UNIVERSITÀ DI PISA

Seminario di Cultura Digitale

VECTOR SPACE MODELS: un approccio alla semantica

Gloria Malorgio, matricola 428840

Abstract

Questa relazione prende spunto dal seminario sui metodi di ottimizzazione per i motori di ricerca.
Dopo una breve panoramica sulle tecniche che riguardano il recupero di documenti e il funzionamento dei motori di ricerca dell'IR tradizionale e del web, si analizzerà brevemente il vector space model, un modello sviluppato con l'intento di superare alcuni dei limiti delle metodologie precedenti.
In particolare, si accennerà brevemente ai criteri di approccio alla semantica che si basano sull'uso dei VSM.

1. Da Information Retrieval a Web Information Retrieval

L'information retrieval è l'insieme delle tecniche utilizzate per gestire la rappresentazione, la memorizzazione, l'organizzazione e l'accesso ad oggetti contenenti risorse quali documenti, pagine web o qualunque altro tipo di materiale informativo.
Lo scopo dell'IR è quello di soddisfare il cosiddetto "bisogno informativo dell'utente", ovvero garantire a quest'ultimo, in seguito ad una sua ricerca, i documenti e le informazioni che rispondono alla sua richiesta. Due concetti sono di fondamentale importanza: **query** e **oggetto**.
Le query sono stringhe che contengono parole-chiave che rappresentano l'informazione richiesta. Vengono digitate dall'utente in un sistema IR e sono la concretizzazione di ciò che egli sta cercando.
Un oggetto è un'entità che possiede informazioni le quali potrebbero essere risposta all'interrogazione dell'utente.

La nascita dell'information retrieval si colloca intorno agli anni '40. Con l'invenzione del computer infatti, nascono i primi sistemi che per mezzo di motori di ricerca recuperano i documenti all'interno di file e cartelle.
A partire dagli anni '60 vengono sviluppati tre principali modelli:
-boolean search engine: uno dei metodi di recupero più semplice, sfrutta la nozione di matching tra query e documenti secondo i principi dell'algebra booleana, nella quale le parole sono combinate per mezzo degli operatori AND, NOT e OR. Il limite di questo modello sta nel fatto che non esiste un concetto di matching parziale tra la query e l'informazione da restituire e di conseguenza il documento o è rilevante o non lo è. Anche i sinonimi e la polisemia rappresentano un ostacolo importante, forse lo svantaggio principale di questo metodo, che, nonostante tutto, resta ancora oggi alla base di vari motori di ricerca del web.
-vector space model search engine: sviluppato da Gerard Salton, supera in parte i limiti del modello precedente, in particolare a livello semantico. I dati testuali vengono trasformati in vettori numerici e quindi in matrici, dalla cui analisi possono emergere somiglianze semantiche strutturali all'interno del testo. Inoltre, a differenza dei motori di ricerca booleani la rilevanza di

un documento viene valutata in ottica *fuzzy* e assume quindi un valore compreso tra 0 e 1, ovvero la probabilità che quel documento sia pertinente alla query digitata dall'utente. Grazie a questo "punteggio", noto come *relevance scoring*, i documenti vengono restituiti in una lista ordinata in base al loro grado di rilevanza. Uno svantaggio di questo modello riguarda la complessità computazionale. Ogni ricerca richiede infatti che venga calcolata la distanza (similarity) tra la query e tutti i documenti dando origine a matrici di dimensioni proibitive nel caso di un ampio numero di documenti.

-**probabilistic model search engine:** tali modelli cercano di stimare la probabilità che un utente trovi utile un determinato documento già ritenuto pertinente. Questi modelli sono complessi e difficili da programmare, inoltre fanno assunzioni di indipendenza statistica non realistiche tra i termini di un documento e i documenti stessi. Queste ipotesi si rivelano restrittive in molti casi, si pensi ad esempio alle collocazioni o alle espressioni polirematiche.

Esiste in realtà anche un quarto modello di motore di ricerca tradizionale, il **meta-search engine** che combina le features dei tre modelli classici basandosi sul principio che se un motore di ricerca procura buoni risultati, due o più ne apporteranno sicuramente di migliori. Esempi di questo tipo sono Copernic (**www.copernic.com**) e SurfWax (**www.surfwax.com**). Il loro funzionamento è riassumibile come segue: una volta digitata, la query viene inviata a più motori di ricerca e il risultato è un'unica lista che contiene tutti i risultati ottenuti.

Nel 1989, con la nascita del World Wide Web, l'accesso, la memorizzazione e il recupero di documenti sono soggetti ad una rivoluzione senza precedenti.

L'information retrieval tradizionale, che si occupava del recupero di documenti contenuti in collezioni "fisicamente" disponibili (come articoli o libri) o anche in formato elettronico (ad esempio in cd) si evolve in "web information retrieval".

Nel 1998, con l'avvento della link analysis, che nella teoria delle reti è una tecnica che si usa per valutare le relazioni (connessioni) tra i nodi, i motori di ricerca iniziano a sfruttare l'informazione aggiuntiva contenuta nelle pagine web, la loro ipertestualità.

Alle metodologie di base proprie dell'information retrieval tradizionale se ne affiancano di nuove per poter far fronte alle nuove esigenze imposte dal web, di fatto la più vasta collezione di documenti che sia mai stata creata.

-Il web è **enorme**:

A tutto il 2013 il contenuto del web è stato stimato in oltre di 13 trilioni di pagine attive, e questo se si considera soltanto il *surface web*, ovvero i contenuti pubblicamente indicizzabili da parte dei vari motori di ricerca. Secondo uno studio condotto nel 2000 da Bright Planet[1], Google indicizzerebbe soltanto meno dell'uno per cento dei documenti presenti in rete, anche se è difficile fornire una stima precisa delle reali dimensioni dell'iceberg che si nasconde al di sotto del web di dominio comune. Secondo BrightPlanet si tratterebbe di oltre 550 bilioni di pagine che, proprio perchè non indicizzate, risultano difficili da reperire.

-il web è **dinamico**:

a differenza di quanto succedeva con le collezioni di documenti tradizionali, nel web le pagine sono in continuo aggiornamento.

In uno studio del 2000, Cho e Garcia-Molina[2] affermano che il contenuto delle pagine del loro dataset cambiava ogni settimana e che il 23% di quelle .com addirittura giornalmente, e, con esso, nella maggior parte dei casi, anche la loro *size*. Senza dimenticare che ogni anno vengono create milioni e milioni di nuove pagine. Qualunque tipo di task diventa più complesso quando si ha a che fare con una collezione di documenti in costante evoluzione.

-il web è **auto-organizzato**:

nessuno si occupa di organizzare e categorizzare i documenti presenti sul web. Chiunque può creare una pagina e non esistono “controllori” riguardo ai contenuti, alla struttura o al formato. I dati sono volatili ed eterogenei, esistono in varie lingue e vari alfabeti. Non esiste un processo di revisione editoriale che ci tenga alla larga da eventuali errori e falsità.

-Il web è **“hyperlinked”**:

questa caratteristica è il punto forte dei motori di ricerca. Grazie alla struttura ipertestuale del web e grazie alle metodologie di ranking, che sfruttano proprio i collegamenti fra le varie pagine, i risultati sono sempre più accurati.

---

1 https://it.wikipedia.org/wiki/Web_invisibile

2 Jungoo Cho Hector Garcia-Molina, *The evolution of the web and implication for an ncremental crawler*. In Proceedings of the twenty-sixth International conference on very large database, pp.. 200-209, ACM Press.New York, 2000.

2. Come funziona un web search engine

Quando un utente interroga un motore di ricerca, i risultati appaiono rapidamente nel giro di una manciata di millisecondi. A prima vista, sembrerebbe che il motore di ricerca esamini tutte le pagine web e ne estragga i risultati in quella piccola frazione di tempo. Però, se un motore di ricerca dovesse esaminare in modo sequenziale ogni parola nei milioni di documenti presenti nel web ci vorrebbero parecchie ore. In realtà, quasi tutto il lavoro svolto dai motori di ricerca avviene prima che qualcuno digiti la query. Si distinguono dunque due fasi. Una fase **query independent**, ovvero una serie di processi che esistono indipendentemente dalle interrogazioni degli utenti, e una fase **query dependent.**

Il primo step di questo processo query independent è compiuto dal *crawler*. Il crawler è un software che analizza i contenuti di un documento web utilizzando dei robots chiamati *spiders* che vengono istruiti sulle pagine da visitare a partire da un set di URLs fornito in input dal motore di ricerca. Durante l'analisi di un URL, il crawler identifica tutti gli hiperlink presenti nel documento e li aggiunge alla lista di URL da visitare. Di conseguenza, se esistono delle pagine del sito che non sono collegate da link, vengono ignorate dal crawler e non saranno indicizzate. Poichè il web è dinamico e i contenuti di una pagina potrebbero cambiare da un giorno all'altro, affinchè gli indici siano costantemente aggiornati il *crawling* non può che essere un processo continuo.

Gli spiders restituiscono quindi delle pagine web che vengono temporaneamente immagazzinate nel cosiddetto *page repository.* Lo step successivo è quello di estrapolare le informazioni vitali per crearne una versione compressa che verrà poi memorizzata in diversi indici all'interno di uno o più database del motore di ricerca.

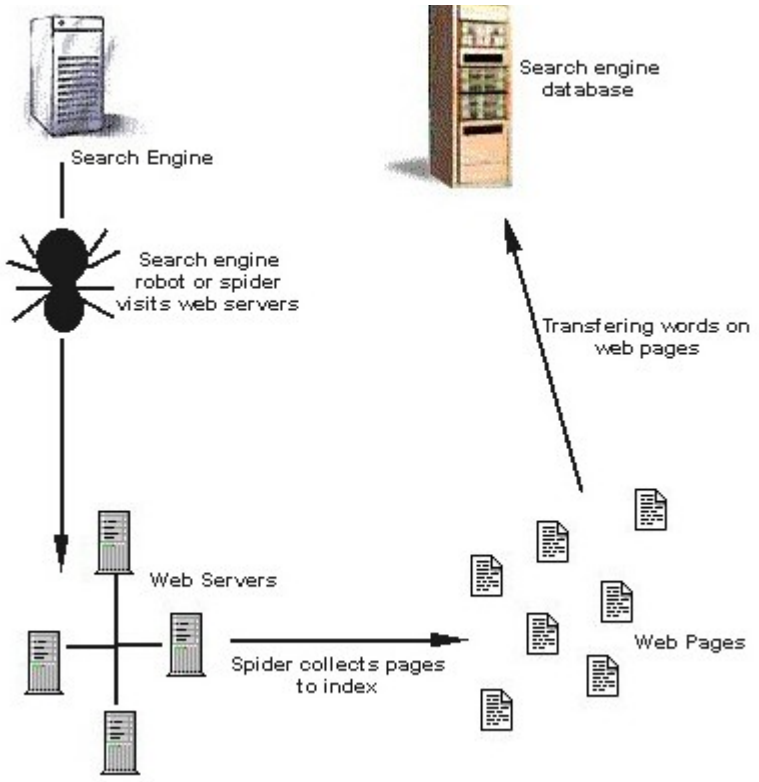

Figura 1: l'attività del crawler/spider. Fonte http://demandgeneration.typepad.com/

Gli indici sono dunque il prodotto finale della fase query-independent e racchiudono tutte le informazioni di interesse di una pagina web.

Ogni nuova pagina recuperata dagli spiders viene analizzata da software dedicati che si occupano del *parsing* e dell'estrapolazione di informazioni relative al suo contenuto, generalmente presenti nel titolo, nella descrizione e nel testo inserito tra gli *anchor tag*.

Tra i vari indici, uno di quelli più importanti è sicuramente quello noto come **content index** o indice di contenuto. Per memorizzare questa informazione si utilizza un indice inverso (**inverted index**), simile a quello che si trova nella parte finale di un libro.

Accanto ad ogni termine di interesse vengono elencate le pagine in cui esso compare. Nel caso più semplice le pagine vengono indicate con un numero che ne è l'identificativo, ma esistono anche indici organizzati in maniera molto più complessa che tengono traccia anche di altre

features.

Ad esempio, si potrebbe voler tenere traccia del fatto che la parola in questione compaia o meno nel titolo, nel meta-tag o annotare il numero di volte in cui quel determinato termine compare all'interno del documento. Una soluzione potrebbe essere quella di inserire un vettore tridimensionale dopo ogni identificativo della pagina.

La fase query-dependent consiste di due moduli: una prima fase in cui una query in linguaggio naturale viene convertita in un linguaggio comprensibile ai sistemi di ricerca e vengono consultati i vari indici per identificare i documenti pertinenti, e una seconda fase, nota come fase di **ranking**, in cui i documenti ritenuti rilevanti vengono valutati e ordinati secondo determinati criteri. Il risultato è una lista ordinata di pagine web, dalla più rilevante alla meno rilevante.

La fase di ranking è forse uno dei componenti più importanti del processo di ricerca e il suo potere di filtraggio dei risultati rende più semplice all'utente muoversi tra le centinaia di migliaia di pagine che vengono recuperate.

Il ranking prende in considerazione due “punteggi”: il **content score** e il **popularity score**.

Esistono regole o euristiche diverse nell'attribuzione di entrambi i punteggi. Ad esempio alcuni motori di ricerca attribuiscono un alto content score a pagine in cui la query word è presente nel titolo o nella descrizione della pagina e penalizzano quelle in cui la parola appare ad esempio all'interno del body.

In ogni caso il content score può essere computato solamente a partire dall'indice di contenuto ed quindi è query-dependent.

Il popularity score varia invece in base allo structure index, ossia a seconda della maniera in cui una pagina è collegata alle altre tramite link e dunque dipende strettamente dalla sua struttura ipertestuale.

Content e popularity score vengono combinati tra loro per ottenere un punteggio complessivo per ciascuna pagina e il set delle pagine rilevanti viene presentato all'utente in una lista ordinata di risultati.

3. Vector Space Models

Precedentemente si è accennato a tre diversi modelli tipici dell'information retrieval tradizionale. In questa sezione si forniscono brevi cenni su uno di questi, il **vector space model**.

Il VSM rappresenta le query e i documenti scritti in linguaggio naturale come vettori in uno spazio multidimensionale.

L'identificazione dei documenti rilevanti per la query avviene mediante il confronto della sua rappresentazione vettoriale con quella di ciascun documento appartenente al corpus di interesse.

La versione classica di VSM (Salton, Wong&Yang, 1975) include un parametro che tiene conto della frequenza del temine nel documento e un parametro che considera il potere discriminante del termine all'interno del corpus.

Tale indice è noto come **tf-idf** è dato dal prodotto di due fattori:

$$(\text{tf-idf})_{\text{i,j}} = \text{tf}_{\text{i,j}} \times \text{idf}_{\text{i}}$$

il primo è la frequenza del termine *t* all'interno del documento *d*

$$\text{tf}_{\text{i,j}} = \frac{n_{i,j}}{|d_j|}$$

Il secondo indica l'importanza generale del termine nella collezione:

$$\text{idf}_{\text{i}} = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

dove |D| è il numero dei documenti della collezione e il denominatore è dato dal numero di documenti d che contengono il termine *t.*

Ovviamente le parole che compaiono in un documento non hanno tutte la stessa importanza; alcune non forniscono nessuna informazione mentre altre sono caratterizzanti e informative.

L'idea alla base dell'*inverse document frequency* consiste nel dare più importanza ai termini che compaiono nel documento, ma che in generale sono poco frequenti.

L'indice tf-idf assegna dunque a un termine *t* presente all'interno di un documento *d* un peso *w* che sarà:

-alto se t è presente spesso all'interno di una collezione composta da un esiguo numero di documenti;

-più basso se t è presente meno volte all'interno di un documento o compare in molti documenti;

-molto basso se presente in tutti i documenti.

Possiamo quindi considerare ogni documento come un vettore costituito da un termine appartenente al vocabolario di termini presenti nel documento a cui è associato un peso corrispondente all'indice tf-idf.

L'overlap score measure, ovvero la rilevanza di un documento in base alla query è dato dalla seguente formula:

$$\text{Score}(q,d) = \sum_{t \in q} \text{tf-idf}_{t,d}.$$

ovvero la somma, per ciascun termine della query q, del numero di volte (o, in questo caso, del peso dato dall'indice tf-idf) in cui il termine compare all'interno del documento.

Come computare dunque la similarità tra vettori?

Se computassimo la differenza tra vettori prendendo in considerazione il loro indice tf-idf i risultati potrebbero non essere attendibili a causa della diversa lunghezza dei documenti.

Per compensare questo problema la similarità viene calcolata in base al coseno dell'angolo compreso tra i due vettori.

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|},$$

Il numeratore rappresenta il loro **dot product**, mentre il denominatore è il prodotto delle loro lunghezze euclidee. In questo modo i vettori vengono normalizzati, ovvero assumono entrambi lunghezza pari all'unità.
Dato un documento *d*, cercare il documento più simile all'interno di un corpus si riduce quindi a calcolare il dot product tra *d* e tutti gli altri e scegliere quello che ha valore maggiore.
Di conseguenza, è possibile usare il coseno per misurare la rilevanza di un documento in base a una query che, per quanto breve, viene comunque considerata uno pseudo-documento.

3.1 Pre-processing linguistico

Prima di poter applicare le tecniche di VSM ad un testo in linguaggio naturale può essere necessario trattarlo e dotarlo di un insieme di informazioni aggiuntive.
Passo preliminare per qualsiasi elaborazione di un testo è la sua **tokenizzazione**, ovvero bisogna decidere quali sono le unità minime di analisi (token) e come estrarle a partire da un testo non strutturato.
Il secondo step consiste nella normalizzazione. Un testo normalizzato è un testo in cui le parole che condividono la stessa radice morfologica vengono ricondotte ad uno stesso “stem”.
Infine, può essere necessario anche un processo di annotazione, ad esempio per distinguere stringhe di caratteri uguali che però hanno significato diverso.

a. Tokenizzazione

Gli ostacoli che si possono riscontrare in questa fase sono vari. Il primo problema è dato dai delimitatori: essi dipendono fortemente dalla lingua adoperata nel testo e possono essere

presenti irregolarità legate ad esempio a convenzioni grafiche. Un buon tokenizzatore deve riuscire a gestire bene la punteggiatura , gli acronimi, le date, e riconoscere token graficamente complessi (nomi composti, espressioni polirematiche).
Un approccio più sofisticato consiste nell'utilizzare un lessico o un'ontologia ma spesso tale tecnica dà luogo a tokenizzazioni non univoche (Sproat & Emerson, 2003). Sicuramente la tokenizzazione è uno dei task più complessi per la maggior parte delle lingue naturali.

b. Normalizzazione

Il motivo principale per cui tale step si rivela il più delle volte indispensabile è legato al fatto che stringhe di caratteri diversi veicolano talvolta lo stesso significato. Visto che lo scopo principale è quello di comprendere il significato generale del documento in questione, sembra ragionevole uniformare la grafia eliminando la variabilità delle parole. I tipi più comuni di normalizzazione sono il **case folding** (convertire tutte le maiuscole in minuscole) e lo **stemming** (ricondurre le parole flesse alla loro radice morfologica condivisa).
In entrambi i casi esistono varie problematiche legate alla lingua del testo. Se infatti, per lingue come l'inglese, gli algoritmi e le euristiche di stemming sembrano funzionare abbastanza bene, per lingue agglutinanti come il turco, il task diventa molto più complesso e i risultati meno soddisfacenti. In una lingua agglutinante infatti, un singolo termine può “contenere” al suo interno anche una frase di 5-6 parole. (Johnson & Martin, 2003).

c. Annotazione

L'annotazione è il processo inverso alla normalizzazione. Così come stringhe di caratteri diversi possono veicolare il medesimo significato, succede anche che stringhe identiche abbiano significati diversi a seconda del contesto. Il processo tipico di annotazione include il *part-of-speech tagging*, il *word sense tagging* e il *parsing*.

3.2 Vector Space Models e semantica

Il filo conduttore che lega le varie tipologie di VSM applicate alla semantica a cui si accennerà brevemente è quello di una teoria semantica generale basata sulla statistica: i pattern statistici soggiacenti all'uso delle parole possono essere sfruttati per comprendere i significati che gli utenti vogliono comunicare per mezzo delle parole.

Se consideriamo un insieme di N documenti come un insieme di N vettori, risulta conveniente organizzarli in matrici. Esistono tre ampie classi di VSM che fanno uso di diversi tipi di matrici tra cui term-document e word-context matrix.

Nel caso della **term-document matrix** le righe della matrice rappresentano i termini (le dimensioni, generalmente le parole) e le colonne i documenti (ad esempio le pagine web).

In una term-document matrix, un vettore rappresenta il documento corrispondente come una **"bag of words"**. Una bag of words (nota anche come *multiset*) è un insieme di elementi non ordinati in cui non sono ammessi duplicati. Dunque le seguenti bag of words:

{a,a,b,c,c,c}

{c,a,c,b,a,c}

sono equivalenti.

A questo punto è possibile, ad esempio, rappresentare i due insiemi con il vettore

x = < 2,1,3>, che tiene conto delle frequenze delle parole all'interno del set.

In information retrieval la **bag of word hypothesis** sostiene come sia possibile stimare la rilevanza di un documento in relazione a una query rappresentandoli entrambi per mezzo di una bag of words, il che equivale a dire che la sola frequenza delle parole in un documento è indice di rilevanza. Nonostante l'ordine delle parole non venga mantenuto, sembra comunque che i motori di ricerca reagiscano sorprendentemente bene e che quindi tale metodo di rappresentazione vettoriale riesca a catturare alcuni aspetti importanti della semantica.[3]

La "bag of word hypotesis", la cui applicazione si riflette nella term-document matrix, è l'idea di base per poter applicare VSM all'information retrieval (Salton et al.,1975). Una

3 From Frequency to Meaning: Vector Space Models of Semantics, Journal of Artificial Intelligence Research 37 (2010) 141-188

giustificazione abbastanza intuitiva potrebbe essere data dal fatto che l'argomento di un documento influenzerà probabilisticamente la scelta delle parole da parte dell'autore. Se due documenti trattano argomenti simili, allora i vettori delle colonne corrispondenti avranno pattern simili.

Salton et al. (1975) si concentrano sulla similarità tra documenti ritenendo che la pertinenza di un documento a una query sia data dalla similarità dei loro vettori.

Secondo Deerwester et al. (1990) bisognerebbe focalizzarsi sulla similarità tra parole anziché tra documenti, prendendo in considerazione non le colonne della matrice bensì le righe, partendo dall'assunto che un documento possa non essere quantitativamente adatto per valutare la word similarity.

La **Distributional Hypostesis** afferma che le parole che compaiono in contesti simili tendono ad avere significati simili (Harris, 1954). Questa ipotesi è alla base dell'applicazione dei VSM nella misurazione della word similarity. Una parola può essere rappresentata da un vettore in cui gli elementi sono derivati dall'occorrenza della parola stessa in vari contesti, ad esempio utilizzando finestre di parole o dipendenze grammaticali (Lund & Burgess, 1996). Simili righe di vettori nella **word-context matrix** stanno ad indicare significati simili.

Weaver sostiene che le tecniche di word sense disambiguation per apprendimento automatico dovrebbero basarsi sulle co-occorrenze delle parole di contesto di una determinata parola target, quella che si vuole disambiguare.

I principali approcci alternativi ai VSM che misurano la similarità semantica tra parole fanno uno di lessici come, ad esempio, WordNet (Resnik, 1995; Jiang & Conrath, 1997; Hirst & St-Onge, 1998; Leacock & Chodrow, 1998; Budanitsky & Hirst, 2001). L'idea è quella di considerare il lessico come un grafo in cui i nodi corrispondono ai sensi e gli archi rappresentano le relazioni di iperonimia e iponimia tra parole. La similarità tra due termini sarà proporzionale alla lunghezze del cammino del grafo che connette i nodi dei due termini.

4. Conclusioni

VSM è stato sviluppato con l'intento di risolvere alcuni dei limiti associati alle tecniche di matching lessicale tra query e documenti, principalmente quelli legati a polisemia e sinonimia.
Poiché le parole sono spesso polisemiche, diventa infatti difficile capire se un termine condiviso da più documenti abbia significati diversi facendo astrazione dal contesto in cui si trova.
Allo stesso modo, alcuni documenti potrebbero far uso di vocaboli dal significato simile per descrivere lo stesso concetto.
Rappresentando query e documenti in uno spazio vettoriale e computandone la similarità, VSM, a differenza di altre tecniche di matching che classificano i risultati ottenuti semplicemente in base al numero di occorrenze della query word, è in grado di indirizzare l'utente verso documenti concettualmente simili e più rilevanti di altri ma esiste comunque il rischio di sovrastimare, nel caso della polisemia, o sottostimare, nel caso dei sinonimi, la reale somiglianza tra i vettori della query e del documento.

LSI, il cui acronimo sta per **Latent Semantic Index** è un metodo che esamina il contenuto di un documento a livello globale e cerca di trovare documenti correlati basandosi sul principio che parole usate nello stesso contesto tendono ad avere significati simili riuscendo a recuperare documenti dal topic simile anche quando all'interno di questi siano presenti parole diverse da quelle contenute nella query.
L'assunzione di base è che esista una struttura semantica “latente” parzialmente oscurata dalla variabilità linguistica nella scelta delle parole.
La **Latent Semantic Analysis** si serve di metodi statistico-matematici per eliminare il “rumore” e sfruttare, durante la ricerca, le relazioni semantiche emergenti tra termini e documenti.
Numerosi scienziati cognitivi sostengono che esistano ragioni teoretiche ed empiriche per credere che VSM come LSA siano modelli plausibili di alcuni aspetti della cognizione umana (Landauer et al., 2007).

Le tecniche di Vector Space Model, sviluppate per il sistema di information retrieval SMART (Salton, 1971)**,** sono state pioniere di molti dei principi che stanno alla base del funzionamento dei moderni motori di ricerca. (Manning, Raghavan, & Schütze, 2008) e il successo di tali metodologie nell'ambito dell'information retrieval ha spinto i ricercatori ad estendere l'applicazione di tal modelli ad altri settori del *natural language processing*. Rapp (2003) ha utilizzato rappresentazioni VSM per rispondere a un test a risposte multiple TOEFL (Test of english as a second language) ottenendo un punteggio del 92.5% mentre la media degli utenti umani si attestava solo sul 64,5%.

I vector space models sono in grado di estrrre conoscenza in maniera automatica a partire da un dato corpus e richiedono uno sforzo di gran lunga inferiore in confronto ad altri approcci alla semantica, ad esempio, la costruzione di un'ontologia.

Bibliografia

Budanitsky, A., & Hirst, G. (2001). *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*. In Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001), pp. 29–24, Pittsburgh, PA.

Deerwester S., Dumais T., Furnas G.W., Landauer T.K., Harshman R., *Indexing by latent semantic analysis (1990)*, Journal of the American Society for Information Science, 41 (6), pp. 391–407.

Harris, Z. (1954) *Distributional structure*. Word, 10 (23), pp.146–162.

Hirst, G., & St-Onge, D. (1998). *Lexical chains as representations of context for the detection and correction of malapropisms*. In Fellbaum, C. (Ed.), WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press.

Jiang, J. J., & Conrath, D. W. (1997). *Semantic similarity based on corpus statistics and lexical taxonomy*. In Proceedings of the International Conference on Research in Computational Linguistics (ROCLING X), pp. 19–33, Tapei, Taiwan.

Johnson, H., & Martin, J. (2003). *Unsupervised learning of morphology for English and Inuktitut*. In Proceedings of HLT-NAACL 2003, pp. 43–45.

Jungoo Cho and Hector Garcia-Molina. *The evolution of the web and implication for an ncremental crawler.* In Proceedings of the twenty-sixth International conference on very large database, pp. 200-209, New York, 2000. ACM Press.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). *Introduction to Latent Semantic Analysis*. Discourse Processes, pp. 25, 259-284.

Langville Amy N., Meyer Carl D., *Google's PageRank and Beyond*, Princetown University Press, 2006.

Leacock, C., & Chodrow, M. (1998). *Combining local context and WordNet similarity for word sense identification*. In Fellbaum, C. (Ed.), WordNet: An Electronic Lexical Database. MIT Press.

Lund, K., & Burgess, C. (1996). *Producing high-dimensional semantic spaces from lexical co-occurrence*. Behavior Research Methods, Instruments, and Computers, 28 (2), pp. 203– 208.

Manning D., Raghavan P., Schütze H., *An introduction to Information Retrieval*, Cambridge University Press, 2009.

Salton, G. (1971). *The SMART retrieval system: Experiments in automatic document processing*. Prentice-Hall, Upper Saddle River, NJ.

Salton G., Wong A, and Yang C. S. (1975), *A vector space model for automathic indexing*, "Communications of the ACM", vol. 18, nr. 11, pp. 613–620.

Turney P. D. , Pantel Patrick, *From frequency to Meaning: Vector Space Models of Semantics*, Journal of Artificial Intelligence Research 37 (2010) pp 141-188.

Rapp, R. (2003). *Word sense discovery based on sense descriptor dissimilarity.* In Proceedings of the Ninth Machine Translation Summit, pp. 315–322.

Resnik, P. (1995). *Using information content to evaluate semantic similarity in a taxonomy*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), pp. 448–453, San Mateo, CA. Morgan Kaufmann.

Wikipedia, voce Information Retrieval, URL https://it.wikipedia.org/wiki/Information_retrieval