati rispettivamente ai dataset con e senza liste di costrizione.

In questo caso, le configurazioni migliori sono state selezionate in base alla combinazione di metriche con valori più alti, e sono state raccolte in Tabella

4.10 e confrontate con LINE. Si evince come LINE non riesca ad estrarre informazioni utili al Clustering dalla struttura ad hyperlink del sito: la bontà dei risultati, infatti, risulta piuttosto bassa.

Considerando solo le informazioni estraibili dalla struttura ad hyperlink del sito Illinois, l'algoritmo più efficiente è **normalSkipgram** con configurazione **[100K, 2, 20]** e **senza liste di costrizione**, abbinato a **KMeans**. In questo caso, le liste di costrizione non hanno migliorato la qualità dei risultati: normalSkipgram è riuscito ad estrarre informazioni utili pur avendo un numero di frasi relativamente piccolo (100.000) avente lunghezza di 20 URL. Una finestra di contesto piccola (2) ha permesso a normalSkipgram di escludere tutte quelle informazioni che potevano compromettere la bontà dell'operazione di Clustering.

Testo

Come nella configurazione precedente, pur trattando le pagine del sito Illinois come semplici documenti, si riescono ad estrapolare dal testo informazioni più significative con **Doc2Vec**, utilizzate in maniera più efficace da **HDBScan** (Tabella 4.11).

Combinato

In Tabella 4.12 sono stati raccolti i risultati del Clustering utilizzando sia l'informazione codificata nella struttura sia quella estrapolata dal testo. Ai fini per la concatenazione, sono stati considerati per leftSkipgram e normal-Skipgram solo le configurazioni migliori, opportunamente analizzate e scelte in precedenza. Si evince come **HDBScan**, sfruttando l'informazioni estratta da **leftSkipgram** con configurazione ottimale [1M, 2, 20] e combinato con i vettori prodotti da **Tf-Idf**, riesca ad assegnare allo stesso Cluster pagine appartenenti alla stessa classe. Anche in questo caso, le liste non hanno migliorato la qualità dei Cluster.

4.5.2 cs.ox.ac.uk

Random Walk

Nelle Tabelle 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, così come nel caso precedente, sono stati riportati i risultati di leftSkipgram e normalSkipgram per tutte le combinazioni. La Tabella 4.20 evidenza che i risultati migliori si concentrano fra le configurazioni [1M, 2, 15] e [1M, 3, 20]. Confrontandole, si nota come l'ultima, a parità di valori, mostra un punteggio più alto nella metrica Completezza (0.02), riuscendo ad assegnare in maniera più opportuna elementi della stessa classe allo stesso raggruppamento rispetto alla prima configurazione.

Come in precedenza, sono state scelte le migliori configurazioni di numero di URL, dimensione della finestra e lunghezza delle frasi, e sono state confrontate con LINE. In Tabella 4.21 si nota come i punteggi delle metriche sono concentrate su **leftSkipgram senza costrizioni**, con una configurazione di **[1M, 5, 20]**. In questo sito, l'analisi del solo contesto sinistro con una dimensione standard della finestra di contesto (5) ha permesso di estrarre informazioni più pertinenti dalla struttura di Oxford, che sono state sfruttate a pieno da **KMeans**. Anche in questo caso, le Liste Web non hanno contribuito a migliorare la qualità dei Cluster.

KMeans e HDBScan, utilizzando le informazioni estrapolate da LINE, hanno raggiunto performance più basse, sia con che senza liste di costrizione.

Testo

In Tabella 4.22 vengono raccolti i valori degli algoritmi di Clustering basati solo sul contenuto testuale. Senza ombra di dubbio, la matrice di termini-documenti prodotta da **Tf-idf** ha estratto informazioni utili per il raggruppamento effettuato da **KMeans**. La qualità dei Cluster è molto alta: le pagine di Oxford si sono rivelate consistenti con gli argomenti trattati, permettendo a KMeans di generare Cluster contenenti tutte quelle pagine che appartengono alla stessa sezione del sito Web. Ad esempio, quelle che appar-

tengono alla sezione "supporto di ricerca" sono state inserite, nella stragrande maggioranza, nello stesso raggruppamento.

Combinato

Confrontando in Tabella 4.23 i risultati della combinazione delle migliori configurazioni di leftSkipgram e NormalSkipgram, concatenati con Doc2Vec e Tf-Idf, si evince come **KMeans** ottenga maggiori performance dalla concatenazione dei vettori di **leftSkipgram** con configurazione [1M, 5, 20] e **Tf-Idf**, arrivando a punteggi molto alti. Come nel caso di combinato di Oxford, non vi è un miglioramento della bontà del processo di Clustering utilizzando delle Liste Web.

4.5.3 cs.princeton.edu

Random Walk

Nelle Tabelle 4.24, 4.25, 4.26, 4.27, 4.28, 4.29, 4.30, 4.31 sono stati raccolti i valori delle metriche, sia per leftSkipgram che per normalSkipgram, come nei casi precedenti. Anche in questo caso sono state scelte le configurazioni aventi metriche con valori più alti. Nel caso della Tabella 4.24 è stata scelta come configurazione migliore per KMeans con leftSkipgram con liste di costruzione [100K, 7, 20] poiché, pur non avendo i più alti valori in Completezza ed Omogeneità, ha il maggior punteggio in V-Measure: di conseguenza, presenta un certo equilibrio tra le metriche precedentemente citate.

Le migliori configurazioni sono state scelte e raccolte nella Tabella 4.32 dove, confrontandole con LINE di primo e secondo ordine di prossimità, **normal-Skipgram con liste di costruzione**, con una configurazione di **[1M, 5, 10]** affiancato a **KMeans**, ottiene punteggi più alti. Considerando sia contesto destro che quello sinistro, con una dimensione della finestra pari a 5 (quindi standard), normalSkipgram ha estratto informazioni utili pur considerando un numero discreto di relazioni indirette tra le pagine di Princeton. In questo

caso, le Liste Web hanno migliorato la qualità dei Cluster creati.

E' interessante notare come LINE con secondo ordine di prossimità, applicato sul dataset di Princeton con liste di costrizione, permette di estrarre informazioni utili per dare più forma ai raggruppamenti: questo algoritmo, infatti, permette di ottenere, abbinato ad HDBScan, un punteggio di Silhouette pari a 0.21, un valore che si distacca in maniera rilevante da quelli degli altri algoritmi messi a confronto. Non solo, sempre utilizzando HDBScan e considerando il dataset senza liste di costrizione, LINE con primo ordine di prossimità permette di estrarre informazioni per aumentare il valore di Completezza, riuscendo a raggiungere lo 0.62 e di raggruppare tutti gli elementi di una classe in uno stesso Cluster.

Testo

Considerando le pagine di Princeton come semplici documenti testuali, da questo caso della sperimentazione, i cui risultati sono stati inseriti nella Tabella 4.33, è evidente che la matrice di **Tf-idf** ha permesso di aumentare le performance di **KMeans**.

Combinato

La Tabella 4.34 mostra i risultati ottenuti dalla combinazione dei vettori prodotti dai diversi algoritmi. Considerando le pagine indicizzate dal Crawler tradizionale e usando le informazioni estratte da **normalSkipgram** con combinazione [1M, 7, 10] e **Tf-Idf**, **KMeans** ha prodotto dei Cluster con un certo livello di bontà. Le Liste Web non hanno migliorato la qualità del processo di Clustering.

4.5.4 cs.stanford.edu

Random Walk

Nelle Tabelle 4.35, 4.36, 4.37, 4.38, 4.39, 4.40, 4.41, 4.42 sono stati raccolti i valori delle metriche, sia per leftSkipgram che per normalSkipgram. Sono state scelte le configurazioni in base alla combinazione di metriche con valori più alti. Si può notare come le configurazioni migliori non hanno permesso di raggiungere elevate performance agli algoritmi di Clustering. La struttura del sito Web non ha permesso di effettuare un processo di Clustering avente una buona qualità, poiché Stanford potrebbe essere un sito Web non altamente strutturato.

Inoltre, in Tabella 4.42, sono due le configurazioni aventi il maggior numero di valori più alti delle metriche: [500K, 2, 20] e [500K, 3, 20]. La configurazione ottimale è la seconda, poiché, pur avendo gli stessi valori per le stesse metriche, differisce in Silhouette di 0.01. Questa configurazione ha permesso di avere dei raggruppamenti aventi una forma leggermente migliore, pur avendo entrambe prodotto Cluster sovrapposti: entrambi i valori, infatti, sono vicini allo 0.

In Tabella 4.43 sono state raccolte tutte le migliori configurazioni di leftSkipgram e normalSkipgram, e confrontate con entrambi gli ordini di prossimità di LINE. L'unico aspetto interessante è il valore della Silhouette prodotto da LINE-2 su un dataset con liste di costrizione, affiancato ad HDBScan pari a 0.01, un valore decisamente più alto rispetto agli altri algoritmi di apprendimento utilizzati su questo sito. Utilizzando solo l'informazione strutturale ottenuta dalle relazioni indirette, i Cluster prodotti da HDBScan sono più definiti. I risultati più alti sono stati ottenuti da **normalSkipgram con liste di costrizione**, affiancato a **KMeans**, con configurazione [1M, 2, 20]. Analizzare, quindi, pagine collegate tra loro da molti elementi, non permette di aumentare le performance del Clustering. Le Liste Web hanno permesso di filtrare in maniera consistente le pagine esplorate in quanto il sito include la maggior parte degli hyperlink in sezioni nascoste, prevalentemente in menu dropdown (a scomparsa).

Testo

Analizzando solo il contenuto testuale delle pagine Web di Stanford, si evince dalla Tabella 4.44 che **Tf-idf** con **KMeans** ha ottenuto le migliori performance, anche se i Cluster risultano non propriamente definiti (-0.30).

Combinato

I risultati prodotti dalla combinazione dei vettori prodotti dai diversi algoritmi, raggruppati nella tabella 4.45, hanno evidenziato che la combinazione dei vettori di **normalSkipgram** e di **Doc2Vec**, con configurazione [1M, 2, 20], utilizzati con **KMeans** hanno prodotto dei Cluster con un certo livello di bontà. Come nel caso dell’analisi della struttura del sito, le Liste Web hanno permesso di filtrare gli hyperlink inseriti nelle sezioni nascoste.

4.6 Considerazioni finali sui risultati

Analizzate le singole configurazioni dei siti, i risultati migliori sono stati raggruppati nelle Tabelle 4.46, 4.47, 4.48, 4.49 e confrontate, per capire quali tra queste forniscono le migliori performance.

Le Tabelle in questione sono strutturate in due parti: la prima parte raccoglie la migliore configurazione analizzando la struttura ad hyperlink del sito (prima riga), la migliore configurazione analizzando il testo delle pagine (seconda riga) e quella migliore combinando le informazioni estratte dalla struttura e quelle dai termini; la seconda parte raccoglie la tripla *[db, window, depth]* della migliore configurazione che analizza la struttura e quella della combinazione di informazioni.

In generale, si nota immediatamente che, utilizzando questi siti Web, il miglior algoritmo di Clustering che riesce a sfruttare al meglio le informazioni estratte è KMeans. Un’analisi più approfondita ci indica che la versione tradizionale di Skip-Gram estrae informazioni più utili rispetto a quella modifi-

cata: l'esclusione del contesto destro causa un filtraggio di informazioni che potrebbero migliorare il processo di Clustering delle pagine del sito. Difatti più è alto il numero di frasi generate e di lunghezza massima di esplorazione del Crawler, maggiore è la quantità di informazioni estratte dalla struttura del sito, utili a migliorare la bontà dei Cluster generati. Inoltre un minor numero di relazioni indirette tra due pagine migliora il Clustering.

L'uso delle liste non aumenta in maniera significativa le performance degli algoritmi di Clustering. Questo può essere l'effetto di due cause: o il Crawler che sfrutta le Liste Web non funziona, oppure che i siti Web in analisi sono altamente strutturati. Effettivamente, i siti sono altamente strutturati poiché, se così non fosse, l'esperto che ha creato la ground truth non avrebbe potuto assegnare manualmente le pagine ai vari Cluster.

In conclusione, la sperimentazione mostra come i migliori risultati sono stati ottenuti combinando l'informazione testuale con quella estratta dalla struttura del sito. Questo è più evidente per Illinois (Tabella 4.46) ed Oxford (Tabella 4.47), dove la struttura del sito e il testo delle pagine codificano informazioni complementari utili per il Clustering. Per Princeton (Tabella 4.48) l'informazione testuale è soddisfacente per realizzare Cluster di buona qualità; mentre per Stanford (Tabella 4.49) l'informazione codificata nella struttura del sito è sufficiente: si può supporre che il sito contenga pochi termini significativi oppure poco testo contenente informazioni utili.

4. Sperimentazione

Tabella 4.2: Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.26	0.12	0.28	0.37	-0.03	0.32
100K	3	10	0.35	0.24	0.37	0.45	-0.03	0.41
100K	5	10	0.60	0.62	0.62	0.65	-0.04	0.63
100K	7	10	0.58	0.61	0.59	0.65	-0.06	0.62
100K	2	15	0.26	0.12	0.28	0.35	-0.02	0.31
100K	3	15	0.46	0.30	0.48	0.63	-0.04	0.54
100K	5	15	0.56	0.52	0.58	0.69	-0.05	0.63
100K	7	15	0.54	0.49	0.55	0.68	-0.06	0.61
100K	2	20	0.26	0.11	0.28	0.37	-0.02	0.32
100K	3	20	0.38	0.24	0.40	0.51	-0.04	0.45
100K	5	20	0.43	0.23	0.45	0.57	-0.06	0.50
100K	7	20	0.45	0.30	0.47	0.56	-0.04	0.51
500K	2	10	0.23	0.09	0.26	0.34	-0.02	0.29
500K	3	10	0.53	0.35	0.54	0.71	-0.03	0.62
500K	5	10	0.61	0.48	0.62	0.77	-0.05	0.68
500K	7	10	0.47	0.29	0.49	0.61	-0.04	0.54
500K	2	15	0.24	0.10	0.26	0.35	-0.02	0.30
500K	3	15	0.59	0.41	0.60	0.80	-0.04	0.69
500K	5	15	0.59	0.41	0.60	0.80	-0.04	0.69
500K	7	15	0.50	0.29	0.51	0.69	-0.03	0.59
500K	2	20	0.20	0.09	0.23	0.30	-0.01	0.26
500K	3	20	0.61	0.42	0.63	0.82	-0.04	0.71
500K	5	20	0.61	0.39	0.63	0.83	-0.04	0.71
500K	7	20	0.62	0.42	0.63	0.83	-0.03	0.72
1M	2	10	0.24	0.10	0.27	0.35	-0.02	0.30
1M	3	10	0.58	0.38	0.59	0.79	-0.06	0.68
1M	5	10	0.57	0.37	0.59	0.76	-0.05	0.66
1M	7	10	0.52	0.32	0.53	0.70	-0.07	0.61
1M	2	15	0.24	0.10	0.26	0.34	-0.02	0.30
1M	3	15	0.60	0.38	0.61	0.82	-0.04	0.70
1M	5	15	0.56	0.36	0.58	0.77	-0.04	0.66
1M	7	15	0.56	0.34	0.58	0.77	-0.04	0.66
1M	2	20	0.23	0.09	0.26	0.34	-0.02	0.30
1M	3	20	0.61	0.38	0.62	0.83	-0.04	0.71
1M	5	20	0.61	0.40	0.62	0.83	-0.04	0.71
1M	7	20	0.60	0.37	0.62	0.84	-0.04	0.71

4. Sperimentazione

Tabella 4.3: Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.06	0.04	0.29	0.07	-0.01	0.12
100K	3	10	0.06	0.00	0.17	0.07	-0.02	0.10
100K	5	10	0.06	0.10	0.23	0.07	0.01	0.11
100K	7	10	0.14	0.16	0.30	0.15	-0.04	0.20
100K	2	15	0.11	0.08	0.39	0.11	-0.02	0.18
100K	3	15	0.08	0.11	0.22	0.09	0.00	0.13
100K	5	15	0.08	0.07	0.30	0.09	-0.05	0.14
100K	7	15	0.13	0.09	0.28	0.15	-0.03	0.19
100K	2	20	0.20	0.18	0.54	0.20	-0.01	0.30
100K	3	20	0.15	0.11	0.29	0.17	-0.04	0.21
100K	5	20	0.16	0.18	0.33	0.18	-0.02	0.23
100K	7	20	0.21	0.20	0.42	0.22	-0.01	0.29
500K	2	10	0.21	0.15	0.53	0.22	-0.02	0.31
500K	3	10	0.18	0.20	0.35	0.19	-0.04	0.25
500K	5	10	0.19	0.15	0.36	0.22	-0.11	0.27
500K	7	10	0.20	0.16	0.43	0.22	-0.15	0.29
500K	2	15	0.18	0.13	0.51	0.19	-0.01	0.27
500K	3	15	0.16	0.15	0.43	0.17	0.02	0.24
500K	5	15	0.05	0.01	0.23	0.06	0.01	0.09
500K	7	15	0.07	0.04	0.28	0.09	-0.04	0.13
500K	2	20	0.15	0.09	0.46	0.16	-0.01	0.24
500K	3	20	0.41	0.13	0.44	0.63	-0.11	0.52
500K	5	20	0.21	0.20	0.53	0.22	-0.01	0.31
500K	7	20	0.14	0.11	0.41	0.15	-0.02	0.22
1M	2	10	0.19	0.12	0.53	0.21	-0.03	0.30
1M	3	10	0.41	0.15	0.45	0.65	-0.15	0.53
1M	5	10	0.43	0.19	0.47	0.73	-0.14	0.57
1M	7	10	0.43	0.18	0.47	0.72	-0.15	0.56
1M	2	15	0.17	0.09	0.49	0.18	-0.03	0.26
1M	3	15	0.40	0.15	0.45	0.75	-0.15	0.56
1M	5	15	0.44	0.19	0.47	0.78	-0.14	0.59
1M	7	15	0.41	0.16	0.45	0.78	-0.14	0.57
1M	2	20	0.17	0.11	0.48	0.18	-0.02	0.27
1M	3	20	0.41	0.17	0.45	0.71	-0.10	0.55
1M	5	20	0.40	0.14	0.44	0.76	-0.13	0.56
1M	7	20	0.39	0.14	0.43	0.76	-0.15	0.55

4. Sperimentazione

Tabella 4.4: Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.53	0.33	0.55	0.73	-0.05	0.62
100K	3	10	0.61	0.51	0.62	0.79	-0.06	0.70
100K	5	10	0.55	0.38	0.57	0.76	-0.05	0.65
100K	7	10	0.54	0.36	0.55	0.73	-0.05	0.63
100K	2	15	0.64	0.44	0.65	0.86	-0.05	0.74
100K	3	15	0.59	0.41	0.60	0.80	-0.05	0.69
100K	5	15	0.60	0.42	0.61	0.82	-0.06	0.70
100K	7	15	0.56	0.39	0.57	0.76	-0.06	0.65
100K	2	20	0.60	0.38	0.61	0.82	-0.05	0.70
100K	3	20	0.56	0.41	0.58	0.75	-0.04	0.66
100K	5	20	0.35	0.13	0.37	0.47	-0.04	0.42
100K	7	20	0.31	0.10	0.34	0.43	-0.04	0.38
500K	2	10	0.65	0.48	0.67	0.87	-0.04	0.75
500K	3	10	0.62	0.40	0.63	0.85	-0.03	0.72
500K	5	10	0.66	0.46	0.67	0.89	-0.03	0.76
500K	7	10	0.56	0.29	0.58	0.77	-0.03	0.66
500K	2	15	0.60	0.36	0.62	0.83	-0.05	0.71
500K	3	15	0.58	0.33	0.59	0.80	-0.04	0.68
500K	5	15	0.61	0.38	0.62	0.83	-0.03	0.71
500K	7	15	0.55	0.33	0.57	0.77	-0.03	0.65
500K	2	20	0.65	0.43	0.66	0.86	-0.04	0.75
500K	3	20	0.58	0.36	0.59	0.80	-0.05	0.68
500K	5	20	0.58	0.34	0.59	0.79	-0.03	0.68
500K	7	20	0.57	0.37	0.58	0.78	-0.03	0.67
1M	2	10	0.61	0.37	0.62	0.83	-0.05	0.71
1M	3	10	0.63	0.40	0.64	0.86	-0.05	0.73
1M	5	10	0.62	0.40	0.64	0.85	-0.04	0.73
1M	7	10	0.61	0.37	0.62	0.84	-0.04	0.71
1M	2	15	0.61	0.38	0.62	0.84	-0.05	0.71
1M	3	15	0.60	0.36	0.61	0.83	-0.04	0.70
1M	5	15	0.60	0.37	0.61	0.82	-0.04	0.70
1M	7	15	0.58	0.36	0.59	0.80	-0.04	0.68
1M	2	20	0.58	0.35	0.60	0.80	-0.04	0.68
1M	3	20	0.64	0.42	0.65	0.85	-0.05	0.74
1M	5	20	0.61	0.39	0.62	0.82	-0.04	0.70
1M	7	20	0.57	0.36	0.58	0.77	-0.04	0.67

4. Sperimentazione

Tabella 4.5: Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.40	0.15	0.45	0.57	-0.12	0.50
100K	3	10	0.39	0.12	0.44	0.54	-0.12	0.48
100K	5	10	0.39	0.11	0.43	0.50	-0.12	0.46
100K	7	10	0.38	0.08	0.42	0.49	-0.11	0.45
100K	2	15	0.39	0.09	0.43	0.59	-0.08	0.49
100K	3	15	0.39	0.11	0.43	0.52	-0.08	0.47
100K	5	15	0.36	0.09	0.41	0.48	-0.11	0.44
100K	7	15	0.37	0.12	0.41	0.48	-0.10	0.44
100K	2	20	0.36	0.09	0.41	0.60	-0.17	0.49
100K	3	20	0.35	0.06	0.40	0.53	-0.15	0.46
100K	5	20	0.13	0.03	0.24	0.16	-0.13	0.20
100K	7	20	0.19	0.06	0.30	0.21	-0.09	0.25
500K	2	10	0.40	0.15	0.45	0.77	-0.11	0.56
500K	3	10	0.41	0.14	0.45	0.73	-0.10	0.55
500K	5	10	0.39	0.11	0.43	0.67	-0.09	0.53
500K	7	10	0.41	0.14	0.45	0.69	-0.08	0.54
500K	2	15	0.40	0.15	0.45	0.76	-0.11	0.56
500K	3	15	0.41	0.15	0.46	0.74	-0.09	0.56
500K	5	15	0.41	0.15	0.46	0.73	-0.08	0.56
500K	7	15	0.41	0.16	0.45	0.72	-0.08	0.55
500K	2	20	0.41	0.15	0.45	0.76	-0.12	0.56
500K	3	20	0.43	0.16	0.47	0.74	-0.10	0.57
500K	5	20	0.42	0.16	0.46	0.74	-0.09	0.57
500K	7	20	0.43	0.17	0.47	0.74	-0.08	0.58
1M	2	10	0.41	0.16	0.45	0.79	-0.12	0.58
1M	3	10	0.41	0.15	0.45	0.77	-0.13	0.57
1M	5	10	0.42	0.17	0.46	0.79	-0.10	0.59
1M	7	10	0.41	0.15	0.45	0.77	-0.11	0.57
1M	2	15	0.39	0.14	0.44	0.74	-0.15	0.55
1M	3	15	0.40	0.15	0.44	0.76	-0.12	0.56
1M	5	15	0.39	0.14	0.44	0.77	-0.13	0.56
1M	7	15	0.38	0.14	0.43	0.75	-0.13	0.55
1M	2	20	0.41	0.16	0.45	0.77	-0.14	0.57
1M	3	20	0.42	0.16	0.46	0.78	-0.13	0.58
1M	5	20	0.41	0.15	0.45	0.76	-0.10	0.57
1M	7	20	0.40	0.14	0.44	0.77	-0.09	0.56

4. Sperimentazione

Tabella 4.6: Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.34	0.10	0.37	0.43	-0.03	0.40
100K	3	10	0.37	0.20	0.40	0.45	-0.03	0.42
100K	5	10	0.41	0.24	0.43	0.47	-0.05	0.45
100K	7	10	0.43	0.27	0.46	0.46	-0.06	0.46
100K	2	15	0.34	0.14	0.36	0.46	-0.02	0.41
100K	3	15	0.44	0.30	0.46	0.53	-0.02	0.49
100K	5	15	0.48	0.41	0.50	0.55	-0.14	0.52
100K	7	15	0.54	0.50	0.56	0.60	-0.07	0.58
100K	2	20	0.34	0.15	0.36	0.48	-0.02	0.42
100K	3	20	0.51	0.41	0.53	0.65	-0.04	0.58
100K	5	20	0.55	0.41	0.56	0.68	-0.07	0.62
100K	7	20	0.55	0.51	0.57	0.62	-0.09	0.59
500K	2	10	0.43	0.24	0.45	0.59	-0.02	0.51
500K	3	10	0.53	0.39	0.55	0.67	-0.04	0.60
500K	5	10	0.53	0.35	0.55	0.72	-0.05	0.62
500K	7	10	0.47	0.28	0.49	0.63	-0.03	0.55
500K	2	15	0.52	0.32	0.54	0.71	-0.02	0.61
500K	3	15	0.57	0.40	0.58	0.77	-0.05	0.66
500K	5	15	0.58	0.39	0.59	0.79	-0.05	0.68
500K	7	15	0.52	0.35	0.53	0.69	-0.05	0.60
500K	2	20	0.55	0.37	0.57	0.73	-0.02	0.64
500K	3	20	0.58	0.40	0.59	0.78	-0.05	0.67
500K	5	20	0.55	0.36	0.56	0.75	-0.04	0.64
500K	7	20	0.51	0.32	0.53	0.71	-0.04	0.61
1M	2	10	0.59	0.40	0.60	0.75	-0.03	0.67
1M	3	10	0.62	0.53	0.64	0.80	-0.06	0.71
1M	5	10	0.61	0.44	0.62	0.82	-0.06	0.71
1M	7	10	0.60	0.43	0.61	0.81	-0.05	0.69
1M	2	15	0.58	0.36	0.59	0.77	-0.02	0.67
1M	3	15	0.64	0.47	0.65	0.85	-0.04	0.73
1M	5	15	0.53	0.33	0.55	0.73	-0.05	0.62
1M	7	15	0.50	0.30	0.52	0.70	-0.04	0.59
1M	2	20	0.56	0.33	0.57	0.76	-0.02	0.65
1M	3	20	0.54	0.33	0.56	0.75	-0.05	0.64
1M	5	20	0.51	0.29	0.53	0.71	-0.04	0.61
1M	7	20	0.51	0.30	0.53	0.71	-0.04	0.60

4. Sperimentazione

Tabella 4.7: Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.11	0.16	0.33	0.12	-0.02	0.18
100K	3	10	0.06	0.06	0.19	0.07	-0.04	0.10
100K	5	10	0.10	0.05	0.23	0.11	-0.11	0.15
100K	7	10	0.15	0.15	0.35	0.16	-0.14	0.22
100K	2	15	0.16	0.18	0.42	0.17	-0.03	0.24
100K	3	15	0.20	0.13	0.39	0.23	-0.08	0.29
100K	5	15	0.18	0.12	0.36	0.20	-0.02	0.25
100K	7	15	0.19	0.20	0.41	0.21	-0.03	0.27
100K	2	20	0.23	0.24	0.53	0.24	-0.02	0.33
100K	3	20	0.18	0.11	0.38	0.20	-0.15	0.26
100K	5	20	0.21	0.21	0.42	0.23	-0.14	0.29
100K	7	20	0.23	0.27	0.46	0.24	-0.12	0.31
500K	2	10	0.16	0.20	0.46	0.17	-0.02	0.25
500K	3	10	0.19	0.08	0.35	0.20	-0.10	0.26
500K	5	10	0.16	0.05	0.37	0.18	-0.10	0.24
500K	7	10	0.16	0.07	0.38	0.18	-0.11	0.24
500K	2	15	0.30	0.33	0.60	0.31	-0.01	0.41
500K	3	15	0.20	0.10	0.39	0.22	-0.02	0.28
500K	5	15	0.17	0.08	0.41	0.18	-0.04	0.25
500K	7	15	0.19	0.14	0.42	0.21	-0.02	0.28
500K	2	20	0.34	0.31	0.70	0.36	-0.02	0.47
500K	3	20	0.53	0.29	0.64	0.55	-0.06	0.59
500K	5	20	0.48	0.36	0.63	0.49	-0.04	0.55
500K	7	20	0.53	0.47	0.64	0.54	-0.04	0.59
1M	2	10	0.48	0.39	0.69	0.50	-0.05	0.58
1M	3	10	0.50	0.29	0.53	0.52	-0.12	0.53
1M	5	10	0.55	0.44	0.57	0.74	-0.11	0.65
1M	7	10	0.54	0.44	0.57	0.67	-0.11	0.61
1M	2	15	0.49	0.41	0.71	0.51	-0.05	0.60
1M	3	15	0.54	0.31	0.56	0.72	-0.09	0.63
1M	5	15	0.49	0.31	0.52	0.78	-0.10	0.62
1M	7	15	0.45	0.20	0.48	0.75	-0.10	0.58
1M	2	20	0.60	0.52	0.76	0.61	-0.03	0.68
1M	3	20	0.47	0.25	0.51	0.76	-0.11	0.61
1M	5	20	0.45	0.21	0.49	0.74	-0.10	0.59
1M	7	20	0.42	0.17	0.46	0.74	-0.10	0.57

4. Sperimentazione

Tabella 4.8: Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.65	0.50	0.66	0.83	-0.05	0.74
100K	3	10	0.65	0.57	0.67	0.81	-0.05	0.73
100K	5	10	0.66	0.55	0.68	0.82	-0.05	0.74
100K	7	10	0.59	0.49	0.60	0.75	-0.05	0.67
100K	2	15	0.64	0.46	0.66	0.83	-0.04	0.73
100K	3	15	0.60	0.38	0.61	0.79	-0.04	0.69
100K	5	15	0.60	0.37	0.61	0.79	-0.05	0.69
100K	7	15	0.60	0.42	0.62	0.77	-0.05	0.68
100K	2	20	0.69	0.61	0.70	0.87	-0.04	0.78
100K	3	20	0.61	0.45	0.62	0.78	-0.04	0.69
100K	5	20	0.58	0.41	0.60	0.77	-0.04	0.67
100K	7	20	0.62	0.47	0.63	0.81	-0.03	0.71
500K	2	10	0.67	0.47	0.69	0.90	-0.03	0.78
500K	3	10	0.62	0.38	0.63	0.84	-0.03	0.72
500K	5	10	0.66	0.46	0.67	0.88	-0.02	0.76
500K	7	10	0.58	0.31	0.60	0.75	-0.02	0.66
500K	2	15	0.65	0.44	0.66	0.89	-0.03	0.76
500K	3	15	0.66	0.45	0.67	0.89	-0.03	0.77
500K	5	15	0.60	0.36	0.61	0.83	-0.02	0.70
500K	7	15	0.56	0.29	0.58	0.76	-0.02	0.66
500K	2	20	0.61	0.38	0.63	0.84	-0.04	0.72
500K	3	20	0.64	0.42	0.65	0.87	-0.03	0.74
500K	5	20	0.58	0.36	0.59	0.79	-0.03	0.68
500K	7	20	0.58	0.36	0.60	0.81	-0.02	0.69
1M	2	10	0.62	0.37	0.63	0.85	-0.04	0.73
1M	3	10	0.61	0.36	0.62	0.83	-0.03	0.71
1M	5	10	0.64	0.43	0.65	0.88	-0.03	0.75
1M	7	10	0.62	0.38	0.63	0.85	-0.03	0.73
1M	2	15	0.63	0.42	0.64	0.87	-0.04	0.74
1M	3	15	0.61	0.36	0.62	0.84	-0.04	0.71
1M	5	15	0.59	0.33	0.60	0.81	-0.03	0.69
1M	7	15	0.60	0.37	0.61	0.83	-0.03	0.71
1M	2	20	0.63	0.38	0.64	0.86	-0.04	0.73
1M	3	20	0.59	0.37	0.60	0.82	-0.03	0.70
1M	5	20	0.61	0.37	0.62	0.83	-0.03	0.71
1M	7	20	0.59	0.35	0.60	0.81	-0.03	0.69

4. Sperimentazione

Tabella 4.9: Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Illinois

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.40	0.09	0.45	0.46	-0.09	0.46
100K	3	10	0.32	0.02	0.40	0.37	-0.10	0.39
100K	5	10	0.27	0.02	0.40	0.32	-0.09	0.36
100K	7	10	0.28	0.03	0.42	0.33	-0.09	0.37
100K	2	15	0.46	0.15	0.49	0.56	-0.08	0.53
100K	3	15	0.39	0.08	0.45	0.43	-0.08	0.44
100K	5	15	0.34	0.08	0.45	0.38	-0.07	0.42
100K	7	15	0.30	0.05	0.44	0.34	-0.07	0.38
100K	2	20	0.41	0.07	0.45	0.51	-0.11	0.48
100K	3	20	0.36	0.02	0.41	0.43	-0.12	0.42
100K	5	20	0.33	0.02	0.40	0.38	-0.10	0.39
100K	7	20	0.30	0.01	0.39	0.35	-0.09	0.37
500K	2	10	0.42	0.16	0.46	0.69	-0.09	0.55
500K	3	10	0.40	0.14	0.45	0.66	-0.08	0.53
500K	5	10	0.41	0.14	0.45	0.68	-0.08	0.54
500K	7	10	0.39	0.08	0.43	0.58	-0.06	0.49
500K	2	15	0.43	0.17	0.47	0.75	-0.09	0.58
500K	3	15	0.39	0.13	0.44	0.69	-0.09	0.54
500K	5	15	0.41	0.12	0.45	0.66	-0.07	0.54
500K	7	15	0.42	0.13	0.46	0.67	-0.06	0.55
500K	2	20	0.41	0.18	0.46	0.75	-0.09	0.57
500K	3	20	0.39	0.12	0.44	0.70	-0.08	0.54
500K	5	20	0.40	0.13	0.44	0.69	-0.07	0.54
500K	7	20	0.41	0.17	0.46	0.73	-0.07	0.56
1M	2	10	0.39	0.15	0.44	0.77	-0.11	0.56
1M	3	10	0.39	0.14	0.43	0.76	-0.11	0.55
1M	5	10	0.38	0.14	0.43	0.79	-0.11	0.56
1M	7	10	0.38	0.13	0.43	0.75	-0.10	0.55
1M	2	15	0.38	0.14	0.43	0.77	-0.11	0.55
1M	3	15	0.39	0.14	0.44	0.78	-0.11	0.56
1M	5	15	0.39	0.14	0.44	0.78	-0.10	0.56
1M	7	15	0.39	0.13	0.44	0.77	-0.10	0.56
1M	2	20	0.37	0.13	0.42	0.76	-0.13	0.54
1M	3	20	0.39	0.14	0.44	0.77	-0.12	0.56
1M	5	20	0.39	0.14	0.44	0.78	-0.11	0.56
1M	7	20	0.40	0.15	0.45	0.79	-0.11	0.57

4. Sperimentazione

Tabella 4.10: Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Illinois

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
leftSkipgram-lc	KMeans	0.62	0.42	0.63	0.83	-0.03	0.72
leftSkipgram-lc	HDBScan	0.44	0.19	0.47	0.78	-0.14	0.59
normalSkipgram-lc	KMeans	0.66	0.46	0.67	0.89	-0.03	0.76
normalSkipgram-lc	HDBScan	0.43	0.17	0.47	0.74	-0.08	0.58
LINE-1-lc	KMeans	0.48	0.31	0.50	0.64	-0.03	0.56
LINE-1-lc	HDBScan	0.52	0.34	0.58	0.56	-0.09	0.57
LINE-2-lc	KMeans	0.30	0.13	0.34	0.42	-0.05	0.37
LINE-2-lc	HDBScan	0.19	-0.01	0.34	0.24	-0.12	0.28
leftSkipgram-nc	KMeans	0.64	0.47	0.65	0.85	-0.04	0.73
leftSkipgram-nc	HDBScan	0.60	0.52	0.76	0.61	-0.03	0.68
normalSkipgram-nc	KMeans	0.69	0.61	0.70	0.87	-0.04	0.78
normalSkipgram-nc	HDBScan	0.46	0.15	0.49	0.56	-0.08	0.53
LINE-1-nc	KMeans	0.51	0.29	0.52	0.66	-0.04	0.58
LINE-1-nc	HDBScan	0.39	0.18	0.44	0.70	-0.10	0.54
LINE-2-nc	KMeans	0.48	0.29	0.50	0.63	-0.05	0.55
LINE-2-nc	HDBScan	0.45	0.18	0.49	0.50	-0.09	0.49

	Clustering	DB size	Window	RW len.
leftSkipgram-lc	KMeans	500K	7	20
leftSkipgram-lc	HDBScan	1M	5	15
normalSkipgram-lc	KMeans	500K	5	10
normalSkipgram-lc	HDBScan	500K	7	20
leftSkipgram-nc	KMeans	1M	3	15
leftSkipgram-nc	HDBScan	1M	2	20
normalSkipgram-nc	KMeans	100K	2	20
normalSkipgram-nc	HDBScan	100K	2	15

Tabella 4.11: Risultati di Doc2Vec e TF-IDF in Illinois

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
Doc2Vec	KMeans	0.52	0.33	0.54	0.71	-0.09	0.61
Doc2Vec	HDBScan	0.76	0.83	0.78	0.77	-0.03	0.77
TF-IDF	KMeans	0.62	0.43	0.63	0.84	-0.02	0.72
TF-IDF	HDBScan	0.54	0.40	0.57	0.69	-0.07	0.62

Tabella 4.12: Risultati della configurazione Combinato in Illinois

ListConstraint	Clustering	AMI	ARI	Com	Hom	Silh	V-M
left + Doc2Vec	KMeans	0.60	0.41	0.61	0.81	-0.03	0.70
left + Doc2Vec	HDBScan	0.43	0.19	0.47	0.77	-0.15	0.58
left + TF-IDF	KMeans	0.69	0.49	0.70	0.92	-0.03	0.80
left + TF-IDF	HDBScan	0.76	0.72	0.77	0.82	-0.02	0.79
normal + Doc2Vec	KMeans	0.66	0.47	0.67	0.88	-0.03	0.76
normal + Doc2Vec	HDBScan	0.43	0.19	0.47	0.73	-0.08	0.57
normal + TF-IDF	KMeans	0.69	0.49	0.70	0.92	-0.02	0.79
normal + TF-IDF	HDBScan	0.55	0.36	0.58	0.77	-0.04	0.66
NoConstraint							
left + Doc2Vec	KMeans	0.67	0.48	0.68	0.88	-0.04	0.77
left + Doc2Vec	HDBScan	0.61	0.53	0.78	0.62	-0.03	0.69
left + TF-IDF	KMeans	0.67	0.54	0.68	0.87	-0.03	0.76
left + TF-IDF	HDBScan	0.78	0.76	0.83	0.79	-0.03	0.81
normal + Doc2Vec	KMeans	0.65	0.47	0.66	0.88	-0.04	0.76
normal + Doc2Vec	HDBScan	0.45	0.17	0.48	0.64	-0.10	0.55
normal + TF-IDF	KMeans	0.68	0.47	0.69	0.92	-0.03	0.79
normal + TF-IDF	HDBScan	0.77	0.68	0.78	0.83	-0.01	0.80

4. Sperimentazione

Tabella 4.13: Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.21	0.25	0.23	0.25	-0.08	0.24
100K	3	10	0.35	0.28	0.37	0.39	-0.04	0.38
100K	5	10	0.48	0.43	0.49	0.54	-0.07	0.51
100K	7	10	0.51	0.44	0.52	0.57	-0.09	0.54
100K	2	15	0.24	0.29	0.26	0.29	-0.09	0.27
100K	3	15	0.41	0.34	0.42	0.48	-0.09	0.45
100K	5	15	0.51	0.43	0.52	0.57	-0.11	0.54
100K	7	15	0.48	0.34	0.50	0.50	-0.09	0.50
100K	2	20	0.24	0.28	0.26	0.30	-0.03	0.28
100K	3	20	0.48	0.41	0.49	0.57	-0.06	0.53
100K	5	20	0.51	0.43	0.52	0.60	-0.06	0.56
100K	7	20	0.51	0.44	0.52	0.59	-0.10	0.56
500K	2	10	0.29	0.29	0.30	0.37	-0.03	0.33
500K	3	10	0.43	0.32	0.44	0.54	-0.02	0.49
500K	5	10	0.46	0.32	0.47	0.57	-0.03	0.52
500K	7	10	0.47	0.34	0.48	0.55	-0.05	0.51
500K	2	15	0.30	0.31	0.31	0.38	-0.03	0.34
500K	3	15	0.48	0.39	0.49	0.60	-0.03	0.54
500K	5	15	0.48	0.36	0.49	0.58	-0.02	0.53
500K	7	15	0.43	0.30	0.44	0.51	-0.02	0.48
500K	2	20	0.29	0.27	0.31	0.38	-0.02	0.34
500K	3	20	0.47	0.37	0.48	0.59	-0.03	0.53
500K	5	20	0.45	0.31	0.46	0.54	-0.03	0.50
500K	7	20	0.44	0.30	0.45	0.52	-0.03	0.48
1M	2	10	0.31	0.27	0.32	0.41	-0.02	0.36
1M	3	10	0.51	0.40	0.52	0.62	-0.06	0.56
1M	5	10	0.51	0.40	0.52	0.61	-0.04	0.56
1M	7	10	0.50	0.38	0.51	0.62	-0.04	0.56
1M	2	15	0.31	0.26	0.32	0.40	-0.03	0.36
1M	3	15	0.50	0.37	0.51	0.60	-0.05	0.55
1M	5	15	0.50	0.36	0.51	0.62	-0.03	0.56
1M	7	15	0.51	0.46	0.52	0.62	-0.03	0.57
1M	2	20	0.30	0.27	0.32	0.39	-0.02	0.35
1M	3	20	0.52	0.40	0.53	0.64	-0.04	0.58
1M	5	20	0.50	0.39	0.51	0.63	-0.02	0.56
1M	7	20	0.51	0.46	0.52	0.62	-0.04	0.57

4. Sperimentazione

Tabella 4.14: Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.09	0.04	0.31	0.10	-0.05	0.15
100K	3	10	0.14	0.07	0.39	0.15	-0.11	0.22
100K	5	10	0.16	0.02	0.42	0.18	-0.16	0.25
100K	7	10	0.05	0.04	0.16	0.05	-0.10	0.08
100K	2	15	0.15	0.11	0.48	0.16	-0.15	0.24
100K	3	15	0.20	0.09	0.50	0.22	-0.09	0.30
100K	5	15	0.22	0.08	0.47	0.24	-0.18	0.32
100K	7	15	0.26	0.12	0.51	0.28	-0.25	0.37
100K	2	20	0.16	0.09	0.50	0.16	-0.11	0.25
100K	3	20	0.26	0.13	0.56	0.28	-0.12	0.37
100K	5	20	0.29	0.14	0.51	0.31	-0.24	0.39
100K	7	20	0.30	0.14	0.53	0.33	-0.30	0.41
500K	2	10	0.19	0.12	0.59	0.20	-0.08	0.30
500K	3	10	0.19	0.06	0.48	0.21	-0.12	0.29
500K	5	10	0.18	0.05	0.48	0.19	-0.12	0.28
500K	7	10	0.13	0.03	0.45	0.15	-0.12	0.22
500K	2	15	0.18	0.11	0.57	0.19	-0.06	0.29
500K	3	15	0.24	0.07	0.49	0.27	-0.15	0.35
500K	5	15	0.18	0.05	0.47	0.20	-0.14	0.28
500K	7	15	0.13	0.04	0.48	0.15	-0.13	0.23
500K	2	20	0.19	0.11	0.60	0.20	-0.05	0.30
500K	3	20	0.26	0.08	0.47	0.29	-0.14	0.36
500K	5	20	0.21	0.06	0.48	0.24	-0.17	0.32
500K	7	20	0.14	0.04	0.44	0.16	-0.13	0.24
1M	2	10	0.24	0.15	0.53	0.26	-0.12	0.35
1M	3	10	0.28	0.10	0.45	0.32	-0.16	0.37
1M	5	10	0.29	0.11	0.46	0.32	-0.16	0.38
1M	7	10	0.30	0.11	0.50	0.32	-0.19	0.39
1M	2	15	0.23	0.10	0.56	0.24	-0.05	0.33
1M	3	15	0.34	0.11	0.45	0.38	-0.15	0.41
1M	5	15	0.35	0.12	0.47	0.38	-0.19	0.42
1M	7	15	0.32	0.13	0.48	0.35	-0.20	0.41
1M	2	20	0.08	0.09	0.27	0.08	-0.03	0.13
1M	3	20	0.37	0.13	0.44	0.42	-0.17	0.43
1M	5	20	0.38	0.13	0.47	0.42	-0.14	0.44
1M	7	20	0.37	0.22	0.49	0.40	-0.14	0.44

4. Sperimentazione

Tabella 4.15: Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.53	0.33	0.54	0.65	-0.02	0.59
100K	3	10	0.49	0.31	0.50	0.63	-0.03	0.56
100K	5	10	0.50	0.30	0.51	0.60	-0.05	0.55
100K	7	10	0.48	0.30	0.49	0.58	-0.05	0.53
100K	2	15	0.53	0.32	0.54	0.65	-0.03	0.59
100K	3	15	0.50	0.35	0.51	0.63	-0.03	0.56
100K	5	15	0.49	0.32	0.50	0.60	-0.02	0.55
100K	7	15	0.44	0.24	0.45	0.53	-0.03	0.49
100K	2	20	0.51	0.35	0.52	0.64	-0.03	0.57
100K	3	20	0.49	0.35	0.50	0.60	-0.03	0.55
100K	5	20	0.46	0.29	0.47	0.56	-0.03	0.51
100K	7	20	0.47	0.28	0.48	0.58	-0.04	0.52
500K	2	10	0.52	0.42	0.53	0.66	-0.02	0.59
500K	3	10	0.51	0.39	0.52	0.64	-0.01	0.57
500K	5	10	0.48	0.36	0.49	0.60	-0.02	0.54
500K	7	10	0.50	0.37	0.51	0.62	-0.01	0.56
500K	2	15	0.53	0.41	0.54	0.66	-0.02	0.59
500K	3	15	0.52	0.48	0.52	0.63	-0.02	0.57
500K	5	15	0.49	0.40	0.50	0.62	-0.01	0.55
500K	7	15	0.50	0.46	0.51	0.60	-0.02	0.55
500K	2	20	0.53	0.41	0.53	0.66	-0.02	0.59
500K	3	20	0.51	0.44	0.52	0.63	-0.02	0.57
500K	5	20	0.49	0.38	0.50	0.62	-0.02	0.55
500K	7	20	0.47	0.34	0.47	0.59	-0.01	0.53
1M	2	10	0.52	0.44	0.53	0.65	-0.02	0.58
1M	3	10	0.53	0.50	0.54	0.65	-0.02	0.59
1M	5	10	0.53	0.48	0.54	0.65	-0.02	0.59
1M	7	10	0.52	0.43	0.53	0.65	-0.02	0.58
1M	2	15	0.53	0.41	0.54	0.67	-0.02	0.60
1M	3	15	0.51	0.39	0.52	0.65	-0.02	0.58
1M	5	15	0.51	0.45	0.51	0.63	-0.02	0.57
1M	7	15	0.52	0.48	0.53	0.65	-0.02	0.58
1M	2	20	0.51	0.41	0.52	0.65	-0.02	0.58
1M	3	20	0.51	0.42	0.52	0.64	-0.02	0.58
1M	5	20	0.50	0.44	0.51	0.63	-0.02	0.56
1M	7	20	0.51	0.48	0.52	0.63	-0.02	0.57

4. Sperimentazione

Tabella 4.16: Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.31	0.08	0.45	0.35	-0.14	0.40
100K	3	10	0.27	0.06	0.45	0.30	-0.16	0.36
100K	5	10	0.22	0.06	0.46	0.25	-0.18	0.32
100K	7	10	0.21	0.05	0.46	0.24	-0.19	0.32
100K	2	15	0.28	0.06	0.40	0.33	-0.16	0.36
100K	3	15	0.27	0.06	0.42	0.30	-0.17	0.35
100K	5	15	0.23	0.05	0.41	0.27	-0.19	0.33
100K	7	15	0.22	0.05	0.43	0.26	-0.19	0.32
100K	2	20	0.30	0.07	0.40	0.35	-0.17	0.37
100K	3	20	0.28	0.06	0.42	0.32	-0.17	0.36
100K	5	20	0.23	0.05	0.42	0.26	-0.19	0.32
100K	7	20	0.22	0.05	0.43	0.25	-0.19	0.32
500K	2	10	0.30	0.09	0.49	0.33	-0.09	0.40
500K	3	10	0.25	0.07	0.51	0.27	-0.08	0.36
500K	5	10	0.14	0.04	0.46	0.17	-0.06	0.25
500K	7	10	0.10	0.02	0.41	0.13	-0.08	0.19
500K	2	15	0.33	0.09	0.46	0.36	-0.09	0.41
500K	3	15	0.29	0.09	0.50	0.32	-0.09	0.39
500K	5	15	0.16	0.04	0.44	0.19	-0.08	0.27
500K	7	15	0.14	0.03	0.45	0.16	-0.08	0.24
500K	2	20	0.31	0.09	0.45	0.35	-0.13	0.40
500K	3	20	0.30	0.09	0.50	0.33	-0.09	0.40
500K	5	20	0.25	0.08	0.50	0.28	-0.09	0.36
500K	7	20	0.00	0.00	0.31	0.01	-0.03	0.01
1M	2	10	0.39	0.17	0.43	0.55	-0.16	0.49
1M	3	10	0.42	0.18	0.46	0.57	-0.15	0.51
1M	5	10	0.41	0.16	0.45	0.52	-0.15	0.48
1M	7	10	0.41	0.15	0.45	0.51	-0.13	0.48
1M	2	15	0.37	0.14	0.41	0.52	-0.15	0.46
1M	3	15	0.41	0.18	0.45	0.55	-0.14	0.50
1M	5	15	0.42	0.20	0.46	0.56	-0.14	0.50
1M	7	15	0.43	0.20	0.47	0.56	-0.11	0.51
1M	2	20	0.40	0.23	0.45	0.58	-0.14	0.51
1M	3	20	0.41	0.22	0.45	0.58	-0.14	0.51
1M	5	20	0.40	0.17	0.44	0.55	-0.12	0.49
1M	7	20	0.42	0.19	0.45	0.56	-0.12	0.50

4. Sperimentazione

Tabella 4.17: Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.24	0.29	0.26	0.27	-0.04	0.26
100K	3	10	0.42	0.43	0.43	0.47	-0.04	0.45
100K	5	10	0.53	0.51	0.54	0.59	-0.07	0.56
100K	7	10	0.54	0.47	0.55	0.56	-0.12	0.55
100K	2	15	0.26	0.29	0.28	0.31	-0.03	0.29
100K	3	15	0.46	0.45	0.48	0.56	-0.04	0.51
100K	5	15	0.50	0.45	0.51	0.58	-0.06	0.54
100K	7	15	0.52	0.42	0.53	0.56	-0.10	0.55
100K	2	20	0.26	0.29	0.28	0.32	-0.03	0.30
100K	3	20	0.46	0.36	0.47	0.52	-0.04	0.50
100K	5	20	0.52	0.44	0.53	0.59	-0.05	0.56
100K	7	20	0.53	0.44	0.54	0.59	-0.16	0.56
500K	2	10	0.31	0.29	0.32	0.38	-0.03	0.35
500K	3	10	0.44	0.31	0.45	0.56	-0.02	0.50
500K	5	10	0.48	0.35	0.49	0.59	-0.03	0.54
500K	7	10	0.46	0.29	0.47	0.56	-0.05	0.51
500K	2	15	0.29	0.24	0.31	0.38	-0.02	0.34
500K	3	15	0.46	0.30	0.47	0.57	-0.02	0.51
500K	5	15	0.45	0.31	0.46	0.55	-0.02	0.50
500K	7	15	0.46	0.33	0.47	0.55	-0.04	0.51
500K	2	20	0.32	0.29	0.33	0.40	-0.02	0.36
500K	3	20	0.49	0.34	0.49	0.61	-0.02	0.54
500K	5	20	0.50	0.33	0.51	0.61	-0.02	0.55
500K	7	20	0.47	0.33	0.48	0.56	-0.23	0.52
1M	2	10	0.34	0.31	0.35	0.42	-0.03	0.38
1M	3	10	0.51	0.48	0.52	0.62	-0.04	0.57
1M	5	10	0.56	0.40	0.57	0.67	-0.05	0.62
1M	7	10	0.55	0.43	0.56	0.66	-0.05	0.61
1M	2	15	0.33	0.32	0.34	0.42	-0.02	0.38
1M	3	15	0.52	0.36	0.52	0.64	-0.03	0.58
1M	5	15	0.56	0.43	0.57	0.69	-0.03	0.62
1M	7	15	0.56	0.45	0.57	0.67	-0.03	0.62
1M	2	20	0.32	0.29	0.33	0.40	-0.03	0.36
1M	3	20	0.54	0.39	0.55	0.67	-0.04	0.61
1M	5	20	0.58	0.52	0.59	0.68	-0.04	0.63
1M	7	20	0.53	0.41	0.53	0.66	-0.04	0.59

4. Sperimentazione

Tabella 4.18: Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.14	0.09	0.41	0.15	-0.21	0.22
100K	3	10	0.16	0.09	0.44	0.16	-0.05	0.24
100K	5	10	0.03	0.03	0.18	0.03	-0.03	0.06
100K	7	10	0.06	0.07	0.20	0.06	-0.21	0.09
100K	2	15	0.14	0.08	0.45	0.15	-0.16	0.22
100K	3	15	0.19	0.08	0.51	0.20	-0.18	0.29
100K	5	15	0.02	0.01	0.16	0.02	-0.02	0.04
100K	7	15	0.02	0.01	0.21	0.03	-0.03	0.05
100K	2	20	0.17	0.11	0.49	0.17	-0.16	0.26
100K	3	20	0.21	0.11	0.53	0.23	-0.07	0.32
100K	5	20	0.24	0.12	0.50	0.26	-0.16	0.35
100K	7	20	0.28	0.13	0.52	0.30	-0.27	0.38
500K	2	10	0.11	0.12	0.35	0.11	-0.02	0.17
500K	3	10	0.12	0.03	0.44	0.14	-0.17	0.22
500K	5	10	0.16	0.04	0.49	0.17	-0.09	0.26
500K	7	10	0.02	-0.00	0.24	0.02	-0.06	0.04
500K	2	15	0.18	0.09	0.52	0.20	-0.11	0.29
500K	3	15	0.17	0.04	0.44	0.19	-0.10	0.27
500K	5	15	0.14	0.04	0.47	0.16	-0.13	0.24
500K	7	15	0.08	0.05	0.24	0.08	-0.05	0.12
500K	2	20	0.22	0.12	0.58	0.22	-0.07	0.32
500K	3	20	0.21	0.06	0.46	0.24	-0.16	0.31
500K	5	20	0.09	0.06	0.27	0.09	-0.08	0.14
500K	7	20	0.05	0.02	0.22	0.05	-0.01	0.09
1M	2	10	0.28	0.16	0.55	0.29	-0.06	0.38
1M	3	10	0.05	0.01	0.27	0.05	-0.02	0.09
1M	5	10	0.32	0.13	0.49	0.35	-0.22	0.40
1M	7	10	0.11	0.03	0.28	0.13	-0.14	0.18
1M	2	15	0.12	0.13	0.38	0.12	-0.03	0.18
1M	3	15	0.33	0.11	0.46	0.36	-0.14	0.40
1M	5	15	0.35	0.12	0.48	0.38	-0.17	0.42
1M	7	15	0.31	0.10	0.49	0.34	-0.18	0.40
1M	2	20	0.11	0.13	0.36	0.12	-0.02	0.17
1M	3	20	0.38	0.12	0.47	0.41	-0.16	0.44
1M	5	20	0.33	0.13	0.47	0.36	-0.17	0.41
1M	7	20	0.32	0.18	0.46	0.34	-0.14	0.40

4. Sperimentazione

Tabella 4.19: Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.51	0.36	0.52	0.64	-0.02	0.58
100K	3	10	0.49	0.27	0.50	0.59	-0.03	0.54
100K	5	10	0.47	0.26	0.48	0.58	-0.03	0.53
100K	7	10	0.48	0.29	0.49	0.60	-0.05	0.54
100K	2	15	0.54	0.34	0.55	0.67	-0.02	0.60
100K	3	15	0.51	0.32	0.52	0.63	-0.03	0.57
100K	5	15	0.48	0.31	0.49	0.58	-0.07	0.53
100K	7	15	0.45	0.29	0.46	0.53	-0.05	0.50
100K	2	20	0.54	0.35	0.55	0.68	-0.03	0.61
100K	3	20	0.51	0.35	0.52	0.62	-0.03	0.56
100K	5	20	0.47	0.31	0.48	0.58	-0.06	0.53
100K	7	20	0.47	0.34	0.48	0.56	-0.03	0.51
500K	2	10	0.55	0.44	0.56	0.68	-0.02	0.62
500K	3	10	0.54	0.49	0.55	0.66	-0.02	0.60
500K	5	10	0.52	0.47	0.53	0.63	-0.02	0.57
500K	7	10	0.52	0.41	0.53	0.64	-0.02	0.58
500K	2	15	0.53	0.39	0.54	0.67	-0.02	0.60
500K	3	15	0.52	0.39	0.53	0.65	-0.01	0.59
500K	5	15	0.50	0.46	0.51	0.60	-0.02	0.55
500K	7	15	0.50	0.46	0.51	0.60	-0.01	0.55
500K	2	20	0.51	0.36	0.52	0.66	-0.02	0.58
500K	3	20	0.52	0.49	0.53	0.63	-0.02	0.57
500K	5	20	0.49	0.38	0.50	0.62	-0.02	0.56
500K	7	20	0.48	0.43	0.49	0.57	-0.05	0.53
1M	2	10	0.54	0.44	0.55	0.67	-0.02	0.60
1M	3	10	0.56	0.50	0.57	0.69	-0.02	0.62
1M	5	10	0.54	0.51	0.55	0.66	-0.03	0.60
1M	7	10	0.54	0.50	0.55	0.66	-0.02	0.60
1M	2	15	0.55	0.43	0.56	0.69	-0.02	0.62
1M	3	15	0.54	0.48	0.55	0.67	-0.02	0.60
1M	5	15	0.51	0.47	0.52	0.64	-0.02	0.57
1M	7	15	0.53	0.48	0.54	0.66	-0.02	0.59
1M	2	20	0.54	0.50	0.55	0.67	-0.02	0.61
1M	3	20	0.55	0.50	0.56	0.68	-0.02	0.61
1M	5	20	0.53	0.47	0.53	0.65	-0.02	0.59
1M	7	20	0.51	0.43	0.52	0.62	-0.02	0.57

4. Sperimentazione

Tabella 4.20: Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Oxford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.27	0.06	0.45	0.30	-0.12	0.36
100K	3	10	0.26	0.07	0.45	0.29	-0.15	0.35
100K	5	10	0.06	0.04	0.19	0.06	-0.07	0.09
100K	7	10	0.06	0.04	0.20	0.06	-0.04	0.10
100K	2	15	0.29	0.07	0.43	0.32	-0.14	0.37
100K	3	15	0.25	0.05	0.43	0.28	-0.22	0.34
100K	5	15	0.07	0.03	0.22	0.07	-0.21	0.11
100K	7	15	0.06	0.04	0.20	0.07	-0.13	0.10
100K	2	20	0.31	0.06	0.42	0.35	-0.14	0.38
100K	3	20	0.26	0.05	0.44	0.29	-0.15	0.35
100K	5	20	0.23	0.05	0.44	0.26	-0.17	0.33
100K	7	20	0.20	0.04	0.44	0.23	-0.16	0.30
500K	2	10	0.21	0.05	0.47	0.24	-0.10	0.32
500K	3	10	0.18	0.04	0.48	0.20	-0.10	0.28
500K	5	10	0.12	0.03	0.46	0.14	-0.08	0.22
500K	7	10	0.09	0.02	0.44	0.11	-0.08	0.18
500K	2	15	0.26	0.07	0.50	0.29	-0.11	0.37
500K	3	15	0.25	0.08	0.54	0.28	-0.11	0.37
500K	5	15	0.16	0.04	0.48	0.18	-0.10	0.27
500K	7	15	0.13	0.03	0.39	0.15	-0.10	0.21
500K	2	20	0.30	0.08	0.46	0.33	-0.10	0.39
500K	3	20	0.22	0.06	0.47	0.25	-0.09	0.33
500K	5	20	0.20	0.06	0.50	0.22	-0.08	0.31
500K	7	20	0.05	0.00	0.38	0.06	-0.00	0.10
1M	2	10	0.39	0.15	0.43	0.54	-0.14	0.48
1M	3	10	0.40	0.13	0.44	0.51	-0.14	0.47
1M	5	10	0.41	0.16	0.45	0.51	-0.13	0.48
1M	7	10	0.41	0.12	0.45	0.48	-0.12	0.46
1M	2	15	0.39	0.16	0.43	0.55	-0.14	0.49
1M	3	15	0.40	0.16	0.44	0.53	-0.14	0.48
1M	5	15	0.40	0.12	0.43	0.45	-0.12	0.44
1M	7	15	0.40	0.14	0.44	0.50	-0.12	0.47
1M	2	20	0.39	0.15	0.43	0.55	-0.16	0.48
1M	3	20	0.41	0.15	0.45	0.55	-0.15	0.49
1M	5	20	0.40	0.13	0.44	0.48	-0.14	0.46
1M	7	20	0.40	0.14	0.44	0.51	-0.11	0.47

Tabella 4.21: Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Oxford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
leftSkipgram-lc	KMeans	0.52	0.40	0.53	0.64	-0.04	0.58
leftSkipgram-lc	HDBScan	0.38	0.13	0.47	0.42	-0.14	0.44
normalSkipgram-lc	KMeans	0.53	0.41	0.54	0.67	-0.02	0.60
normalSkipgram-lc	HDBScan	0.40	0.23	0.45	0.58	-0.14	0.51
LINE-1-lc	KMeans	0.30	0.14	0.31	0.39	-0.05	0.35
LINE-1-lc	HDBScan	0.33	0.03	0.37	0.39	-0.13	0.38
LINE-2-lc	KMeans	0.24	0.16	0.26	0.33	-0.06	0.29
LINE-2-lc	HDBScan	0.20	-0.00	0.34	0.24	-0.17	0.28
leftSkipgram-nc	KMeans	0.58	0.52	0.59	0.68	-0.04	0.63
leftSkipgram-nc	HDBScan	0.38	0.12	0.47	0.41	-0.16	0.44
normalSkipgram-nc	KMeans	0.56	0.50	0.57	0.69	-0.02	0.62
normalSkipgram-nc	HDBScan	0.41	0.15	0.45	0.55	-0.15	0.49
LINE-1-nc	KMeans	0.30	0.14	0.31	0.38	-0.06	0.34
LINE-1-nc	HDBScan	0.15	0.03	0.34	0.17	-0.16	0.23
LINE-2-nc	KMeans	0.26	0.24	0.27	0.32	-0.10	0.29
LINE-2-nc	HDBScan	0.19	0.16	0.41	0.21	-0.17	0.27

	Clustering	DB size	Window	RW len.
leftSkipgram-lc	KMeans	1M	3	20
leftSkipgram-lc	HDBScan	1M	5	20
normalSkipgram-lc	KMeans	1M	2	15
normalSkipgram-lc	HDBScan	1M	2	20
leftSkipgram-nc	KMeans	1M	5	20
leftSkipgram-nc	HDBScan	1M	3	20
normalSkipgram-nc	KMeans	1M	3	10
normalSkipgram-nc	HDBScan	1M	3	20

Tabella 4.22: Risultati di Doc2Vec e TF-IDF in Oxford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
Doc2Vec	KMeans	0.45	0.36	0.46	0.54	-0.14	0.50
Doc2Vec	HDBScan	0.35	0.06	0.43	0.39	-0.30	0.40
TF-IDF	KMeans	0.60	0.53	0.61	0.73	-0.00	0.67
TF-IDF	HDBScan	0.45	0.20	0.49	0.48	-0.20	0.48

Tabella 4.23: Risultati della configurazione Combinato in Oxford

ListConstraint	Clustering	AMI	ARI	Com	Hom	Silh	V-M
left + Doc2Vec	KMeans	0.55	0.42	0.56	0.68	-0.06	0.61
left + Doc2Vec	HDBScan	0.39	0.11	0.49	0.42	-0.14	0.45
left + TF-IDF	KMeans	0.63	0.57	0.64	0.75	-0.04	0.69
left + TF-IDF	HDBScan	0.54	0.21	0.58	0.56	-0.09	0.57
normal + Doc2Vec	KMeans	0.53	0.38	0.54	0.65	-0.05	0.59
normal + Doc2Vec	HDBScan	0.43	0.22	0.46	0.58	-0.13	0.51
normal + TF-IDF	KMeans	0.63	0.54	0.64	0.77	-0.03	0.70
normal + TF-IDF	HDBScan	0.51	0.17	0.53	0.59	-0.10	0.56
NoConstraint							
left + Doc2Vec	KMeans	0.56	0.49	0.57	0.68	-0.08	0.62
left + Doc2Vec	HDBScan	0.46	0.17	0.52	0.49	-0.13	0.50
left + TF-IDF	KMeans	0.68	0.60	0.68	0.80	-0.04	0.74
left + TF-IDF	HDBScan	0.51	0.19	0.55	0.53	-0.11	0.54
normal + Doc2Vec	KMeans	0.57	0.49	0.58	0.70	-0.05	0.63
normal + Doc2Vec	HDBScan	0.42	0.16	0.46	0.55	-0.13	0.50
normal + TF-IDF	KMeans	0.63	0.55	0.64	0.76	-0.04	0.69
normal + TF-IDF	HDBScan	0.49	0.13	0.52	0.55	-0.11	0.53

4. Sperimentazione

Tabella 4.24: Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.27	0.31	0.29	0.41	-0.04	0.34
100K	3	10	0.26	0.31	0.28	0.37	-0.05	0.32
100K	5	10	0.35	0.38	0.36	0.48	-0.07	0.41
100K	7	10	0.38	0.41	0.39	0.50	-0.10	0.44
100K	2	15	0.32	0.46	0.34	0.47	-0.04	0.39
100K	3	15	0.31	0.42	0.33	0.43	-0.07	0.37
100K	5	15	0.38	0.46	0.39	0.51	-0.10	0.44
100K	7	15	0.39	0.48	0.40	0.51	-0.09	0.45
100K	2	20	0.34	0.50	0.35	0.49	-0.05	0.41
100K	3	20	0.35	0.49	0.36	0.48	-0.06	0.41
100K	5	20	0.40	0.51	0.41	0.52	-0.08	0.46
100K	7	20	0.41	0.51	0.42	0.52	-0.07	0.47
500K	2	10	0.31	0.36	0.32	0.47	-0.03	0.38
500K	3	10	0.33	0.44	0.34	0.47	-0.04	0.40
500K	5	10	0.40	0.51	0.41	0.54	-0.05	0.46
500K	7	10	0.36	0.35	0.37	0.52	-0.03	0.43
500K	2	15	0.35	0.54	0.36	0.49	-0.05	0.41
500K	3	15	0.37	0.38	0.38	0.53	-0.04	0.44
500K	5	15	0.36	0.38	0.37	0.52	-0.04	0.44
500K	7	15	0.36	0.35	0.37	0.51	-0.04	0.43
500K	2	20	0.35	0.56	0.36	0.49	-0.02	0.41
500K	3	20	0.38	0.39	0.39	0.54	-0.04	0.46
500K	5	20	0.38	0.40	0.39	0.54	-0.04	0.45
500K	7	20	0.35	0.32	0.36	0.51	-0.05	0.42
1M	2	10	0.34	0.49	0.35	0.48	-0.03	0.40
1M	3	10	0.40	0.50	0.41	0.53	-0.04	0.46
1M	5	10	0.35	0.26	0.36	0.53	-0.06	0.43
1M	7	10	0.36	0.31	0.37	0.52	-0.03	0.43
1M	2	15	0.36	0.57	0.37	0.50	-0.03	0.43
1M	3	15	0.38	0.39	0.39	0.55	-0.05	0.46
1M	5	15	0.37	0.34	0.39	0.54	-0.03	0.45
1M	7	15	0.36	0.31	0.37	0.53	-0.02	0.44
1M	2	20	0.36	0.56	0.38	0.50	-0.05	0.43
1M	3	20	0.38	0.36	0.39	0.56	-0.04	0.46
1M	5	20	0.37	0.30	0.38	0.55	-0.04	0.45
1M	7	20	0.34	0.26	0.35	0.50	-0.04	0.42

Tabella 4.25: Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.09	0.10	0.22	0.09	-0.08	0.13
100K	3	10	0.06	0.06	0.15	0.06	-0.07	0.09
100K	5	10	0.07	0.09	0.23	0.07	-0.11	0.11
100K	7	10	0.07	0.08	0.37	0.08	-0.14	0.13
100K	2	15	0.08	0.02	0.22	0.09	-0.14	0.13
100K	3	15	0.17	0.11	0.30	0.19	-0.21	0.23
100K	5	15	0.05	0.05	0.35	0.05	-0.03	0.09
100K	7	15	0.06	0.06	0.37	0.06	-0.03	0.10
100K	2	20	0.17	0.17	0.37	0.18	-0.11	0.24
100K	3	20	0.06	0.09	0.23	0.07	-0.15	0.10
100K	5	20	0.24	0.08	0.32	0.27	-0.29	0.29
100K	7	20	0.29	0.16	0.35	0.32	-0.32	0.33
500K	2	10	0.08	-0.02	0.24	0.10	-0.09	0.14
500K	3	10	0.08	0.10	0.20	0.08	-0.01	0.12
500K	5	10	0.14	-0.00	0.24	0.17	-0.14	0.20
500K	7	10	0.08	0.10	0.28	0.08	-0.08	0.13
500K	2	15	0.19	0.10	0.40	0.20	-0.12	0.27
500K	3	15	0.17	0.00	0.26	0.20	-0.13	0.22
500K	5	15	0.21	0.03	0.27	0.24	-0.29	0.25
500K	7	15	0.20	0.02	0.27	0.23	-0.22	0.25
500K	2	20	0.17	0.06	0.36	0.18	-0.13	0.24
500K	3	20	0.23	0.05	0.28	0.26	-0.16	0.27
500K	5	20	0.24	0.05	0.28	0.28	-0.19	0.28
500K	7	20	0.23	0.05	0.29	0.26	-0.19	0.28
1M	2	10	0.26	0.26	0.46	0.28	-0.09	0.34
1M	3	10	0.19	0.17	0.29	0.21	-0.16	0.24
1M	5	10	0.26	0.06	0.29	0.31	-0.21	0.30
1M	7	10	0.25	0.07	0.29	0.31	-0.18	0.30
1M	2	15	0.32	0.33	0.52	0.34	-0.16	0.41
1M	3	15	0.25	0.07	0.30	0.33	-0.18	0.31
1M	5	15	0.27	0.10	0.31	0.36	-0.19	0.33
1M	7	15	0.24	0.10	0.28	0.36	-0.15	0.32
1M	2	20	0.35	0.38	0.56	0.37	-0.17	0.44
1M	3	20	0.27	0.07	0.31	0.34	-0.21	0.32
1M	5	20	0.28	0.15	0.32	0.37	-0.18	0.34
1M	7	20	0.26	0.15	0.30	0.36	-0.17	0.33

4. Sperimentazione

Tabella 4.26: Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.31	0.21	0.33	0.45	-0.04	0.38
100K	3	10	0.28	0.16	0.30	0.45	-0.03	0.36
100K	5	10	0.29	0.19	0.31	0.44	-0.03	0.36
100K	7	10	0.29	0.18	0.31	0.44	-0.03	0.36
100K	2	15	0.31	0.20	0.32	0.49	-0.03	0.39
100K	3	15	0.32	0.25	0.33	0.47	-0.03	0.39
100K	5	15	0.29	0.20	0.31	0.44	-0.07	0.36
100K	7	15	0.28	0.18	0.29	0.42	-0.03	0.34
100K	2	20	0.30	0.20	0.31	0.47	-0.03	0.38
100K	3	20	0.31	0.24	0.32	0.46	-0.02	0.38
100K	5	20	0.30	0.19	0.32	0.48	-0.03	0.38
100K	7	20	0.30	0.22	0.31	0.43	-0.03	0.36
500K	2	10	0.36	0.26	0.37	0.55	-0.02	0.44
500K	3	10	0.37	0.26	0.38	0.58	-0.02	0.46
500K	5	10	0.37	0.27	0.38	0.58	-0.02	0.46
500K	7	10	0.38	0.33	0.39	0.59	-0.02	0.47
500K	2	15	0.37	0.32	0.38	0.58	-0.02	0.46
500K	3	15	0.35	0.24	0.36	0.56	-0.02	0.44
500K	5	15	0.35	0.30	0.36	0.51	-0.02	0.42
500K	7	15	0.37	0.43	0.38	0.54	-0.02	0.44
500K	2	20	0.40	0.37	0.41	0.60	-0.02	0.49
500K	3	20	0.39	0.35	0.40	0.59	-0.02	0.48
500K	5	20	0.36	0.24	0.37	0.55	-0.01	0.44
500K	7	20	0.34	0.26	0.35	0.53	-0.01	0.42
1M	2	10	0.46	0.42	0.47	0.69	-0.03	0.56
1M	3	10	0.43	0.39	0.44	0.64	-0.02	0.52
1M	5	10	0.49	0.65	0.49	0.67	-0.02	0.57
1M	7	10	0.47	0.60	0.48	0.67	-0.02	0.56
1M	2	15	0.46	0.41	0.47	0.69	-0.02	0.56
1M	3	15	0.43	0.35	0.44	0.65	-0.02	0.52
1M	5	15	0.44	0.46	0.45	0.65	-0.02	0.53
1M	7	15	0.43	0.49	0.44	0.62	-0.02	0.51
1M	2	20	0.45	0.40	0.46	0.67	-0.02	0.54
1M	3	20	0.41	0.35	0.42	0.63	-0.02	0.50
1M	5	20	0.46	0.46	0.47	0.65	-0.02	0.54
1M	7	20	0.46	0.49	0.47	0.66	-0.02	0.55

Tabella 4.27: Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.21	0.02	0.26	0.30	-0.12	0.28
100K	3	10	0.24	0.02	0.29	0.32	-0.12	0.30
100K	5	10	0.25	0.03	0.30	0.32	-0.13	0.31
100K	7	10	0.25	0.03	0.30	0.31	-0.14	0.30
100K	2	15	0.23	0.03	0.27	0.32	-0.13	0.29
100K	3	15	0.24	0.03	0.28	0.30	-0.15	0.29
100K	5	15	0.27	0.04	0.31	0.31	-0.11	0.31
100K	7	15	0.26	0.04	0.30	0.29	-0.10	0.30
100K	2	20	0.23	0.04	0.27	0.32	-0.15	0.29
100K	3	20	0.24	0.04	0.28	0.32	-0.15	0.30
100K	5	20	0.27	0.05	0.31	0.31	-0.15	0.31
100K	7	20	0.24	0.05	0.30	0.28	-0.15	0.29
500K	2	10	0.30	0.24	0.34	0.42	-0.10	0.38
500K	3	10	0.29	0.23	0.34	0.40	-0.10	0.37
500K	5	10	0.29	0.22	0.33	0.39	-0.09	0.35
500K	7	10	0.26	0.16	0.30	0.35	-0.09	0.32
500K	2	15	0.30	0.26	0.34	0.43	-0.10	0.38
500K	3	15	0.31	0.25	0.35	0.43	-0.09	0.39
500K	5	15	0.29	0.22	0.34	0.39	-0.09	0.36
500K	7	15	0.29	0.22	0.33	0.37	-0.11	0.35
500K	2	20	0.32	0.26	0.36	0.45	-0.09	0.40
500K	3	20	0.30	0.26	0.34	0.44	-0.09	0.38
500K	5	20	0.28	0.20	0.33	0.38	-0.09	0.35
500K	7	20	0.28	0.19	0.33	0.36	-0.08	0.34
1M	2	10	0.31	0.28	0.35	0.51	-0.12	0.41
1M	3	10	0.30	0.25	0.34	0.51	-0.11	0.41
1M	5	10	0.34	0.34	0.37	0.61	-0.11	0.46
1M	7	10	0.31	0.21	0.35	0.58	-0.10	0.43
1M	2	15	0.30	0.24	0.34	0.49	-0.12	0.40
1M	3	15	0.30	0.25	0.34	0.48	-0.11	0.40
1M	5	15	0.30	0.25	0.34	0.48	-0.11	0.40
1M	7	15	0.29	0.20	0.33	0.48	-0.11	0.39
1M	2	20	0.29	0.22	0.33	0.49	-0.11	0.40
1M	3	20	0.29	0.23	0.33	0.49	-0.12	0.39
1M	5	20	0.31	0.27	0.35	0.50	-0.09	0.41
1M	7	20	0.31	0.27	0.35	0.51	-0.11	0.41

4. Sperimentazione

Tabella 4.28: Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.28	0.34	0.30	0.41	-0.03	0.34
100K	3	10	0.31	0.33	0.32	0.41	-0.05	0.36
100K	5	10	0.34	0.36	0.36	0.48	-0.09	0.41
100K	7	10	0.37	0.40	0.38	0.50	-0.06	0.43
100K	2	15	0.31	0.43	0.32	0.44	-0.03	0.37
100K	3	15	0.32	0.41	0.33	0.43	-0.05	0.37
100K	5	15	0.37	0.44	0.38	0.48	-0.09	0.43
100K	7	15	0.38	0.46	0.40	0.50	-0.25	0.44
100K	2	20	0.33	0.47	0.34	0.47	-0.03	0.40
100K	3	20	0.34	0.47	0.36	0.47	-0.10	0.41
100K	5	20	0.38	0.48	0.39	0.50	-0.11	0.44
100K	7	20	0.39	0.47	0.40	0.51	-0.07	0.45
500K	2	10	0.30	0.39	0.31	0.46	-0.03	0.37
500K	3	10	0.34	0.44	0.36	0.49	-0.03	0.41
500K	5	10	0.33	0.32	0.34	0.49	-0.03	0.40
500K	7	10	0.35	0.33	0.36	0.50	-0.05	0.42
500K	2	15	0.33	0.43	0.34	0.48	-0.03	0.40
500K	3	15	0.39	0.52	0.40	0.52	-0.04	0.45
500K	5	15	0.35	0.40	0.36	0.49	-0.04	0.42
500K	7	15	0.32	0.29	0.33	0.47	-0.03	0.39
500K	2	20	0.35	0.52	0.36	0.49	-0.03	0.42
500K	3	20	0.37	0.39	0.38	0.53	-0.02	0.45
500K	5	20	0.35	0.37	0.36	0.50	-0.04	0.42
500K	7	20	0.28	0.21	0.29	0.39	-0.03	0.33
1M	2	10	0.33	0.48	0.34	0.47	-0.04	0.40
1M	3	10	0.36	0.34	0.37	0.52	-0.10	0.43
1M	5	10	0.33	0.25	0.34	0.50	-0.04	0.40
1M	7	10	0.33	0.28	0.34	0.50	-0.03	0.41
1M	2	15	0.33	0.36	0.34	0.49	-0.05	0.40
1M	3	15	0.38	0.36	0.39	0.54	-0.04	0.46
1M	5	15	0.35	0.30	0.36	0.51	-0.06	0.42
1M	7	15	0.33	0.27	0.34	0.49	-0.06	0.40
1M	2	20	0.35	0.55	0.36	0.50	-0.03	0.42
1M	3	20	0.38	0.36	0.39	0.56	-0.02	0.46
1M	5	20	0.36	0.31	0.37	0.52	-0.06	0.43
1M	7	20	0.34	0.28	0.35	0.50	-0.03	0.41

4. Sperimentazione

Tabella 4.29: Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.03	0.00	0.09	0.04	-0.06	0.05
100K	3	10	0.06	0.07	0.16	0.07	-0.04	0.09
100K	5	10	0.06	0.08	0.18	0.07	-0.00	0.10
100K	7	10	0.08	0.09	0.22	0.09	-0.09	0.13
100K	2	15	0.09	0.02	0.25	0.10	-0.09	0.14
100K	3	15	0.14	0.06	0.28	0.16	-0.19	0.21
100K	5	15	0.21	0.10	0.33	0.24	-0.27	0.28
100K	7	15	0.23	0.10	0.31	0.26	-0.22	0.28
100K	2	20	0.14	0.09	0.32	0.15	-0.13	0.20
100K	3	20	0.19	0.05	0.31	0.22	-0.19	0.26
100K	5	20	0.28	0.16	0.35	0.31	-0.28	0.33
100K	7	20	0.29	0.18	0.35	0.32	-0.31	0.33
500K	2	10	0.07	-0.02	0.24	0.08	-0.09	0.12
500K	3	10	0.10	0.11	0.24	0.10	-0.05	0.14
500K	5	10	0.15	0.00	0.25	0.17	-0.15	0.21
500K	7	10	0.15	0.00	0.27	0.18	-0.18	0.21
500K	2	15	0.15	0.21	0.39	0.15	-0.06	0.22
500K	3	15	0.17	0.01	0.25	0.20	-0.15	0.22
500K	5	15	0.21	0.03	0.27	0.25	-0.14	0.26
500K	7	15	0.20	0.03	0.28	0.23	-0.13	0.26
500K	2	20	0.17	0.05	0.37	0.18	-0.25	0.25
500K	3	20	0.23	0.05	0.29	0.26	-0.15	0.27
500K	5	20	0.24	0.05	0.29	0.27	-0.16	0.28
500K	7	20	0.22	0.04	0.28	0.25	-0.20	0.26
1M	2	10	0.27	0.22	0.45	0.28	-0.09	0.35
1M	3	10	0.25	0.04	0.29	0.28	-0.24	0.29
1M	5	10	0.26	0.06	0.30	0.33	-0.21	0.31
1M	7	10	0.25	0.07	0.28	0.33	-0.21	0.31
1M	2	15	0.33	0.33	0.50	0.34	-0.11	0.40
1M	3	15	0.26	0.06	0.30	0.33	-0.23	0.31
1M	5	15	0.26	0.10	0.30	0.34	-0.22	0.32
1M	7	15	0.26	0.11	0.30	0.35	-0.19	0.32
1M	2	20	0.29	0.27	0.47	0.31	-0.12	0.37
1M	3	20	0.27	0.07	0.31	0.34	-0.21	0.33
1M	5	20	0.27	0.11	0.30	0.37	-0.23	0.33
1M	7	20	0.28	0.14	0.31	0.39	-0.16	0.34

4. Sperimentazione

Tabella 4.30: Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.30	0.18	0.31	0.46	-0.03	0.37
100K	3	10	0.28	0.18	0.30	0.44	-0.03	0.36
100K	5	10	0.28	0.19	0.30	0.44	-0.03	0.36
100K	7	10	0.30	0.21	0.31	0.44	-0.04	0.37
100K	2	15	0.31	0.22	0.32	0.47	-0.03	0.38
100K	3	15	0.30	0.20	0.31	0.45	-0.03	0.37
100K	5	15	0.30	0.21	0.31	0.46	-0.03	0.37
100K	7	15	0.31	0.22	0.32	0.46	-0.05	0.38
100K	2	20	0.32	0.21	0.33	0.50	-0.02	0.40
100K	3	20	0.30	0.21	0.31	0.46	-0.03	0.37
100K	5	20	0.31	0.24	0.32	0.45	-0.03	0.37
100K	7	20	0.28	0.19	0.29	0.44	-0.02	0.35
500K	2	10	0.36	0.27	0.37	0.57	-0.02	0.45
500K	3	10	0.36	0.30	0.37	0.54	-0.04	0.44
500K	5	10	0.36	0.23	0.37	0.55	-0.04	0.44
500K	7	10	0.34	0.28	0.35	0.50	-0.02	0.41
500K	2	15	0.37	0.29	0.38	0.58	-0.02	0.46
500K	3	15	0.35	0.25	0.36	0.54	-0.02	0.43
500K	5	15	0.36	0.30	0.37	0.55	-0.02	0.44
500K	7	15	0.36	0.43	0.37	0.52	-0.02	0.44
500K	2	20	0.37	0.27	0.38	0.58	-0.02	0.46
500K	3	20	0.36	0.29	0.37	0.55	-0.02	0.44
500K	5	20	0.33	0.30	0.34	0.48	-0.02	0.40
500K	7	20	0.34	0.30	0.35	0.50	-0.02	0.41
1M	2	10	0.41	0.36	0.42	0.64	-0.02	0.51
1M	3	10	0.43	0.39	0.44	0.66	-0.02	0.53
1M	5	10	0.41	0.37	0.42	0.63	-0.02	0.51
1M	7	10	0.46	0.53	0.47	0.66	-0.02	0.55
1M	2	15	0.44	0.44	0.45	0.64	-0.03	0.53
1M	3	15	0.41	0.35	0.42	0.63	-0.03	0.50
1M	5	15	0.41	0.40	0.42	0.61	-0.03	0.50
1M	7	15	0.44	0.43	0.45	0.64	-0.02	0.53
1M	2	20	0.43	0.50	0.44	0.62	-0.02	0.51
1M	3	20	0.41	0.42	0.42	0.62	-0.02	0.50
1M	5	20	0.41	0.44	0.42	0.59	-0.02	0.49
1M	7	20	0.41	0.37	0.42	0.60	-0.02	0.49

4. Sperimentazione

Tabella 4.31: Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Princeton

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.22	0.02	0.27	0.32	-0.16	0.29
100K	3	10	0.21	0.02	0.26	0.29	-0.15	0.28
100K	5	10	0.25	0.04	0.30	0.32	-0.14	0.31
100K	7	10	0.25	0.04	0.30	0.31	-0.13	0.30
100K	2	15	0.24	0.03	0.28	0.30	-0.12	0.29
100K	3	15	0.25	0.04	0.29	0.31	-0.13	0.30
100K	5	15	0.27	0.05	0.31	0.32	-0.12	0.31
100K	7	15	0.26	0.04	0.31	0.30	-0.11	0.30
100K	2	20	0.23	0.03	0.27	0.31	-0.14	0.29
100K	3	20	0.23	0.03	0.27	0.30	-0.13	0.28
100K	5	20	0.27	0.05	0.31	0.31	-0.14	0.31
100K	7	20	0.25	0.05	0.31	0.29	-0.13	0.30
500K	2	10	0.28	0.18	0.32	0.35	-0.10	0.33
500K	3	10	0.29	0.18	0.33	0.35	-0.13	0.34
500K	5	10	0.27	0.16	0.31	0.31	-0.09	0.31
500K	7	10	0.26	0.15	0.32	0.30	-0.14	0.31
500K	2	15	0.29	0.21	0.33	0.38	-0.09	0.35
500K	3	15	0.31	0.22	0.35	0.41	-0.09	0.38
500K	5	15	0.30	0.19	0.34	0.36	-0.08	0.35
500K	7	15	0.30	0.18	0.34	0.34	-0.07	0.34
500K	2	20	0.31	0.23	0.35	0.42	-0.09	0.38
500K	3	20	0.32	0.24	0.35	0.43	-0.09	0.39
500K	5	20	0.30	0.19	0.33	0.35	-0.07	0.34
500K	7	20	0.30	0.19	0.34	0.35	-0.07	0.34
1M	2	10	0.29	0.23	0.33	0.50	-0.13	0.40
1M	3	10	0.31	0.28	0.35	0.51	-0.12	0.41
1M	5	10	0.30	0.22	0.34	0.53	-0.11	0.41
1M	7	10	0.32	0.30	0.36	0.52	-0.11	0.42
1M	2	15	0.30	0.24	0.34	0.49	-0.12	0.40
1M	3	15	0.31	0.26	0.35	0.49	-0.12	0.41
1M	5	15	0.30	0.28	0.34	0.48	-0.10	0.40
1M	7	15	0.30	0.28	0.34	0.48	-0.10	0.40
1M	2	20	0.31	0.23	0.34	0.50	-0.12	0.41
1M	3	20	0.31	0.27	0.35	0.49	-0.11	0.41
1M	5	20	0.31	0.27	0.35	0.48	-0.10	0.40
1M	7	20	0.31	0.27	0.34	0.48	-0.09	0.40

Tabella 4.32: Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Princeton

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
leftSkipgram-lc	KMeans	0.41	0.51	0.42	0.52	-0.07	0.47
leftSkipgram-lc	HDBScan	0.35	0.38	0.56	0.37	-0.17	0.44
normalSkipgram-lc	KMeans	0.49	0.65	0.49	0.67	-0.02	0.57
normalSkipgram-lc	HDBScan	0.34	0.34	0.37	0.61	-0.11	0.46
LINE-1-lc	KMeans	0.15	0.07	0.17	0.24	-0.10	0.20
LINE-1-lc	HDBScan	0.09	0.01	0.20	0.10	-0.30	0.13
LINE-2-lc	KMeans	0.28	0.29	0.30	0.36	-0.17	0.33
LINE-2-lc	HDBScan	0.07	0.11	0.28	0.08	0.21	0.12
leftSkipgram-nc	KMeans	0.38	0.36	0.39	0.56	-0.02	0.46
leftSkipgram-nc	HDBScan	0.33	0.33	0.50	0.34	-0.11	0.40
normalSkipgram-nc	KMeans	0.46	0.53	0.47	0.66	-0.02	0.55
normalSkipgram-nc	HDBScan	0.32	0.30	0.36	0.52	-0.11	0.42
LINE-1-nc	KMeans	0.23	0.10	0.25	0.36	-0.02	0.29
LINE-1-nc	HDBScan	0.33	0.43	0.62	0.33	-0.02	0.44
LINE-2-nc	KMeans	0.23	0.13	0.25	0.38	-0.07	0.30
LINE-2-nc	HDBScan	0.34	0.48	0.55	0.34	0.07	0.42

	Clustering	DB size	Window	RW len.
leftSkipgram-lc	KMeans	100K	7	20
leftSkipgram-lc	HDBScan	1M	2	20
normalSkipgram-lc	KMeans	1M	5	10
normalSkipgram-lc	HDBScan	1M	5	10
leftSkipgram-nc	KMeans	1M	3	20
leftSkipgram-nc	HDBScan	1M	2	15
normalSkipgram-nc	KMeans	1M	7	10
normalSkipgram-nc	HDBScan	1M	7	10

Tabella 4.33: Risultati di Doc2Vec e TF-IDF in Princeton

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
Doc2Vec	KMeans	0.47	0.43	0.47	0.57	-0.04	0.52
Doc2Vec	HDBScan	0.41	0.26	0.44	0.50	-0.38	0.47
TF-IDF	KMeans	0.57	0.68	0.58	0.71	0.05	0.64
TF-IDF	HDBScan	0.30	0.16	0.33	0.38	-0.32	0.36

Tabella 4.34: Risultati della configurazione Combinato in Priceton

ListConstraint	Clustering	AMI	ARI	Com	Hom	Silh	V-M
left + Doc2Vec	KMeans	0.48	0.55	0.49	0.62	-0.13	0.55
left + Doc2Vec	HDBScan	0.36	0.39	0.57	0.37	-0.26	0.45
left + TF-IDF	KMeans	0.49	0.55	0.50	0.64	-0.04	0.56
left + TF-IDF	HDBScan	0.38	0.38	0.50	0.39	-0.23	0.44
normal + Doc2Vec	KMeans	0.42	0.38	0.43	0.63	-0.03	0.51
normal + Doc2Vec	HDBScan	0.36	0.22	0.39	0.57	-0.12	0.46
normal + TF-IDF	KMeans	0.48	0.41	0.49	0.71	-0.03	0.58
normal + TF-IDF	HDBScan	0.39	0.17	0.41	0.50	-0.16	0.45
NoConstraint							
left + Doc2Vec	KMeans	0.43	0.37	0.44	0.64	-0.02	0.52
left + Doc2Vec	HDBScan	0.35	0.36	0.51	0.36	-0.12	0.42
left + TF-IDF	KMeans	0.52	0.64	0.53	0.73	-0.02	0.61
left + TF-IDF	HDBScan	0.38	0.37	0.47	0.40	-0.17	0.43
normal + Doc2Vec	KMeans	0.50	0.47	0.50	0.71	-0.02	0.59
normal + Doc2Vec	HDBScan	0.35	0.17	0.38	0.53	-0.12	0.44
normal + TF-IDF	KMeans	0.53	0.65	0.54	0.73	-0.03	0.62
normal + TF-IDF	HDBScan	0.38	0.18	0.41	0.47	-0.16	0.44

4. Sperimentazione

Tabella 4.35: Risultati di KMeans con leftSkipgram con liste di costrizione nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.24	0.14	0.36	0.34	-0.11	0.35
100K	3	10	0.25	0.10	0.42	0.33	-0.07	0.37
100K	5	10	0.33	0.17	0.53	0.40	-0.06	0.45
100K	7	10	0.27	0.14	0.40	0.35	-0.07	0.37
100K	2	15	0.24	0.09	0.35	0.34	-0.08	0.35
100K	3	15	0.28	0.14	0.43	0.37	-0.20	0.40
100K	5	15	0.22	0.08	0.38	0.31	-0.06	0.34
100K	7	15	0.21	0.12	0.30	0.32	-0.08	0.31
100K	2	20	0.28	0.15	0.43	0.37	-0.11	0.39
100K	3	20	0.27	0.13	0.37	0.37	-0.03	0.37
100K	5	20	0.28	0.14	0.40	0.36	-0.05	0.38
100K	7	20	0.21	0.07	0.51	0.29	-0.06	0.37
500K	2	10	0.22	0.15	0.32	0.32	-0.07	0.32
500K	3	10	0.32	0.23	0.41	0.40	-0.05	0.41
500K	5	10	0.29	0.11	0.40	0.38	-0.09	0.39
500K	7	10	0.39	0.23	0.47	0.47	-0.05	0.47
500K	2	15	0.32	0.17	0.41	0.42	-0.04	0.42
500K	3	15	0.32	0.17	0.51	0.39	-0.08	0.44
500K	5	15	0.38	0.18	0.46	0.45	-0.05	0.46
500K	7	15	0.29	0.11	0.55	0.37	-0.07	0.44
500K	2	20	0.29	0.20	0.38	0.40	-0.05	0.39
500K	3	20	0.30	0.21	0.40	0.39	-0.07	0.39
500K	5	20	0.29	0.13	0.40	0.39	-0.06	0.39
500K	7	20	0.27	0.12	0.37	0.37	-0.05	0.37
1M	2	10	0.23	0.16	0.33	0.34	-0.11	0.33
1M	3	10	0.35	0.14	0.57	0.42	-0.07	0.48
1M	5	10	0.29	0.11	0.46	0.37	-0.08	0.41
1M	7	10	0.41	0.22	0.49	0.48	-0.07	0.49
1M	2	15	0.25	0.16	0.35	0.38	-0.04	0.36
1M	3	15	0.27	0.11	0.52	0.34	-0.05	0.41
1M	5	15	0.34	0.15	0.44	0.41	-0.04	0.43
1M	7	15	0.34	0.17	0.44	0.41	-0.04	0.43
1M	2	20	0.25	0.11	0.35	0.36	-0.04	0.35
1M	3	20	0.31	0.11	0.53	0.40	-0.05	0.45
1M	5	20	0.35	0.16	0.46	0.43	-0.07	0.45
1M	7	20	0.36	0.15	0.53	0.44	-0.06	0.48

Tabella 4.36: Risultati di HDBScan con leftSkipgram con liste di costrizione nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.10	0.03	0.55	0.13	-0.02	0.21
100K	3	10	0.19	0.09	0.59	0.24	-0.09	0.34
100K	5	10	0.21	0.10	0.61	0.25	-0.08	0.36
100K	7	10	0.20	0.10	0.60	0.25	-0.08	0.35
100K	2	15	0.13	0.05	0.56	0.16	0.04	0.25
100K	3	15	0.22	0.11	0.64	0.27	-0.03	0.38
100K	5	15	0.20	0.10	0.61	0.25	-0.02	0.35
100K	7	15	0.20	0.10	0.61	0.25	-0.02	0.35
100K	2	20	0.10	0.03	0.43	0.14	-0.01	0.21
100K	3	20	0.21	0.11	0.63	0.26	-0.03	0.36
100K	5	20	0.21	0.11	0.63	0.26	-0.02	0.37
100K	7	20	0.22	0.07	0.56	0.27	-0.02	0.37
500K	2	10	0.24	0.14	0.47	0.29	-0.02	0.36
500K	3	10	0.18	0.08	0.65	0.22	-0.01	0.33
500K	5	10	0.21	0.11	0.68	0.24	-0.01	0.36
500K	7	10	0.23	0.08	0.60	0.27	-0.01	0.38
500K	2	15	0.15	0.05	0.42	0.20	-0.01	0.27
500K	3	15	0.20	0.10	0.61	0.25	-0.05	0.36
500K	5	15	0.21	0.07	0.57	0.26	-0.01	0.36
500K	7	15	0.19	0.06	0.54	0.24	-0.01	0.34
500K	2	20	0.23	0.13	0.45	0.28	-0.02	0.34
500K	3	20	0.22	0.09	0.57	0.27	-0.01	0.37
500K	5	20	0.20	0.06	0.53	0.25	-0.02	0.34
500K	7	20	0.20	0.06	0.54	0.25	-0.02	0.34
1M	2	10	0.15	0.08	0.35	0.19	-0.01	0.25
1M	3	10	0.19	0.06	0.53	0.25	-0.04	0.34
1M	5	10	0.19	0.08	0.44	0.25	-0.09	0.32
1M	7	10	0.21	0.08	0.45	0.27	-0.06	0.34
1M	2	15	0.14	0.04	0.43	0.19	-0.03	0.26
1M	3	15	0.20	0.05	0.55	0.24	-0.03	0.34
1M	5	15	0.24	0.08	0.51	0.29	-0.03	0.37
1M	7	15	0.22	0.04	0.48	0.29	-0.04	0.36
1M	2	20	0.16	0.01	0.42	0.22	-0.04	0.29
1M	3	20	0.22	0.07	0.47	0.28	-0.03	0.35
1M	5	20	0.28	0.12	0.57	0.33	-0.05	0.42
1M	7	20	0.25	0.09	0.53	0.31	-0.04	0.39

Tabella 4.37: Risultati di KMeans con normalSkipgram con liste di costrizione nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.35	0.16	0.45	0.43	-0.04	0.44
100K	3	10	0.38	0.21	0.46	0.46	-0.06	0.46
100K	5	10	0.24	0.08	0.37	0.34	-0.06	0.35
100K	7	10	0.28	0.09	0.47	0.37	-0.07	0.41
100K	2	15	0.31	0.06	0.48	0.39	-0.05	0.43
100K	3	15	0.36	0.14	0.44	0.44	-0.05	0.44
100K	5	15	0.28	0.15	0.37	0.38	-0.05	0.37
100K	7	15	0.24	0.09	0.46	0.33	-0.07	0.39
100K	2	20	0.46	0.28	0.53	0.55	-0.05	0.54
100K	3	20	0.35	0.17	0.49	0.43	-0.08	0.46
100K	5	20	0.25	0.12	0.36	0.34	-0.03	0.35
100K	7	20	0.22	0.09	0.35	0.31	-0.04	0.33
500K	2	10	0.47	0.27	0.54	0.57	-0.04	0.55
500K	3	10	0.35	0.19	0.48	0.42	-0.03	0.45
500K	5	10	0.30	0.12	0.43	0.38	-0.05	0.40
500K	7	10	0.35	0.15	0.52	0.43	-0.05	0.47
500K	2	15	0.42	0.20	0.49	0.49	-0.04	0.49
500K	3	15	0.38	0.24	0.52	0.45	-0.05	0.48
500K	5	15	0.26	0.09	0.42	0.35	-0.06	0.38
500K	7	15	0.26	0.12	0.36	0.35	-0.04	0.35
500K	2	20	0.40	0.26	0.49	0.47	-0.06	0.48
500K	3	20	0.33	0.15	0.42	0.41	-0.04	0.42
500K	5	20	0.27	0.10	0.52	0.35	-0.07	0.42
500K	7	20	0.24	0.10	0.41	0.32	-0.05	0.36
1M	2	10	0.43	0.17	0.51	0.51	-0.06	0.51
1M	3	10	0.45	0.22	0.52	0.54	-0.05	0.53
1M	5	10	0.40	0.25	0.51	0.47	-0.03	0.49
1M	7	10	0.33	0.14	0.50	0.41	-0.03	0.45
1M	2	15	0.42	0.21	0.52	0.49	-0.04	0.51
1M	3	15	0.43	0.22	0.53	0.50	-0.04	0.51
1M	5	15	0.36	0.16	0.57	0.43	-0.04	0.49
1M	7	15	0.28	0.14	0.39	0.37	-0.03	0.38
1M	2	20	0.47	0.26	0.57	0.53	-0.03	0.55
1M	3	20	0.38	0.18	0.47	0.46	-0.04	0.46
1M	5	20	0.25	0.08	0.42	0.34	-0.05	0.37
1M	7	20	0.27	0.09	0.52	0.35	-0.05	0.42

Tabella 4.38: Risultati di HDBScan con normalSkipgram con liste di costrizione nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.06	0.01	0.29	0.09	-0.01	0.13
100K	3	10	0.06	0.02	0.23	0.08	-0.03	0.12
100K	5	10	0.12	0.06	0.31	0.16	-0.02	0.21
100K	7	10	0.05	0.02	0.22	0.08	-0.02	0.12
100K	2	15	0.26	0.15	0.47	0.32	-0.02	0.38
100K	3	15	0.21	0.17	0.44	0.25	-0.02	0.32
100K	5	15	0.19	0.12	0.39	0.23	-0.04	0.29
100K	7	15	0.20	0.13	0.41	0.25	-0.04	0.31
100K	2	20	0.04	-0.00	0.27	0.07	-0.02	0.11
100K	3	20	0.04	-0.01	0.22	0.06	-0.01	0.10
100K	5	20	0.14	0.11	0.36	0.18	-0.04	0.24
100K	7	20	0.12	0.11	0.32	0.16	-0.05	0.21
500K	2	10	0.23	0.10	0.41	0.29	-0.02	0.34
500K	3	10	0.29	0.13	0.46	0.35	-0.02	0.40
500K	5	10	0.28	0.15	0.47	0.34	-0.07	0.39
500K	7	10	0.27	0.13	0.46	0.33	-0.07	0.38
500K	2	15	0.28	0.13	0.46	0.34	-0.02	0.39
500K	3	15	0.25	0.13	0.44	0.29	-0.02	0.35
500K	5	15	0.24	0.12	0.44	0.30	-0.04	0.35
500K	7	15	0.22	0.10	0.41	0.28	-0.04	0.33
500K	2	20	0.24	0.14	0.44	0.29	-0.02	0.35
500K	3	20	0.25	0.12	0.46	0.30	-0.04	0.37
500K	5	20	0.28	0.13	0.50	0.34	-0.04	0.40
500K	7	20	0.27	0.13	0.49	0.32	-0.03	0.39
1M	2	10	0.34	0.18	0.49	0.41	-0.03	0.45
1M	3	10	0.22	0.08	0.40	0.28	-0.02	0.33
1M	5	10	0.26	0.14	0.44	0.33	-0.04	0.37
1M	7	10	0.23	0.11	0.41	0.29	-0.02	0.34
1M	2	15	0.35	0.16	0.48	0.42	-0.04	0.45
1M	3	15	0.25	0.11	0.43	0.30	-0.02	0.36
1M	5	15	0.28	0.14	0.48	0.35	-0.04	0.40
1M	7	15	0.25	0.11	0.43	0.31	-0.03	0.36
1M	2	20	0.37	0.18	0.52	0.44	-0.08	0.48
1M	3	20	0.07	0.01	0.48	0.09	-0.04	0.15
1M	5	20	0.07	0.02	0.52	0.09	-0.02	0.15
1M	7	20	0.07	0.02	0.52	0.09	-0.04	0.15

4. Sperimentazione

Tabella 4.39: Risultati di KMeans con leftSkipgram senza costrizioni nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.11	0.04	0.27	0.22	-0.12	0.24
100K	3	10	0.10	0.03	0.27	0.21	-0.07	0.24
100K	5	10	0.12	0.07	0.25	0.27	-0.08	0.26
100K	7	10	0.17	0.07	0.42	0.25	-0.11	0.32
100K	2	15	0.11	0.03	0.25	0.23	-0.05	0.24
100K	3	15	0.15	0.06	0.46	0.24	-0.20	0.32
100K	5	15	0.19	0.07	0.36	0.28	-0.07	0.31
100K	7	15	0.20	0.10	0.38	0.29	-0.12	0.33
100K	2	20	0.14	0.07	0.28	0.25	-0.28	0.26
100K	3	20	0.17	0.07	0.58	0.25	-0.22	0.35
100K	5	20	0.24	0.10	0.37	0.33	-0.06	0.35
100K	7	20	0.19	0.10	0.34	0.29	-0.09	0.32
500K	2	10	0.08	0.03	0.20	0.21	-0.03	0.20
500K	3	10	0.20	0.06	0.51	0.28	-0.17	0.36
500K	5	10	0.18	0.05	0.49	0.27	-0.08	0.34
500K	7	10	0.19	0.03	0.41	0.28	-0.06	0.33
500K	2	15	0.16	0.05	0.30	0.27	-0.10	0.28
500K	3	15	0.26	0.08	0.47	0.35	-0.10	0.40
500K	5	15	0.19	0.05	0.49	0.28	-0.03	0.36
500K	7	15	0.13	0.02	0.43	0.22	-0.04	0.29
500K	2	20	0.19	0.05	0.31	0.30	-0.05	0.31
500K	3	20	0.29	0.10	0.63	0.36	-0.04	0.46
500K	5	20	0.32	0.12	0.53	0.40	-0.07	0.46
500K	7	20	0.33	0.12	0.59	0.40	-0.04	0.48
1M	2	10	0.10	0.05	0.22	0.23	-0.05	0.22
1M	3	10	0.22	0.12	0.33	0.36	-0.04	0.34
1M	5	10	0.27	0.08	0.45	0.34	-0.05	0.39
1M	7	10	0.26	0.09	0.52	0.34	-0.04	0.41
1M	2	15	0.07	0.04	0.20	0.21	-0.03	0.20
1M	3	15	0.25	0.08	0.55	0.33	-0.03	0.41
1M	5	15	0.24	0.06	0.56	0.32	-0.05	0.41
1M	7	15	0.18	0.05	0.37	0.26	-0.03	0.31
1M	2	20	0.16	0.07	0.27	0.27	-0.03	0.27
1M	3	20	0.25	0.07	0.57	0.33	-0.03	0.42
1M	5	20	0.27	0.10	0.44	0.36	-0.03	0.40
1M	7	20	0.27	0.08	0.44	0.35	-0.05	0.39

4. Sperimentazione

Tabella 4.40: Risultati di HDBScan con leftSkipgram senza costrizioni nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.06	0.01	0.40	0.09	0.01	0.15
100K	3	10	0.08	0.02	0.43	0.11	-0.01	0.17
100K	5	10	0.08	0.03	0.45	0.11	-0.02	0.18
100K	7	10	0.09	0.04	0.46	0.12	-0.03	0.19
100K	2	15	0.08	0.05	0.44	0.11	0.01	0.18
100K	3	15	0.09	0.05	0.29	0.12	-0.05	0.17
100K	5	15	0.07	0.02	0.37	0.10	-0.05	0.16
100K	7	15	0.07	0.03	0.57	0.10	-0.04	0.16
100K	2	20	0.09	0.03	0.48	0.12	-0.03	0.19
100K	3	20	0.13	0.04	0.47	0.16	-0.02	0.24
100K	5	20	0.06	0.02	0.44	0.09	-0.06	0.16
100K	7	20	0.05	0.02	0.55	0.07	-0.15	0.13
500K	2	10	0.11	0.07	0.33	0.14	-0.01	0.20
500K	3	10	0.16	0.05	0.56	0.20	-0.03	0.29
500K	5	10	0.19	0.06	0.60	0.24	-0.04	0.34
500K	7	10	0.19	0.04	0.55	0.25	-0.06	0.35
500K	2	15	0.12	0.02	0.47	0.16	-0.00	0.24
500K	3	15	0.21	0.06	0.58	0.27	-0.04	0.37
500K	5	15	0.11	0.02	0.51	0.15	-0.01	0.24
500K	7	15	0.12	0.02	0.53	0.16	-0.01	0.25
500K	2	20	0.13	0.01	0.49	0.17	-0.02	0.26
500K	3	20	0.19	0.06	0.58	0.23	-0.03	0.33
500K	5	20	0.29	0.10	0.64	0.36	-0.05	0.46
500K	7	20	0.27	0.10	0.61	0.32	-0.03	0.42
1M	2	10	0.06	-0.00	0.45	0.09	0.00	0.15
1M	3	10	0.15	0.04	0.55	0.20	-0.03	0.30
1M	5	10	0.28	0.12	0.68	0.32	-0.04	0.43
1M	7	10	0.23	0.10	0.57	0.28	-0.02	0.38
1M	2	15	0.07	-0.00	0.40	0.10	-0.01	0.17
1M	3	15	0.21	0.06	0.59	0.27	-0.03	0.37
1M	5	15	0.19	0.05	0.55	0.24	-0.02	0.33
1M	7	15	0.22	0.08	0.47	0.27	-0.02	0.34
1M	2	20	0.12	0.01	0.44	0.16	-0.02	0.24
1M	3	20	0.28	0.11	0.68	0.33	-0.02	0.44
1M	5	20	0.25	0.07	0.58	0.32	-0.03	0.41
1M	7	20	0.31	0.12	0.68	0.35	-0.02	0.46

Tabella 4.41: Risultati di KMeans con normalSkipgram senza costrizioni nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.18	0.08	0.30	0.30	-0.06	0.30
100K	3	10	0.12	0.07	0.24	0.26	-0.05	0.25
100K	5	10	0.15	0.06	0.27	0.28	-0.05	0.27
100K	7	10	0.15	0.10	0.27	0.29	-0.03	0.28
100K	2	15	0.20	0.09	0.31	0.33	-0.05	0.32
100K	3	15	0.27	0.12	0.37	0.39	-0.06	0.38
100K	5	15	0.31	0.16	0.41	0.43	-0.04	0.42
100K	7	15	0.20	0.07	0.31	0.33	-0.03	0.32
100K	2	20	0.30	0.13	0.41	0.39	-0.09	0.40
100K	3	20	0.31	0.13	0.41	0.42	-0.07	0.42
100K	5	20	0.30	0.09	0.46	0.39	-0.06	0.42
100K	7	20	0.24	0.10	0.35	0.37	-0.05	0.36
500K	2	10	0.28	0.13	0.56	0.36	-0.08	0.44
500K	3	10	0.33	0.12	0.43	0.42	-0.07	0.42
500K	5	10	0.35	0.15	0.45	0.44	-0.05	0.45
500K	7	10	0.30	0.13	0.40	0.42	-0.04	0.41
500K	2	15	0.33	0.12	0.50	0.41	-0.06	0.45
500K	3	15	0.33	0.12	0.43	0.42	-0.04	0.42
500K	5	15	0.23	0.08	0.37	0.33	-0.03	0.35
500K	7	15	0.16	0.08	0.28	0.30	-0.03	0.29
500K	2	20	0.32	0.15	0.41	0.44	-0.05	0.42
500K	3	20	0.34	0.14	0.44	0.43	-0.06	0.43
500K	5	20	0.34	0.14	0.46	0.43	-0.04	0.44
500K	7	20	0.33	0.13	0.44	0.42	-0.03	0.43
1M	2	10	0.32	0.12	0.42	0.41	-0.06	0.41
1M	3	10	0.33	0.15	0.47	0.41	-0.05	0.44
1M	5	10	0.32	0.13	0.42	0.41	-0.04	0.41
1M	7	10	0.32	0.12	0.64	0.39	-0.05	0.49
1M	2	15	0.34	0.11	0.47	0.42	-0.05	0.45
1M	3	15	0.34	0.13	0.49	0.42	-0.03	0.45
1M	5	15	0.33	0.13	0.45	0.42	-0.03	0.44
1M	7	15	0.32	0.13	0.67	0.39	-0.02	0.49
1M	2	20	0.33	0.12	0.65	0.40	-0.04	0.49
1M	3	20	0.33	0.13	0.47	0.41	-0.04	0.44
1M	5	20	0.30	0.11	0.42	0.40	-0.04	0.41
1M	7	20	0.33	0.11	0.61	0.41	-0.05	0.49

Tabella 4.42: Risultati di HDBScan con normalSkipgram senza costrizioni nella configurazione Random Walk in Stanford

DB size	Window	RW len.	AMI	ARI	Com	Hom	Silh	V-M
100K	2	10	0.22	0.11	0.59	0.25	-0.01	0.35
100K	3	10	0.18	0.10	0.45	0.21	-0.01	0.28
100K	5	10	0.25	0.13	0.64	0.27	-0.01	0.38
100K	7	10	0.23	0.13	0.58	0.26	-0.01	0.35
100K	2	15	0.11	0.07	0.43	0.14	-0.01	0.21
100K	3	15	0.17	0.07	0.62	0.21	-0.03	0.31
100K	5	15	0.09	0.05	0.70	0.12	-0.02	0.20
100K	7	15	0.09	0.04	0.66	0.12	-0.01	0.20
100K	2	20	0.31	0.11	0.52	0.38	-0.09	0.44
100K	3	20	0.23	0.10	0.45	0.29	-0.06	0.35
100K	5	20	0.33	0.14	0.71	0.38	-0.05	0.50
100K	7	20	0.31	0.13	0.71	0.37	-0.05	0.48
500K	2	10	0.19	0.13	0.45	0.22	-0.00	0.29
500K	3	10	0.32	0.13	0.53	0.39	-0.05	0.45
500K	5	10	0.32	0.13	0.70	0.37	-0.05	0.48
500K	7	10	0.34	0.14	0.71	0.39	-0.04	0.50
500K	2	15	0.35	0.12	0.60	0.42	-0.04	0.49
500K	3	15	0.31	0.10	0.64	0.38	-0.03	0.47
500K	5	15	0.14	0.06	0.52	0.17	-0.01	0.26
500K	7	15	0.15	0.02	0.49	0.20	-0.02	0.29
500K	2	20	0.36	0.12	0.62	0.43	-0.04	0.51
500K	3	20	0.36	0.12	0.62	0.43	-0.03	0.51
500K	5	20	0.33	0.13	0.56	0.40	-0.03	0.47
500K	7	20	0.33	0.12	0.64	0.39	-0.03	0.48
1M	2	10	0.33	0.09	0.53	0.40	-0.05	0.46
1M	3	10	0.31	0.12	0.47	0.37	-0.03	0.42
1M	5	10	0.32	0.13	0.56	0.40	-0.04	0.46
1M	7	10	0.21	0.05	0.57	0.27	-0.03	0.36
1M	2	15	0.30	0.10	0.63	0.37	-0.04	0.47
1M	3	15	0.30	0.11	0.63	0.36	-0.03	0.46
1M	5	15	0.30	0.10	0.63	0.37	-0.03	0.47
1M	7	15	0.30	0.10	0.63	0.37	-0.02	0.46
1M	2	20	0.27	0.07	0.51	0.35	-0.04	0.41
1M	3	20	0.28	0.08	0.54	0.35	-0.04	0.43
1M	5	20	0.28	0.09	0.60	0.35	-0.03	0.44
1M	7	20	0.28	0.08	0.61	0.35	-0.03	0.44

Tabella 4.43: Risultati migliori tra leftSkipgram, normalSkipgram e LINE in Stanford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
leftSkipgram-lc	KMeans	0.41	0.22	0.49	0.48	-0.07	0.49
leftSkipgram-lc	HDBScan	0.28	0.12	0.57	0.33	-0.05	0.42
normalSkipgram-lc	KMeans	0.47	0.26	0.57	0.53	-0.03	0.55
normalSkipgram-lc	HDBScan	0.37	0.18	0.52	0.44	-0.08	0.48
LINE-1-lc	KMeans	0.17	0.09	0.28	0.30	-0.08	0.29
LINE-1-lc	HDBScan	0.15	0.06	0.57	0.19	0.00	0.28
LINE-2-lc	KMeans	0.19	0.12	0.29	0.33	-0.09	0.31
LINE-2-lc	HDBScan	0.18	0.06	0.52	0.21	0.01	0.30
leftSkipgram-nc	KMeans	0.33	0.12	0.59	0.40	-0.04	0.48
leftSkipgram-nc	HDBScan	0.31	0.12	0.68	0.35	-0.02	0.46
normalSkipgram-nc	KMeans	0.32	0.13	0.67	0.39	-0.02	0.49
normalSkipgram-nc	HDBScan	0.36	0.12	0.62	0.43	-0.03	0.51
LINE-1-nc	KMeans	0.10	0.08	0.22	0.23	-0.22	0.22
LINE-1-nc	HDBScan	0.08	0.08	0.21	0.15	-0.16	0.17
LINE-2-nc	KMeans	0.20	0.12	0.31	0.34	-0.11	0.32
LINE-2-nc	HDBScan	0.14	0.04	0.28	0.21	-0.10	0.24

	Clustering	DB size	Window	RW len.
leftSkipgram-lc	KMeans	1M	7	10
leftSkipgram-lc	HDBScan	1M	5	20
normalSkipgram-lc	KMeans	1M	2	20
normalSkipgram-lc	HDBScan	1M	2	20
leftSkipgram-nc	KMeans	500K	7	20
leftSkipgram-nc	HDBScan	1M	7	20
normalSkipgram-nc	KMeans	1M	7	15
normalSkipgram-nc	HDBScan	500K	3	20

Tabella 4.44: Risultati di Doc2Vec e TF-IDF in Stanford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
Doc2Vec	KMeans	0.24	0.08	0.36	0.33	-0.32	0.34
Doc2Vec	HDBScan	0.13	0.07	0.34	0.18	-0.14	0.23
TF-IDF	KMeans	0.35	0.13	0.47	0.44	-0.30	0.45
TF-IDF	HDBScan	0.15	0.06	0.37	0.19	-0.07	0.25

Tabella 4.45: Risultati della configurazione Combinato in Stanford

ListConstraint	Clustering	AMI	ARI	Com	Hom	Silh	V-M
left + Doc2Vec	KMeans	0.36	0.18	0.44	0.45	-0.05	0.44
left + Doc2Vec	HDBScan	0.28	0.10	0.57	0.33	-0.03	0.42
left + TF-IDF	KMeans	0.33	0.11	0.47	0.42	-0.05	0.44
left + TF-IDF	HDBScan	0.16	0.07	0.56	0.20	-0.04	0.29
normal + Doc2Vec	KMeans	0.46	0.27	0.54	0.53	-0.03	0.53
normal + Doc2Vec	HDBScan	0.08	0.03	0.43	0.11	-0.03	0.18
normal + TF-IDF	KMeans	0.26	0.07	0.42	0.35	-0.12	0.38
normal + TF-IDF	HDBScan	0.16	0.07	0.56	0.20	-0.03	0.29
NoConstraint							
left + Doc2Vec	KMeans	0.34	0.13	0.61	0.42	-0.04	0.50
left + Doc2Vec	HDBScan	0.32	0.13	0.69	0.36	-0.02	0.47
left + TF-IDF	KMeans	0.40	0.19	0.51	0.47	-0.04	0.49
left + TF-IDF	HDBScan	0.14	0.08	0.49	0.17	-0.00	0.25
normal + Doc2Vec	KMeans	0.30	0.10	0.45	0.40	-0.04	0.42
normal + Doc2Vec	HDBScan	0.36	0.12	0.59	0.43	-0.04	0.50
normal + TF-IDF	KMeans	0.40	0.15	0.55	0.48	-0.08	0.52
normal + TF-IDF	HDBScan	0.16	0.09	0.62	0.19	-0.02	0.29

Tabella 4.46: Risultati finali di Illinois

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
normal-nc	KMeans	0.69	0.61	0.70	0.87	-0.04	0.78
Doc2Vec	HDBScan	0.76	0.83	0.78	0.77	-0.03	0.77
left-nc + TF-IDF	HDBScan	0.78	0.76	0.83	0.79	-0.03	0.81
		DB size	Window	RW len.			
normal-nc	100K		2	20			
left-nc + TF-IDF	1M		2	20			

Tabella 4.47: Risultati finali di Oxford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
left-nc	KMeans	0.58	0.52	0.59	0.68	-0.04	0.63
TF-IDF	KMeans	0.60	0.53	0.61	0.73	-0.00	0.67
left-nc + TF-IDF	KMeans	0.68	0.60	0.68	0.80	-0.04	0.74
		DB size	Window	RW len.			
left-nc	1M		5	20			
left-nc + TF-IDF	1M		5	20			

Tabella 4.48: Risultati finali di Princeton

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
normal-lc	KMeans	0.49	0.65	0.49	0.67	-0.02	0.57
TF-IDF	KMeans	0.57	0.68	0.58	0.71	0.05	0.64
normal-nc + TF-IDF	KMeans	0.53	0.65	0.54	0.73	-0.03	0.62
		DB size	Window	RW len.			
normal-lc	1M	5		10			
normal-nc + TF-IDF	1M	7		10			

Tabella 4.49: Risultati finali di Stanford

	Clustering	AMI	ARI	Com	Hom	Silh	V-M
normal-lc	KMeans	0.47	0.26	0.57	0.53	-0.03	0.55
TF-IDF	KMeans	0.35	0.13	0.47	0.44	-0.30	0.45
normal-lc + Doc2Vec	KMeans	0.46	0.27	0.54	0.53	-0.03	0.53
		DB size	Window	RW len.			
normal-lc	1M	2		20			
normal-lc + Doc2Vec	1M	2		20			

Capitolo 5

Conclusioni e sviluppi futuri

In questa tesi si è trattato del Clustering di pagine Web, proponendo un nuovo metodo che combina l'informazione estratta dal contenuto dei testi delle pagine e quella dalla struttura ad hyperlink del sito Web in un singolo spazio vettoriale, che può essere usato dagli algoritmi di Clustering tradizionali meglio performanti. Durante la sperimentazione, si è cercato di capire se l'utilizzo di Skip-Gram che considera solo il contesto sinistro potesse migliorare la qualità dei raggruppamenti prodotti dai vari algoritmi di Clustering, se effettivamente combinare l'informazione del contenuto e della struttura potesse aumentare le performance del processo di raggruppamento e se utilizzare le Liste Web per ridurre il rumore potesse migliorare i risultati del Clustering. I risultati della sperimentazione ci mostrano che il testo delle pagine e la struttura del sito Web forniscono informazioni diverse e complementari che possono migliorare l'efficacia degli algoritmi di Clustering. Non sono state riscontrate differenze statisticamente significative nell'utilizzo delle liste di costrizione e nell'applicazione di Skip-Gram modificato.

Futuri lavori potrebbero incentrarsi sull'applicazione della metodologia descritta in questa tesi su più siti Web e meno strutturati, in modo da osservare se l'uso delle Liste influenza il processo di Clustering delle pagine Web.

Bibliografia

- [1] B. S. Anami, R. S. Wadawadagi, and V. B. Pagi. Machine learning techniques in web content mining: A comparative analysis. *Journal of Information and Knowledge Management (JIKM)*, 13(1), 2014.
- [2] Ralitsa Angelova and Stefan Siersdorfer. A neighborhood-based approach for clustering of linked document collections. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, number 2 in CIKM '06, pages 778–779, New York, NY, USA, 2006. ACM.
- [3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(19):1137–1155, mar 2003.
- [4] Paul Bohunsky and Wolfgang Gatterbauer. Visual structure–based web page clustering and retrieval. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 1067–1068, 2010.
- [5] Morteza Haghir Chehreghani, Hassan Abolhassani, and Mostafa Haghir Chehreghani. Improving density-based methods for hierarchical clustering of web pages. *Data Knowl. Eng.*, 67(1):30–50, October 2008.
- [6] G. Convertino, L. Di Pace, P. Leo, A. Maffione, D. Malerba, and G. Vespucci. Tecniche di web mining per supportare l’attività di navigazione in rete. (2):12.

- [7] Daniel Crabtree, Peter Andreae, and Xiaoying Gao. Query directed web page clustering. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, number 9 in WI '06, pages 202–210, Washington, DC, USA, 2006. IEEE Computer Society.
- [8] Valter Crescenzi, Paolo Merialdo, and Paolo Missier. Clustering web pages based on their structure. *Data and Knowledge Engineering*, 54:279–299, 2005.
- [9] Marco Cristo, Pável Calado, Edleno Silva de Moura, Nivio Ziviani, and Berthier Ribeiro-Neto. *Link Information as a Similarity Measure in Web Classification*, pages 43–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [10] Isabel Drost, Steffen Bickel, and Tobias Scheffer. *Discovering Communities in Linked Data by Multi-view Clustering*, pages 342–349. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [11] Mirela-Stefania Duma and Wolfgang Menzel. Data selection for it texts using paragraph vector. In *Proceedings of the First Conference on Machine Translation*, pages 428–434, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [12] M. Fisher and R. Everson. When are links useful? experiments in text classification. In *Proceedings of the 25th European Conference on IR Research*, pages 41–56, 2003.
- [13] Justin Garden, Kenji Sagae, Volkan Ustun, and Morteza Dehghani. Combining distributed vector representations for words. In *Proceedings of NAACL-HLT 2015*, pages 85–101, June 2015.
- [14] O. Gornerup, D. Gillblad, and T. Vasiloudis. Knowing an object by the company it keeps: A domain-agnostic scheme for similarity discovery. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*, ICDM '15, pages 121–130, Washington, DC, USA, 2015. IEEE Computer Society.

- [15] Taher H. Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. Evaluating strategies for similarity search on the web. In *Proceedings of the 11th International Conference on World Wide Web*, number 11, pages 432–442, New York, NY, USA, 2002. ACM.
- [16] Xiaofeng He, Hongyuan Zha, Chris H. Q. Ding, Horst D. Simon, Horst, and D. Simon. Web document clustering using hyperlink structures, 2001.
- [17] Seokho Hong. Improving paragraph2vec.
- [18] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- [19] M. M. Kessler. Bibliographic coupling between scientific papers. In *American Documentation*, volume 14, pages 10–25, 1963.
- [20] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, sep 1999.
- [21] Omer Levy, Yoav Goldberg, and Ido Dogan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3(211-225), 2015.
- [22] C. X. Lin, Y. Yu, J. Han, and B. Liu. Hierarchical web-page clustering via in-page and cross-page link structures. In *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 222–229, 2010.
- [23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*.
- [24] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*, pages 746–751. Association for Computational Linguistics, June 2013.

- [25] Dharmendra S. Modha and W. Scott Spangler. Clustering hypertext with applications to web searching. In *Proceedings of the Eleventh ACM on Hypertext and Hypermedia*, number 10 in HYPERTEXT '00, pages 143–152, New York, NY, USA, 2000. ACM.
- [26] Christopher Olston and Marc Najork. Web crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246, 2010.
- [27] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.
- [28] N. V. Pardakhe and R. R. Prof. Keole. Analysis of various web page ranking algorithms in web structure mining. 2:6, 2013.
- [29] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, New York, NY, USA, 2014. ACM.
- [30] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in NLP and Computational Natural Language Learning(EMNLP-CoNLL)*, number 410-420, page 10, 2007.
- [31] Magnus Sahlgren. *The Word-Space Model*. PhD thesis, University of Stockholm (Sweden), 2006.
- [32] Cyrus Shahabi, Amir M. Zarkesh, Jafar Adibi, and Vishal Shah. Knowledge discovery from users web-page navigation. In *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications*, RIDE '97, Washington, DC, USA, 1997. IEEE Computer Society.

- [33] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269, 1973.
- [34] Jaideep Srivastava, Prasanna Desikan, and Vipin Kumar. *Web Mining - Concepts, Applications, and Research Directions*, chapter 21, pages 275–307.
- [35] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Cluster Analysis: Basic Concepts and Algorithms*, chapter 8, pages 488–568. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, first edition edition, 2005.
- [36] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*. ACM, 2015.
- [37] Lita Van Wel and Lambèr Royakkers. Ethical issues in web data mining. *Ethics and Inf. Technol.*, 6(12):129–140, 2004.
- [38] Yitong Wang and Masaru Kitsuregawa. Evaluating contents-link coupled web page clustering for web search results. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, number 8 in CIKM ’02, pages 499–506, New York, NY, USA, 2002. ACM.
- [39] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’98*, pages 46–54, New York, NY, USA, 1998. ACM.
- [40] Y. Zhou, H. Cheng, and J. X. Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM ’10*, pages 689–698, Washington, DC, USA, 2010. IEEE Computer Society.
- [41] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th*

BIBLIOGRAFIA

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 487–494, New York, NY, USA, 2007. ACM.