UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

RELAZIONE

Visualizzatore di spazi semantici: un approccio paradigmatico.

Candidato: *Fausto Fratelli*

Relatore: *Alessandro Lenci*

Correlatore: *Maria Simi*

Anno Accademico 2011-2012

INDICE

Introduzione

Con il presente lavoro si vuole esporre la progettazione di un interfaccia grafica che permetta la visualizzazione e la navigazione di dati estratti da corpora testuali secondo le regole e le tecniche utilizzate dai modelli facenti parte della famiglia di *Distributional Semantic Memory.*

I concetti chiave che sono stati studiati per la realizzazione di questo lavoro riguardano la Semantica Distribuzionale ed in particolare un nuovo filone chiamato *Distributional Semantic Memory* nato negli ultimi anni dagli studi congiunti di Alessandro Lenci e Marco Baroni.

La semantica, in generale, è quella parte della linguistica che studia il significato delle parole, di insiemi di parole, delle frasi e dei testi; mentre la Semantica Distribuzionale, partendo da questi presupposti si occupa dello studio delle parole in relazione alla frequenza e ai tipi delle loro distribuzioni nei testi. Questa nasce in linguistica computazionale e nelle scienze cognitive come una famiglia di approcci all'analisi del significato fornendo una serie di modelli, i cosiddetti "word space models[1]", che mirano alla definizione di una rappresentazione generale del significato lessicale.

Il mio lavoro si basa principalmente su quest'ultimo aspetto della semantica, che recentemente ha attraversato un periodo di grande crescita dovuto al sempre maggiore interesse verso lo studio delle scienze cognitive e grazie alla sempre maggiore disponibilità di corpora testuali digitali.

L'innovazione che ha portato la Distributional Semantic Memory riguarda un nuovo approccio nel concepire la vicinanza semantica delle parole; tale vicinanza infatti non viene data, di per se, dal significato intrinseco della parola intesa come rappresentazione grafica di simboli, bensì deve essere ricavata dai rapporti che la parola intrattiene con le altre parole che si trovano nel medesimo spazio semantico e quindi misurata.

L'obiettivo primario di questo lavoro è quello di visualizzare le relazioni di similarità distribuzionale di tipo paradigmatico, in modo da favorire l'esplorazione degli spazi semantici estratti dai corpora.

1 Sahlgren, 2006

Per raggiungere questo scopo si e deciso di rappresentare i dati tramite un grafo dinamico (*R-Graph*) facente parte della libreria JavaScript Infovis Toolkit, che presenta una struttura *JSON* di tipo *Genitore-Figlio* in cui gli elementi si dispongono secondo delle gerarchie.

Infine, nel capitolo conclusivo di questa breve relazione descriverò in maniera più dettagliata le operazioni preliminari e i procedimenti che mi hanno permesso la realizzazione del visualizzatore di spazi semantici distribuzionali, a partire dallo studio della struttura del database, passando per le query utilizzate per interrogare il database, fino all'uso e all'integrazione dei dati con l'interfaccia grafica.

1 – Semantica Distribuzionale

1.1 Introduzione

Il concetto di semantica distribuzionale ha preso forma partendo dall'*ipotesi distribuzionale* (ID), sviluppato da Zellig Sabbettai Harris nel 1954, secondo la quale, due parole sono tanto più semanticamente simili quanto più tendono a ricorrere in contesti linguistici simili.

"Il grado di somiglianza semantica tra due espressioni linguistiche A e B è una funzione della somiglianza dei contesti linguistici in cui A e B possono co-occorrere."[2]

In seguito, molteplici studiosi del settore, avendo incentrato i propri studi su tale ipotesi, hanno riconosciuto questa come valida ponendola come principio fondamentale per le proprie ricerche.
L'ID, inoltre, trova un effettivo riscontro nelle ricerche di John Firth, il quale sosteneva che il significato di un termine dalle relazioni che intrattiene con gli altri. Questo giustifica l'idea che il significato delle parole possa essere modellato usando le informazioni contestuali rappresentate in vettori spaziali.
Grazie a Firth, infatti, vi è una valorizzazione del contesto di una parola e quindi il significato di una parola può essere dedotto dai legami che instaura con le altre espressioni linguistiche presenti nello stesso contesto.

"You shall know a word by the company it keeps. An abstraction of information in the set of natural linguistic context in which a word occurs."[3]

2 Alessandro Lenci, 2008
3 John Firth, 1957; Traduzione: *Si dovrebbe conoscere una parola basandosi sulle altre che la accompagnano. Un'informazione astratta nel contesto linguistico naturale in cui la parola occorre.*

L’Ipotesi Distribuzionale si basa essenzialmente sul misurare la distanza semantica che intercorre tra due elementi all’interno di uno spazio distribuzionale. Se ciò può ritenersi sufficiente per individuare casi di sinonimia (in quanto essa costituisce una relazione simmetrica tra due elementi), altrettanto non può essere detto per quanto riguarda altre tipologie di relazioni.
Gentner (1983), infatti, ritiene necessario operare una distinzione tra similarità attributiva e similarità relazionale.
Con l'utilizzo del modello distribuzionale, ad esempio, si riesce ad individuare i sinonimi poiché questi ultimi sono termini che denotano concetti che condividono simili attributi e, di conseguenza, parecchi contesti linguistici. Parole come “*fotografia”* e “*istantanea”* sono simili dal punto di vista attributivo poiché i loro significati condividono un’ampia classe di attributi.

La similarità attributiva tra due parole a e b, sima (a, b) ∈ R dipende dal grado di corrispondenza tra le proprietà di a e b. Maggiore è la corrispondenza, maggiore sarà la loro similarità attributiva.[4]

In certi casi la corrispondenza semantica che lega coppie di parole non è da riscontrare sul piano della similarità degli attributi come nel caso della sinonimia o della co-iponimia ma su quello della similarità relazionale. Ad esempio, si pensi a coppie del tipo “*contadino* : *terra*” e “*falegname* : *legno*”, le quali sono legate dalla relazione comune “*lavoratore* : *materiale lavorato”*.

“La similarità tra due coppie di parole a : b e c : d, simr (a:b, c:d) ∈R,dipende dal grado di corrispondenza tra le relazioni di a : b e c : d. Maggiore è la corrispondenza, maggiore sarà la loro similarità relazionale”.[5]

Un’ulteriore distinzione riguardo le modalità in cui le parole possono essere distribuite in un corpus di testi e stata fatta nel 1993 da Hinrich Schütze e Jan Pedersen, i quali hanno affermato che *“Se due parole sono tipicamente vicine l'una dell'altra, allora sono collegate in modo sintagmatico; se invece due parole hanno*

4 Peter D. Turney, 2006
5 Peter D. Turney, 2006

dei vicini simili a destra o a sinistra, allora sono collegate in modo paradigmatico. I vicini sintagmatici rappresentano spesso diverse parti del discorso, mentre i vicini paradigmatici sono di solito la stessa parte del discorso".

Negli ultimi anni i corpora hanno svolto un ruolo fondamentale per il progresso della semantica distribuzionale, in quanto rappresentano la fonte primaria di informazione per individuare le caratteristiche distribuzionali della parola; infatti la disponibilità attuale di grandi collezioni di testi e lo sviluppo e miglioramento di tecniche linguistiche computazionali sofisticate per estrarre schemi distribuzionali dei lessemi ha portato l'ipotesi distribuzionale alla realizzazione di modelli computazionali in grado di costruire spazi semantico-lessicali che sono stati applicati alla simulazione di aspetti diversi della competenza semantica.

In quanto attualmente la semantica distribuzionale è basata essenzialmente sui corpora, questo impedisce di poter modellare aspetti del significato intrinsecamente legati alle nostre esperienze extralinguistiche:

"Gli esseri umani non apprendono l'utilizzo di una parola grazie alla lettura della rispettiva definizione in un vocabolario ma, più facilmente, mediante l'osservazione di come questa viene usata in situazioni contingenti".[6]

Ciò nonostante questi modelli, anche se non in grado di proporre una teoria esaustiva del significato, sono potenti strumenti di controllo per testare, analizzare, confermare o falsificare teorie cognitive e del significato, attraverso analisi statistiche rigorose, estensive e approfondite delle produzioni verbali.

1.2 Modelli Semantici Distribuzionali

Una delle nuove tecnologie semantiche in grado di dare un ulteriore spinta a quel processo che vede i computer avvicinarsi sempre più alla comprensione del

6 Miller & Charles, 1991.

linguaggio umano sono i “Word Space Models”, o i cosiddetti “Modelli semantici distribuzionali”.

I WSM stanno riscuotendo un discreto successo all’interno della comunità scientifica, per la qualità dei risultati raggiunti e la plausibilità dei principi teorici su cui si basano. L’implementazione di questi modelli, che appaiono come spazi semantici di tipo distribuzionale, si fonda su una metafora concettuale, espressa da Sahlgren (2006) nei seguenti termini:

“Meanings are locations in a semantic space, and semantic similarity is proximity between the locations.[7]*”*

I modelli ottenuti mostrano dunque spazi semantici, dove parole simili sono rappresentate come vicini, grazie a una metafora concettuale esprimibile nei termini “*similarity-is-proximity*”. La somiglianza semantica tra significati si ottiene grazie al paragone tra i contesti di occorrenza delle parole, in quanto parole con simili proprietà distribuzionali e dunque con simili contesti di occorrenza e simili argomenti, presenterebbero anche forti legami semantici[8].

Come sottolinea Lenci (2009), grazie all'ipotesi distribuzionale si individuano proprietà paradigmatiche tra le parole, analizzando la loro occorrenza in contesto, sul piano sintagmatico. Sul piano cognitivo, questo corrisponderebbe a un modello del lessico mentale in cui i significati non sono organizzati come le definizioni dei sensi di un dizionario, ma piuttosto come rappresentazioni contestuali, del tutto dipendenti dal tipo di contesto che si prende in considerazione.

Questa ipotesi, come accennato in precedenza, è la base di partenza dei modelli semantici distribuzionali applicabili utilizzando algoritmi concreti per misurare il grado di similarità delle parole, i quali si servono di vettori, matrici e tensori di ordine superiore. Una parola, ad esempio, può essere rappresentata da un vettore nel quale gli elementi sono estratti da occorrenze della parola stessa in vari contesti

7 Traduzione: I significati son posizione in uno spazio semantico, e la similarità semantica è la vicinanza tra le posizioni.

8 Lenci a questo proposito si sofferma sull’ipotesi che entrambe le direzioni dell’affermazione siano valide. In particolare, il fatto che distribuzioni simili possano generare similarità semantiche tra due parole, potrebbe essere la spiegazione per l’uso di metafore, analogie e sensi figurati, fenomeni inerentemente cognitivi e diffusissimi nell’uso della lingua.

linguistici; infatti questi modelli sono conosciuti anche come “Vector Space Models”. Vi sono vari modelli di spazi semantici vettoriali che si fondano sulla stessa idea di base ma che differiscono gli uni dagli altri sia per l'implementazione di algoritmi differenti sia per le diverse finalità teoriche o applicative che fanno da riferimento a ciascun modello. I più noti sono :

- **LSA** (*Latent Semantic Analysis*), che, tramite ciascuna dimensione di un vettore, registra le occorrenze di una parola *w* in un documento, il quale rappresenta uno specifico contesto. Inoltre è stato dimostrato sperimentalmente che è in grado di identificare casi di sinonimia delle parole ottenendo prestazioni comparabili ai soggetti umani[9].
- **HAL** (*Hyperspace Analogue to Language*) che ritiene che il contesto di una parola sia costituito solamente dalle parole che lo circondano immediatamente;
- **RI** (*Random Indexing*) che, come alternativa alla LSA, è un processo composto da due fasi: la prima è l'assegnazione di ogni contesto, parola o documento nei dati ad un’unica rappresentazione chiamata indice vettoriale; successivamente, dalla scansione del testo vengono prodotti i vettori contesto e ogni volta che una parola occorre in un contesto, tale indice vettoriale del contesto si aggiunge al vettore contesto per la parola in questione.

Dagli studi effettuati si è evinto che questi modelli distribuzionali risultano avere buone prestazioni quando utilizzati per compiti che misurano la somiglianza di significato tra le parole, tra le frasi e tra i documenti.
Di fatto, la maggior parte dei motori di ricerca utilizza i WSM proprio per misurare la somiglianza di una query con un documento.
Inoltre, i modelli semantici distribuzionali sono stati applicati anche all'acquisizione lessicale; e cioè simulando l'espansione del vocabolario da parte del bambino attraverso un processo di induzione del contenuto semantico delle parole da statistiche di co-occorrenza nell'input dell'adulto.

9 Il test standard appropriato per questi esperimenti è il TOEFL (*Test of English as a Foreign Language*) che comprende 80 parole delle quali si deve individuare il sinonimo che più si adegua tra quattro possibili alternative.

Un esempio di quanto appena detto è il word space model presentato nel 2007 da Baroni *et al.* costruito a partire da un campione dell'input linguistico a cui è esposta Lara, una bambina inglese tra i due e i tre anni di età. Nello spazio semantico definito dal modello si è potuto notare come nomi di umani (es. *nonno, madre, padre*) e nomi di animali (es. *cane, gatto, giraffa*) vengono a confondersi gli uni con gli altri comparendo in posizioni estremamente ravvicinate. Un dettagliata analisi del corpus usato per costruire lo spazio ha rivelato il perché queste due categorie distinte vengono invece riconosciute come un'unica classe semantica che le comprende entrambe. I nomi di animali, infatti, non si riferiscono ad animali reali ma a nomi di personaggi di favole, cartoni animati e giocattoli i quali, come é tipico del linguaggio infantile, vengono umanizzati, ovvero caratterizzati da comportamenti di tipo umano. Per questo vi é una forte corrispondenza nell'input a cui sono esposti i bambini (favole, racconti e giochi con pupazzi di animali), ed ancora per questo motivo troviamo un'elevata similarità semantica tra nomi di categorie diverse.
Ad ogni modo i modelli WSM offrono uno strumento potente per l'analisi dei contesti di occorrenza delle parole, che può contribuire a far luce sui meccanismi cognitivi che caratterizzano l'apprendimento e il reperimento di informazioni semantiche dai contesti linguistici.

1.2.1 Costruzione di Spazi Semantici

Il contenuto semantico di una parola, come già detto, è rappresentato dalla sua posizione in uno spazio; questo, inteso come spazio semantico, è definito da un sistema di coordinate, il quale è determinato dai contesti linguistici in cui la parola può ricorrere: nello specifico dalla quadrupla $<T, B, M, S>$[10], dove *T* sta per *target,* ed è l'insieme delle parole che formano gli elementi nello spazio; *B* è la base che indica le dimensioni dello spazio; *M* è una matrice di co-occorrenza che fornisce una rappresentazione vettoriale di ogni parola in *T*; ed *S* è la metrica che misura la distanza tra i punti nello spazio.

10 Lowe, 2001; Padó e Lapata, 2007.

Secondo questo schema, le parole vengono proiettate in questo spazio metrico e ordinate secondo una certa distanza dipendente dal loro grado di similarità semantica, la quale, seguendo la regola dell'ipotesi distribuzionale *"due parole sono tanto più semanticamente simili quanto più tendono a ricorrere in contesti linguistici simili"*[11] viene calcolata grazie alle distribuzioni statistiche di co-occorrenza nei testi.

1.2.2 Matrici e Vettori

Un importante passo in avanti dei WSM è l'utilizzo delle matrici costruite secondo la "Bag of Words Hypothesis"[12]. Ad esempio, se si vuole ricercare la similarità semantica tra interi documenti si utilizzerà una matrice *termine-documento.*
Questo tipo di matrice è formata da tante righe quante sono le parole (o termini) e da tante colonne quanti sono i documenti (es.: pagine web). Un vettore documento conterrà un elemento per ogni parola tipo che il documento contiene (cosi da evitare ripetizioni di una medesima parola) ed ogni elemento corrisponderà al numero di occorrenze di una determinata parola tipo contenuta nel documento stesso.

	Doc A	*Doc B*	*Doc C*	*Doc D*
Cane	1	5	68	44
Gatto	0	2	59	37
Guidare	13	89	2	2
Fumare	66	26	1	1

Tabella 1: Matrice *termine-documento*.

Pertanto, se due documenti trattano argomenti simili, allora i due vettori corrispondenti tendono ad avere valori simili. Ad esempio osserviamo, nella tabella qui in alto, come le parole *Cane* e *Gatto* hanno simili valori per quanto riguarda i

11 Miller & Charles, 1991.
12 Salton et al., 1975: *Bag of words hypotesis*.

documenti *C* e *D* perché notiamo che ricorrono più volte. Di conseguenza questo ci porta a pensare che Cane e Gatto siano distribuzionalmente e dunque semanticamente simili perché ricorrono negli stessi documenti.

Un approccio diverso è quello usato dalla matrice *parola-contesto.* Questo tipo di matrice registra le co-occorrenze di una parola target (vettori-riga) e una parola facente parte della sua finestra di contesto (vettori-colonna). Quando le co-occorrenze si verificano frequentemente vengono chiamate *collocazioni* e l'elemento che co-occorre nel formare una collocazione è chiamato collocato. Di seguito una tabella di esempio di quest'ultima matrice:

	Computer	*Giocare*	*Armadio*	*Scrivere*
Cane	4	7	2	0
Fumare	0	1	0	4
Quaderno	12	4	0	20
Gatto	0	1	2	1

Tabella 2: Matrice *parola-contesto*.

Per concludere, un miglioramento dell'uso delle matrici ci è stato dato nel 2001 dagli studi di Lin e Pantel con l'*ipotesi Distribuzionale Estesa.* Questa ipotesi si avvale dell'utilizzo dei vettori-riga come corrispondenti a coppie di parole e dei vettori-colonna corrispondenti a contesti in cui le coppie di parole co-occorrono. Questo tipo di matrice è detta matrice *coppia-pattern* ed è stata introdotta alla scopo di misurare il grado di similarità semantica dei contesti, ovvero dei vettori-colonna. Al contrario, l'*ipotesi di Relazione Latente* di Turney *et al*. (2003) utilizza la stessa tipologia di matrice ma sposta l'attenzione sui vettori-riga. Quindi se coppie di parole ricorrono in contesti simili tendono ad avere simili relazioni semantiche.

La matrice che riassume entrambe le ipotesi può essere rappresentata in questo modo:

	Contesto 1	*Contesto 2*	*Contesto 3*	*Contesto 4*
Coppia 1	6	3	1	0
Coppia 2	4	1	2	3
Coppia 3	1	5	3	1
Coppia n	*n*	*n*	*n*	*n*

Tabella 3

1.3 Distributional Memory

Questo modello è composto da una *memoria semantica distribuzionale* (DM) ricavata da un corpus e si propone, sulla base degli studi di Alessandro Lenci e Marco Baroni, come un approccio differente alla semantica basata sui corpus.
Le funzionalità principali del modello in questione sono generalmente due:

- la possibilità di costruire spazi semantici a partire da matrici di co-occorrenza definite selezionando diverse unità concettuali;
- la possibilità di misurare il grado di similarità nella matrice risultante tra specifiche righe in cui gli elementi che le costituiscono condividono particolari proprietà.

Tra i modelli WSM, Distributional Memory si distingue per la sua portabilità, cioè per la sua capacità di adattarsi a svolgere diversi compiti. Quest'ultimi vengono svolti attraverso un procedimento che si basa sull'estrazione di triplette di parole, chiamate tuple, strutturate nel seguente modo: una parola, un elemento di connessione e un'altra parola, cioè Word, Link, Word (W1-link-W2). Nell'esempio seguente, sono riportate alcune tuple del nome *e-books*:

W1	*Link*	*W2*	*LMI*
e-books-n	about	debate-n	6.8417
e-books-n	about	do-v	4.9422
e-books-n	about	feel-v	6.48
e-books-n	about	information-n	5.58
e-books-n	about	know-v	5.4722
e-books-n	about	part-n	5.1990
e-books-n	about	talk-v	6.9317
e-books-n	about	thing-n	5.7047
e-books-n	about	topic-n	6.8398
e-books-n	about	wonder-v	8.9829
e-books-n	about	business-n	5.8122
e-books-n	around	book-n	6.79
e-books-n	as	couple-n	5.52
e-books-n	as	date-v	6.58
e-books-n	as	define-v	5.36

Tabella 4: Esempio di alcune tuple con il rispettivo peso.

Le tuple estratte sono poi ‘pesate’, cioè associate ad un valore (riportato sulla quarta colonna) che indica una misura di associazione tra gli elementi costituenti. Questa misura di associazione è la Local Mutual Information (una versione modificata della Mutual Information[13]). Qui di seguito la formula:

$$LMI = f(< t_i, t_j >) \cdot \mathrm{LOG}_2 \frac{p(t_i,t_j)}{p(t_i)p(t_j)}$$

Seguendo questo schema base, Distributional Memory ha implementato tre differenti modelli in base ai differenti modi di utilizzo della struttura della tupla, nello specifico

13 Mutual Information: unità di misura, la quale permette di calcolare il rapporto logaritmico tra la frequenza osservata e la frequenza attesa di una determinata co-occorrenza, attraverso una formula in cui si esprime il rapporto (logaritmico) tra la frequenza della co-occorrenza dei due elementi e la frequenza delle singole occorrenze dei due elementi.

ciò che cambia è la complessità del *link*. Questi modelli sono stati creati grazie alla concatenazione dei corpus *Web-derived ukWaC* (circa 1.915 miliardi di tokens), *English Wikipedia* (820 milioni di tokens scaricati a metà 2009) e il *British National Corpus* (circa 95 milioni di tokens), i quali, in seguito ad un processo di tokenizzazione e lemmatizzazione, hanno dato vita ad un unico corpus delle dimensioni di 2.83 miliardi di tokens.

Il primo modello realizzato prende il nome di ***DepDM*** e si basa principalmente sulla classica intuizione che i legami sintattici rappresentano una buona approssimazione delle relazioni semantiche tra le parole. Questo è il modello con il minor grado di lessicalizzazione dei link e ne contiene 796 tipi diversi per un totale di 110 milioni di tuple.

Il secondo modello, che contiene 3,352,148 links (includendo gli inversi), ha il nome di ***LexDM*** e pone grande attenzione sul materiale lessicale che collega due parole. Infatti quest'ultimo risulta essere molto informativo riguardo la relazione sintagmatica tra le parole prese in analisi.

Infine, il terzo e ultimo modello è ***TypeDM***, testato da Baroni *et al* nel 2010, e si fonda sull'idea che ancora più importante della frequenza di un link è la varietà delle forme con cui questo viene espresso. Questo modello contiene 25,336 tipi di link per un totale di 130 milioni di tuple; insomma rappresenta una sorta di livello medio tra gli altri due modelli. Di seguito alcuni esempi di tuple dei tre modelli:

DepDM:

Link: Sub-tr, *soggetto di un verbo che occorre con un oggetto diretto*:

Il professore sta leggendo un libro.

Tupla: {professore, *sub-tr*, leggere};

Link: Preposition, *preposizione*:

Ho visto il calciatore colpirlo con la palla.

Tupla*:* {palla, *con*, calciatore};

LexDM:

Link: Verb, *il verbo collega soggetto con complemento; se il verbo è presente nella lista dei 52 verbi più frequenti allora, nella creazione della tupla, il generico link "verb" è sostituito dallo stesso verbo*:
Il soldato usa una pistola.
Tupla: {soldato, *usare*, pistola} o {soldato, *verb*, pistola};

Link: Is, *copula con aggettivo*:
Francesco è alto .
Tupla: {alto, *is*, Francesco};

TypeDM:
Link: Obj, *oggetto diretto*:
Il fruttivendolo sta mangiando una mela.
Tupla: {fruttivendolo, *obj*, mangiare};

Link: Coord, *due nomi coordinati*:
Calciatore e ingegnere.
Tupla: {insegnante, *coord*, soldato};

Distributional Memory, comunque, utilizzando delle tuple pesate dalle quali è stato possibile estrarre una serie di spazi semantici e spiegare vari elementi di ricerca semantica basata sui corpus, risulta essere un potente strumento di base per ulteriori ricerche semantiche quali la categorizzazione di unità concettuali, le analogie relazionali tra coppie di unità concettuali e la misurazione della similarità semantica grazie agli insiemi di vettori in spazi vettoriali diversi che la stessa DM fornisce.

1.3.1 Vicini Paradigmatici: il coseno

Fin dalla nascita dell'ipotesi distribuzionale, uno degli obiettivi principali dei modelli che ne derivano è stato comprendere il grado di similarità semantica tra parole.
Quest'ultima può essere definita e misurata attraverso la proiezione delle parole stesse in uno spazio attraverso dei vettori *n*-dimensionali.
Una delle unità di misura più comunemente usate per determinare la distanza spaziale tra due termini lessicali, o per meglio dire, tra due vettori è il coseno: e cioè l'angolo che si viene a formare tra due vettori in esame.[14]
In alternativa, la distanza tra due vettori può essere misurata utilizzando la classica *metrica euclidea*, generalizzata al caso dello spazio *n*-dimensionale.[15]
Se i vettori sono normalizzati, il coseno produce un ordinamento di similarità equivalente a quello stabilito calcolando la distanza euclidea: in altri termini, se vogliamo sapere quale tra due parole *w1 e w2* siano più vicine a una terza parola *w3*, la distanza euclidea e il coseno ci forniscono la medesima risposta.
Nei "*word spaces models*" il significato di una parola è totalmente e unicamente definito dalla sua posizione all'interno dello spazio multidimensionale determinato dalla base contestuale.
Le dimensioni dei vettori delle parole, di per sé, non sono direttamente interpretabili: servono solo a registrare la posizione delle parole, stabilita dai rapporti sintagmatici nei contesti linguistici, e determinare la distanza nello spazio di queste con le altre potendo, appunto, calcolarne il coseno. Quindi, quest'ultimo ci dice semplicemente che due parole vicine nello spazio vettoriale hanno simili distribuzioni statistiche nei contesti linguistici. Il significato nasce solo dalle configurazioni di punti nello spazio, collocati secondo rapporti proporzionali al loro grado di similarità distribuzionale.

14 Se due vettori sono normalizzati, il loro coseno è equivalente alla somma dei prodotti delle rispettive dimensioni. Un vettore si dice *normalizzato* se la sua lunghezza è uguale a 1. :a *lunghezza* (o *norma*) di un vettore è uguale alla radice quadrata della somma dei quadrati delle sue dimensioni per la norma del vettore.
15 La *distanza euclidea* tra due vettori è uguale alla radice quadrata della somma dei quadrati delle differenze delle loro dimensioni

Siano *x* e *y* due vettori, ognuno con *n* elementi:

$$\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle$$
$$\mathbf{y} = \langle y_1, y_2, \ldots, y_n \rangle$$

Il coseno dell'angolo θ tra *x* ed *y* può essere calcolato come segue:

$$\begin{aligned} \cos(\mathbf{x}, \mathbf{y}) &= \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \cdot \sum_{i=1}^{n} y_i^2}} \\ &= \frac{\mathbf{x} \cdot \mathbf{y}}{\sqrt{\mathbf{x} \cdot \mathbf{x}} \cdot \sqrt{\mathbf{y} \cdot \mathbf{y}}} \\ &= \frac{\mathbf{x}}{\|\mathbf{x}\|} \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|} \end{aligned}$$

In altre parole, il coseno dell'angolo tra due vettori è il prodotto interno dei vettori, dopo che sono state normalizzate per unità di lunghezza. Se x e y sono vettori di frequenza per le parole, una parola frequente avrà un vettore di lunghezza maggiore e una parola rara avrà un vettore di lunghezza minore, ma considerando il fatto che comunque le parole potrebbero essere sinonimi, la lunghezza dei vettori viene ad essere irrilevante; ciò significa che quel che importa maggiormente è l'angolo che si viene a formare tra questi.

Questa unità di misura per calcolare la distanza semantica tra due termini varia da 0 (-1) a 1. Ciò equivale a dire che: *se due vettori sono geometricamente allineati sulla stessa linea e nella stessa direzione, l'angolo che si formerà tra loro sarà pari a 0° ed il loro coseno pari ad 1, quindi il grado di similarità sarà massimo; al contrario, se due vettori sono indipendenti l'uno dall'altro puntando in direzioni opposte* (ortogonali)*, il loro angolo sarà pari a 90° e il coseno sarà uguale a 0, quindi assenza di similarità.*

Word	*Neighbour*	*Cosine*
superstate-n	battleground-n	0.36
superstate-n	republic-n	0.35
superstate-n	powerhouse-n	0.35
superstate-n	superstar-n	0.34
superstate-n	commonplace-n	0.34
superstate-n	favorite-n	0.33
superstate-n	best-seller-*n*	0.32
indepth-n	detailed-j	0.38
indepth-n	cursory-j	0.31
indepth-n	incisive-j	0.31
indepth-n	book-lenght-n	0.28
indepth-n	qualitative-j	0.28
indepth-n	first-hand-j	0.27

Tabella 5: coseni tra parole in uno spazio distribuzionale

Visto e considerato che le parole più simili dal punto di vista semantico hanno effettivamente un coseno più elevato, l"ipotesi distribuzionale, la quale costituisce il fondamento dell'analogia SIGNIFICATO = SPAZIO DI PAROLE, trova cosi un effettivo riscontro nelle rappresentazioni computazionali costruite da DM.

2 – Visualizzazione dell’informazione

2.1 Introduzione.

Lo scopo principale della rappresentazione grafica è quello di fornire un aiuto visivo per poter riflettere e discutere determinati problemi di carattere statistico.
Rappresentare una certa quantità di dati graficamente significa avere la possibilità di cogliere immediatamente le caratteristiche essenziali di una distribuzione e confrontare i dati stessi tra di loro.
Esiste una grande varietà di rappresentazioni grafiche: i grafici più semplici, più efficaci e comunemente utilizzati sono: i diagrammi a barre verticali o orizzontali, i grafici a settori circolari, gli istogrammi, i grafici a punti e i grafi ad albero.
La scelta tra le molteplici rappresentazioni grafiche esistenti dipende essenzialmente dalla natura del fenomeno che si vuole rappresentare, dal tipo di carattere che descrive il fenomeno e dal numero di caratteri coinvolti nel fenomeno (ad esempio se si tratta di distribuzioni semplici o multiple).
E’ ovvio che l’uso di tali mezzi è a favore degli scopi del lavoro statistico, che consistono nella presentazione dei dati con una chiarezza che riduca i dubbi e le cattive interpretazioni al minimo.

2.2 Javascript Infovis Toolkit

La *Javascript Infovis Toolkit* (JIT) è una raccolta di grafi e grafici interattivi scritta in *JavaScript* per facilitare lo sviluppo di applicazioni di visualizzazione delle informazioni.
Quando fu ideata da Nicolas Garcia Belmonte questa raccolta conosciuta in versione 0.9 beta implementava già nove tipi di visualizzazione. Negli ultimi anni il toolkit è divenuto molto popolare sia nel settore industriale che nel settore della ricerca; nel

novembre 2010 è stato acquistato dalla Fondazione Labs Sencha estendendosi ulteriormente e arrivando a coinvolgere il supporto di animazioni CSS3 e molti altri tipi di visualizzazioni; e nel 2011 è stato scelto come guida per l'organizzazione e progettazione per il Google Summer of Code.
Attualmente la libreria JIT sta acquistando grande popolarità riuscendo ad offrire maggiori prestazioni senza perdere la sua flessibilità ed è per questo motivo che è riconosciuta come un mezzo importante per l'esplorazione e l'analisi di dati.

Le sue caratteristiche principali sono:

- Strutture di dati generici adatti alla visualizzazione.
- Algoritmi specifici per visualizzare particolari strutture di dati.
- Meccanismi e componenti per eseguire la manipolazione diretta delle visualizzazioni dei dati.
- Componenti per eseguire l'etichettatura e la deformazione spaziale.

Il toolkit supporta principalmente due strutture di dati: tabelle e grafici ad alberi; inoltre fornisce strumenti per la creazione di visualizzazioni di dati interattive per il web come TreeMaps, (alberi della tipologia SpaceTree), matrici di adiacenza e diagrammi *Node-Link* (i quali forniscono visualizzazioni in diverse varianti, 8 per i grafici e 4 per gli alberi), una disposizione radiale di alberi con animazioni chiamata *R-Graph*, e altre tipologie di visualizzazioni più comuni come grafici a barre, grafici a torta e grafici ad aerea, comprende anche funzioni avanzate che permettono di effettuare query dinamiche ottenendo una veloce combinazione azione-risultato.
A sua volta ogni visualizzazione include un elenco di attributi visivi che possono essere associati in colonne, le quali gestiscono delle righe di elementi come stringhe, numeri interi o decimali. Le colonne hanno un ruolo fondamentale nella formazione della struttura dei dati in quanto sono implementate con array primitivi. Ciò significa che si può memorizzare in modo rapido ed efficiente un grande insieme di dati sotto forma di colonne, le quali vengono organizzate in una tabella formando cosi una struttura tipo, che viene utilizzata per visualizzazioni di alberi e grafi. La rappresentazione ad albero consiste nell'aggiungere una colonna “genitore”, una

colonna “figlia” , una colonna “sorella” e una colonna “sottofiglia” se presente. Un altra caratteristica interessante sono i visualizzatori. Questi contengono una lista di attributi visuali, come il colore, la dimensione, l'etichetta e la trasparenza, che possono essere associati con le colonne; in più hanno il compito di eseguire operazioni quali il filtraggio, lo zoom, la navigazione e il raggruppamento. *InfoVis toolkit,* inoltre, fornisce dei componenti che consentono di manipolare interattivamente questi visualizzatori tramite un pannello di controllo affiancato ad ogni visualizzatore organizzato a schede che ne consentono la manipolazione o la configurazione.
Questa libreria composta da circa 30.000 righe di Java comunque sta subendo ulteriori miglioramenti soprattutto per quanto riguarda il sistema Agile2D (un framework per la sperimentazione di alternative Java2D) così da poter offrire nuove astrazioni sempre più innovative ed efficienti.

2.2.1 R-Graph

R-Graph è una sotto raccolta di grafi facente parte della ormai grande famiglia del Toolkit Infovis, che si basa sulla visualizzazione ad albero radiale (*radial-tree*). Tale rappresentazione dei dati tratta il grafo come un albero radicato nel nodo di messa a fuoco, dal quale si diramano una serie di nodi considerati come figli del nodo centrale, che funge quindi da padre. Una delle particolarità che rende tale grafico efficiente e funzionale per la distribuzione dei dati a livello visuale è che il numero di nodi non ha un limite, ma, al contrario, ha la possibilità di espandersi fino a poter visualizzare una quantità di dati praticamente infinita.

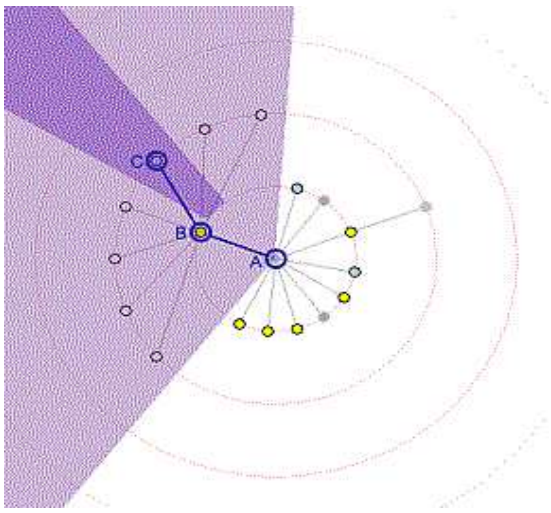

Figura 1: Illustrazione della tecnica di layout radiale.

Detto questo è sottinteso che si tratta di una rappresentazione grafica basata principalmente sui rapporti di parentela tra gli elementi e che quindi i figli possono a sua volta fungere la funzione di padre per altri elementi con la possibilità di prendere la posizione del nodo centrale grazie soprattutto ad un animazione di spostamento e organizzazione ottimizzata degli elementi; questa animazione fa sì che tutti gli elementi presenti si dispongano in una posizione diversa da quella iniziale ma mantenendo viva la gerarchia formatasi dalla prima ricerca. L'effetto di spostamento crea una nuova vista dell'intero grafo in modo graduale; infatti si è preferito scegliere un effetto di tipo *slow-in, slow-out,* ovvero un effetto la cui velocità di esecuzione, sia all'inizio che alla fine dell'animazione ha un accelerazione bassa e che solo nel momento centrale della stessa esecuzione aumenta leggermente; in questa maniera, chi naviga il grafo può spostarsi facilmente da un nodo ad un altro senza perdere l'orientamento all'interno dello spazio creato.

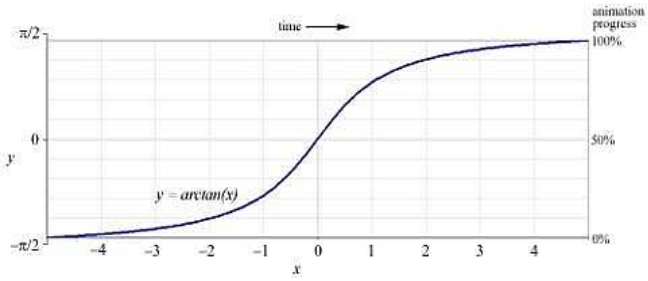

Figura 2: Animazione temporale *slow-in, slow-out*.

I nodi che rimangono su un determinato anello scivolano sulla sua circonferenza, mentre i nodi che cambiano anello si spostano in maniera uniforme con un movimento a spirale da un anello all'altro.
Tutti i nodi, meno il focus, sono distribuiti nel grafo e disposti secondo cerchi concentrici, e la posizione angolare di ogni nodo su un anello corrispondente è determinata sia dalla vicinanza del suddetto con il nodo centrale, sia dal numero di nodi presenti sull'anello.

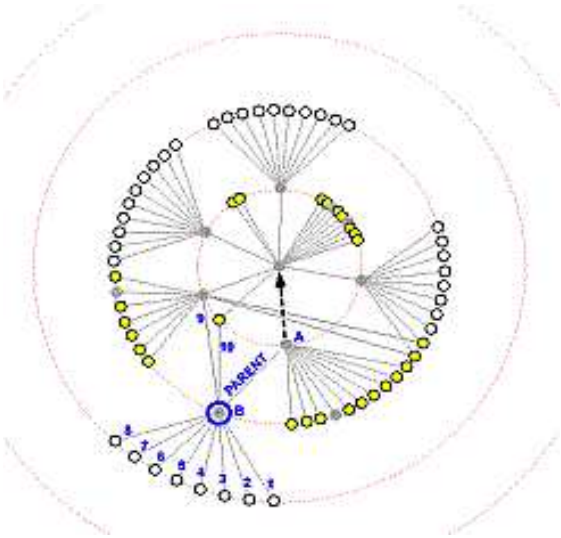

Figura 3: Spostamento dei nodi all'interno dello spazio creato.

Ciò significa che lo spazio presente su ogni anello risulta essere suddiviso in base al numero dei nodi che vi risiedono creando un angolo identico tra loro e il nodo centrale. I nodi che sono posizionati sugli anelli più esterni,
rispetto a quello centrale, invece si predispongono su una precisa porzione dell'anello che corrisponde alla proiezione immaginaria dell'angolo del nodo *genitore* rispetto al nodo centrale.
Concludendo, *R-Graph* si basa su una struttura *JSON (JavaScript Object Notation)* che presenta una serie di proprietà, quali *id, name, data, relation, children ed adjacent.*

Queste proprietà disposte in un determinato ordine formano i nodi veri e propri che verrano visualizzati nell'interfaccia grafica.
Solitamente la disposizione di quest'ultime è la seguente:

```
var json = [
{
        “id”:  “aUniqueIdentifier”,
        “name”:  “usually a nodes name”,
        “data”:  [
                {key: ”some key”, relation: “some relation”, value:  “some value”},
        children: “id”:  “aUniqueIdentifier”,
                “name”:  “usually a nodes name”,
                “data”:  [
                        {key: ”some key”, relation: “some relation”, value:  “some value”},
        children: “id”:  “aUniqueIdentifier”,
                “name”:  “usually a nodes name”,
                “data”:  [
                        {key: ”some key”, relation: “some relation”, value:  “some value”},
                        {key: ”some key”, relation: “some relation”, value:  “some value”},
        ],
                                } /* … more nodes here … */ ]
```

dove:

- l'**id** rappresenta un identificatore univoco del nodo;
- **name** è il nome del nodo che apparirà nel grafo;
- **data** conterrà una serie di informazioni aggiuntive non obbligatoriamente necessarie;
- **children** conterrà a sua volta le proprietà di base dei comuni nodi, cosi da poter creare molteplici generazioni di sotto-figli;
- **adjacent** infine, si utilizza nel caso in cui si vogliano creare più di tre generazioni di sotto-figli; utilizzando questa proprietà viene a cambiare, quasi totalmente, la struttura del grafo a livello di codice modificandone la disposizione delle altre proprietà e aggiungendone

altre come **nodeFrom** e **nodeTo**; in questo caso la struttura prenderà la seguente disposizione:

```
var json = [
{
        “id”:  “aUniqueIdentifier”,
        “name”:  “usually a nodes name”,
        “data”:
                { key: ”some key”, relation: “some relation”, value:  “some value”},
        "adjacencies": [{
                        "nodeFrom": "node name",
                        "nodeTo": "node name",
                        "data": {
                        "weight": 3
                }
},

                {
                        "nodeFrom": "node name",
                        "nodeTo": "node name",
                        "data": {
                        "weight": 3
                }
},
                “id”:  “aUniqueIdentifier”,
                “name”:  “usually a nodes name”,
                “data”:  [
                        { key: ”some key”, relation: “some relation”, value:  “some value”},
                "adjacencies": [{
                        "nodeFrom": "node name",
                        "nodeTo": "node name",
                        "data": {
                        "weight": 3
                }
},

} /* … more nodes here … */ ;
```

In altre parole, definito un nodo madre, i nodi abbracciati dall'attributo *adjacent*, posto immediatamente di seguito, saranno riconosciuti come figli di quello e verranno posti nel grafo uno accanto all'altro sullo stesso anello.

Queste strutture, necessarie per il corretto funzionamento dell'*albero radiale*, possono essere implementate attraverso la creazione di un array statico o dinamico generato attraverso delle query.

R-Graph comprende anche un apposito foglio di stile CSS da cui sarà possibile personalizzare tutti gli aspetti grafici a piacimento dell'utente.

3 – Visualizzatore di spazi semantici

3.1 Introduzione.

Dopo la presentazione nei capitoli precedenti dei modelli e dei concetti chiave che formano parte della semantica distribuzionale, in questa parte della relazione si esporranno passo per passo le tappe che hanno visto la realizzazione dell'interfaccia grafica di spazi semantici, includendo una descrizione di come è stato possibile interagire con il database, con il grafo *R-Graph* e la sua struttura attraverso il linguaggio di programmazione php che ha permesso la manipolazioni dei dati.

3.2 Il database: paradigmatic neighbours

Paradigmatic neighbours, la lista di neighbours utilizzata per la creazione del database, è stata costruita da Partha Pratim Talukdar, uno studioso ampiamente interessato all'apprendimento automatico, all'elaborazione del linguaggio naturale, all'integrazione dei dati e alla scienza cognitiva.
Il pacchetto che ha permesso la sua realizzazione si chiama *S-Space*, una serie di strumenti ed un software per la creazione di spazi semantici che applicano algoritmi semantici spaziali per prendere le regolarità statistiche delle parole in un corpus e mappare ogni parola ad un vettore alto-dimensionale che rappresenta la semantica. Questo pacchetto è stato utilizzato per computare i coseni tra i vettori parola, i quali sono formati da dati distribuzionali costituiti dal tensore TypeDM.
Il database, gestito tramite il *tool open source* PhpMyAdmin, contiene 306.860 linee di record. Ogni record consiste in una coppia di unità lessicali, tra cui nomi, verbi e aggettivi contenute nei campi *word* e *neighbour*, corrispondenti rispettivamente all'unità focus e al suo vicino; entrambi seguiti da un campo (*suffix* e *suffix_n)* che ne indica il tipo di unità lessicale. Infine è presente un campo *cosine* che, come

suggerisce la parola, indica il coseno dell'angolo tra *word* e *neighbour* ed un campo *id* per identificare univocamente l'intero record. Di seguito un frammento del database raffigura quanto appena detto.

word	suffix	neighbour	suffix_n	cosine
eye-catching	j	glossy	j	0.34578288719851946
eye-catching	j	fabulous	j	0.34624398913282606
eye-catching	j	promotional	j	0.3601095961152367
eye-catching	j	iconic	j	0.36329559803923056
eye-catching	j	colorful	j	0.36703824425518355
eye-catching	j	stylish	j	0.3671448475228121
eye-catching	j	striking	j	0.37426771404313836
eye-catching	j	garish	j	0.37969172734994716
eye-catching	j	stunning	j	0.3833047500933139
eye-catching	j	colourful	j	0.43354311185101163
no-fault	n	indiana	n	0.2305953608222968
no-fault	n	churchill	n	0.23731136613935386
no-fault	n	ohio	n	0.23866687609083748
no-fault	n	massachusetts	n	0.2521522699830262
no-fault	n	arizona	n	0.25247731449277006
no-fault	n	anti-terror	n	0.2649170623397245
no-fault	n	illinois	n	0.26576706005978673
no-fault	n	agri-environment	n	0.27006983566205767
no-fault	n	florida	n	0.2804667036315603
no-fault	n	hatred	j	0.30675463664223046
spin-off	n	special	n	0.363504247079579
spin-off	n	sitcom	n	0.36484289576996864
spin-off	n	prequel	n	0.3660431802155663
spin-off	n	miniseries	n	0.3755822277073014
spin-off	n	series	n	0.3806196883692266
spin-off	n	adaptation	n	0.3807191310469776
spin-off	n	serial	n	0.3847335304346124

Figura 4: Frammento di Paradigmatic neighbours.

Questa coppia, quindi, mostra la vicinanza paradigmatica delle due parole in questione, cioè la capacita che hanno queste di apparire, con un valore semantico molto vicino, nei medesimi contesti.

3.2.1 Query al database

Per far interagire i dati del database *Paradigmatic neighbours* con la libreria *Infovis Toolkit* ed in particolare con il grafo *Rgraph*, è stato necessario elaborare una query che consente di raccogliere i dati in base alle richieste dell'utente per poterli poi organizzare secondo la struttura dell'albero radiale proprio di *Rgraph.*
Infatti, grazie all'interazione del linguaggio di programmazione Php con il gestore MySQL, l'utente ha la possibilità di immettere una parola attraverso l'apposita form cosi da avviare una ricerca della stessa nel database la quale avrà come risultato tutte le parole contenute nel campo *neighbour* e, contemporaneamente, associate al campo *word* ricercato.

```
$query= 'SELECT * FROM paradigmatic WHERE word LIKE "'.$_GET['parola'].'"
ORDER BY cosine DESC LIMIT 10';
```

Nel caso in cui la parola fosse presente nel database, automaticamente si avvierà una seconda ricerca per dar vita agli array dei sotto-figli dove le parole da ricercare saranno i risultati (le parole presenti nel campo *neighbour)* della ricerca precedente.

```
$query= 'SELECT * FROM paradigmatic WHERE word LIKE neighbour
ORDER BY cosine DESC LIMIT 10';
```

In questo modo ciò che viene restituito è un array di array contenente tutti i dati ricavati dalla ricerca effettuata in precedenza e disposti secondo un ordine preciso, ovvero quello previsto da *Rgraph* sotto la proprietà *children* sopra illustrata.
Così facendo si può estendere il grafo a più generazioni con l'obiettivo di avere una visione chiara e completa della vicinanza tra le parole.

3.3 Struttura del visualizzatore

Una volta ricavato l'array madre si procede con l'assegnazione e l'organizzazione di questo dentro la variabile ***json*** che sarà poi passata al file ***Rgraph.js*** tramite l'apposita funzione ***json_encode*** che codifica i dati php. Questa funzione di codifica in realtà lavora solo con stringhe codificate UTF-8 ed UTF-16 ed internamente si riduce solo alla codifica UTF-8, il che è una limitazione; infatti, assegnandogli stringhe con altri tipi di *encoding* si corre il rischio di ottenere come risultato una stringa json valida ma incompleta. Per questo, nel nostro caso specifico, con l'utilizzo della codifica ISO-8859-1 da parte del linguaggio php, si è dovuto ricorrere ad un lavoro di *assegnazione* (simbolo di assegnazione: "=>") degli elementi da codificare (singoli record o interi array) agli elementi propri di json precedentemente elencati (*'id', 'name', 'children'*) :

```
$data = array('id'=>$record['id'], name'=>$record['word'],
'children'=>$array_children, 'data' => $relation_root);
```

Struttura dell'array madre tramite assegnazione degli elementi da codificare.

```
var json = <?php echo json_encode($data); ?>
```

Funzione per il passaggio dei dati a JSON.

A questo punto Rgraph.js ha il compito di elaborare la variabile, estrarre la struttura precedentemente ordinata secondo la sintassi prevista ed infine, tramite la funzione ***init***, caricare l'interfaccia grafica.

In quanto rappresenta uno dei file chiave per l'avvio dell'albero radiale, esso comprende tutte le funzioni riguardanti la forma in cui il grafo si presenta definendo le modalità di navigazione (es. *scroll zoom*), la posizione e lo stile degli archi, dei nodi e dei rami e l'animazione di caricamento iniziale e di spostamento degli elementi (*nodi*) all'interno dello spazio con la possibilità di modifiche da parte di chi ne usufruisce. Inoltre, grazie alla gestione diretta degli *id* di ogni elemento, è possibile anche implementare il codice con funzioni che permettono di interagire

direttamente con l'interfaccia (es. funzione di ricerca immediata di un elemento dopo il click dello stesso nel grafo: *function onClick*).
Ad ogni modo il file principale è *jit.js* che comprende la vera e propria libreria. Quest'ultimo è formato da 17270 righe di codice e si preoccupa del corretto funzionamento di tutti i grafici e grafi implementati dalla libreria includendo tutte le funzioni avanzate per la visualizzazione delle informazioni e le relative descrizioni.

3.4 Visualizzazione grafica dei dati

Dando avvio al server Apache ed al gestore MySQL si carica il database e i dati contenuti nei file js, php e css necessari per il corretto funzionamento dello strumento.
Una volta terminato il caricamento, è possibile accedere alla pagina dove l'interfaccia grafica è stata integrata.
La prima schermata, una volta entrati, è immediatamente quella relativa a paradigmatic neighbours. In questa sezione sono presenti vari box quali:

- un box centrale che rappresenta il grafo
- un minibox superiore che contiene informazioni relative allo stato della nostra interfaccia
- un box sul lato destro della pagina restituisce invece informazioni relative all'uso del visualizzatore e all'interpretazione dei dati.
- un box per effettuare la ricerca, infine, sulla barra che divide quanto appena descritto dall'header della pagina.

Come già accennato il visualizzatore consente di fare una ricerca tramite l'apposito box.

Figura 5: Box dal quale si effettuano le query;

Al momento del completamento della ricerca la pagina si aggiornerà popolando il box centrale di dati. A questo punto è possibile iniziare a navigare all'interno del grafo: infatti si ha la possibilità di spostarsi da un elemento ad un altro e con l'implementazione della funzione *onClick* descritta in precedenza si ottiene, non solo l'effetto di animazione impostato, ma anche il caricamento di due generazioni di figli e sotto-figli a partire dall'elemento cliccato spostatosi nel focus.

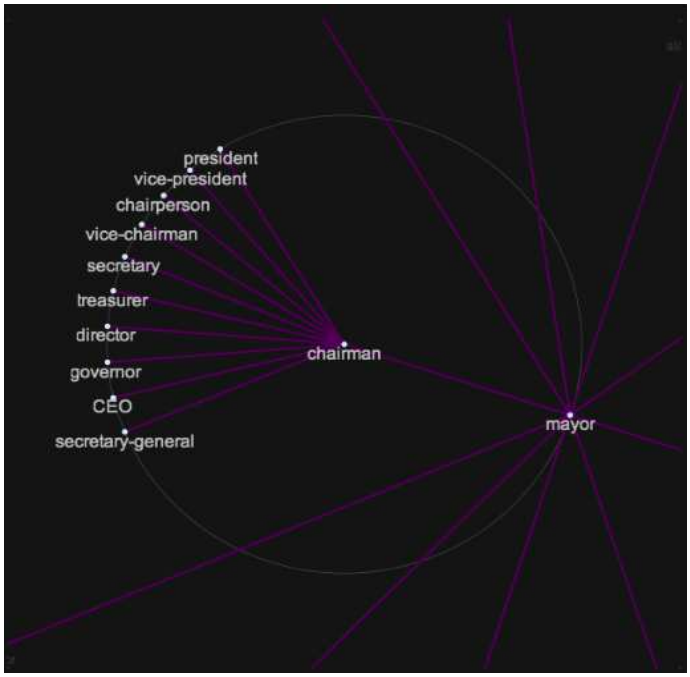

Figura 6: Rappresentazione del Visualizzatore dopo una ricerca

Durante queste operazioni di ricerca di navigazione i dati verranno ulteriormente restituiti secondo un altra disposizione nel box destro che, dopo aver effettuato la ricerca, informa in maniera dettagliata l'utente sulla ricerca appena fatta; nello specifico è presente in alto la parola che in quel momento si trova al centro del grafo (riconosciuto come il focus dell'albero) seguita da tre campi quali *Neighbour*, *Cosine* e *Similarity.* Quest'ultimi dati, restituiti sotto forma di tabella, mostrano, dunque, tutti i paradigmatic neighbours (vicini paradigmatici) della parola centrale con i relativi coseni espressi da un grado di similarità posto all'estrema destra del box per ogni record riportato (es. Really High, Medium, Low). Inoltre visto che si tratta di coseni, e che questo valore può variare tra 0 ed 1, è stata aggiunta una barra che

esprime il valore trovato in percentuale caricandosi ogni volta che una nuova parola si posta al centro del grafo.

.

Figura 7: Parte del box destro dopo aver effettuato una ricerca

Conclusione

Questo lavoro è concentrato su uno degli approcci principali alla semantica distribuzionale: l'approccio paradigmatico. Il grado di vicinanza paradigmatica tra due parole viene misurato calcolando il coseno tra i loro vettori distribuzionali.
Lo scopo del progetto è stato quello di costruire un tool che permetta l'esplorazione dei dati, ricavati da Distributional Memory in TypeDm, secondo una rappresentazione grafica che rappresenta la distanza semantica che intercorre tra due termini.
L’interfaccia grafica realizzata, grazie all'utilizzo della libreria JIT, mostra in maniera molto intuitiva la similarità semantica che intercorre tra le parole presenti nel database precedentemente creato.
Pur avendo raggiunto dei risultati soddisfacenti il grafo potrebbe essere ulteriormente migliorato attraverso l'implementazione di funzioni e di modifiche della libreria che consentano, ad esempio, di impostare la lunghezza dei rami in relazione alla misura del coseno cosi da evidenziare più facilmente, in termini spaziali, la vicinanza paradigmatica. Un ulteriore miglioramento riguarderebbe la query utilizzata per effettuare la ricerca; infatti con strumenti più veloci e potenti si potrebbero creare array contenenti più dati e di conseguenza più generazioni per una navigazione estesa all'interno del grafo.

Bibliografia

Baroni, M., & Lenci, A. (2009), "*One semantic memory, many semantic tasks"*. Proceedings of the EACL *Workshop on Geometrical Models of Natural Language Semantics*, Athens, 31st March.

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

Feteke, J.D. (2004), "*The InfoVis Toolkit*". Université Paris-Sud.

Firth, J. R. (1957), "*A synopsis of linguistic theory 1930-1955*". In *Studies in Linguistic Analysis*, pp. 1-32. Blackwell, Oxford.

Gentner, D. (1983), "*Structure-mapping: A theoretical framework for analogy*". *Cognitive Science*, 7 (2), 155-170.

Harris Zellig S. (1954), "*Distributional structure*". 146-62 [reprinted in Harris Zellig S. (1970), "*Papers in Structural and Transformational Linguistics"*. Dordrecht: Reidel. 775-794].

Lenci, A. (2008), "*Distributional semantics in linguistic and cognitive research*", in A. Lenci (a cura di), *From context to meaning: distributional models of the lexicon in linguistics and cognitive science*, numero speciale dell'*Italian Journal of Linguistics*, XX/1:1-31.

Lenci, A. (2009), "*Spazi di Parole. Metafore e Rappresentazioni Semantiche*". *Paradigmi*, 27:83-100.

Lin, D., & Pantel, P. (2001), "*DIRT – discover of inference rules from text*". In Proceedings of ACM SIGKDD Conference on *Knowledge Discovery and Data Mining* 2001, pp. 323-328.

Lowe, W. (2001), "*Towards a theory of semantic space*". In *Proceedings of the Twenty-rst Annual Conference of the Cognitive Science Society*, pp. 576-581.

Sahlgren M., 2006, *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in Highdimensional Vector Spaces*, Ph.D. dissertation, Department of Linguistics, Stockholm University.

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

Padó, S., & Lapata , M. (2007), "*Dependency-based construction of semantic space models*". *Computational Linguistics* XXXIII/2. 161-199.

Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2005), "*Using context-window overlapping in synonym discovery and ontology extension"*. Department of Computer Science Universidad Autonoma de Madrid.

Sahlgren, M. (2006), "*The Word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in highdimensional vector spaces*". Ph.D. Dissertation, Department of Linguistics, Stockholm University.

Salton, G., Wong, A., & Yang, C. (1975), "*A vector space model for automatic indexing*". *Communications of the ACM*, 18 (11), 613-620.

Schutze, H., & Pedersen, J. (1993), "*A vector model for syntagmatic and paradigmatic relatedness*". In *Making Sense of Words: Proceedings of the Conference*, pp. 104-113, Oxford, England.

Turney, P. D., Littman, M. L., Bigham, J., & Shnayder, V. (2003), "C*ombining independent modules to solve multiple-choice synonym and analogy problems*". In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (RANLP-03), pp. 482-489, Borovets, Bulgaria.

Turney, P.D. (2006), "*Similarity of semantic relations*". *Computational Linguistics*, 32(3): 379–416.

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

Lakoff, G. & Johnson, M. (1980), "*Metaphors we live by"*. Chicago: University Press.

Miller, G. A., & Charles, W. G. (1991), "*Contextual correlates of semantic similarity*". *Language and Cognitive Processes*, VI: 1-28.

Ringraziamenti

Per concludere questo lavoro di tesi non posso non ringraziare chi mi ha sostenuto durante tutto questo percorso. Un ringraziamento particolare va alla mia famiglia che sempre mi ha sostenuto sia da un punto di vista morale che economico. Ringrazio anche i colleghi di corso che in questi anni hanno contribuito alla mia crescita professionale e che mi hanno sempre sostenuto, in particolare Alessandro, Francesco, Christian, Giancarlo e Tommaso.