Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea magistrale in matematica

Entropia, contenuto informativo e network linguistici: applicazioni al manoscritto di Voynich

Relatore:
Mirko Degli Esposti

Presentata da:
Gianluca Bosi

Correlatore:
Giampaolo Cristadoro

III Sessione, 17 luglio 2015
Anno Accademico 2013/2014

Contents

Introduction

The main goal of this thesis is to investigate, throughout a matematical approach, the information content and the language morphology of the Voynich Manuscript (VMS), known to be written in an unknown and still undeciphered alphabet.
Firstly, we develop and test some mathematical tecniques which we need to study the linguistic information.
Starting from the concept of entropy, developed in the context of information theory, we construct a measure of the information content in a text (Montemurro-Zanette measure), which allow us to perform a keyword extraction.
In the particular case of natural languages, we verify in section 1.3 that the information content is invariant under lemmatizzation and, in 1.4, we show that the nouns of a text contain more information that the other parts of speech, so we refine the keywords extraction by means of Pos-tagging algorithms. Finally, in the last section of the first chapter we show another keyword extraction method, based on intermittence in the word frequency patterns (burstiness), and we verify that there is some correlation between the results obtained by the two methods.
In chapter 3 we make some experiments to determine how the information content of a text varies under human and machine translation; in particular, we noticed that in this latter case the information content tends to decrease.
Then, a very important approach we used in our analysis, is to represent different language levels (as pages, chpaters, word windows) in a vectors space, basically using the co-occurence patterns of the keywords; so we were able to compare and regroup them by means of some clustering methods, which are presented and tested in appendix C.
This approach, applied in chapter 2 to the VMS, allows us to investigate the thematic structure of the manuscript and to discover semantical relations among his contents: in particular, we show that the classical thematic subdivision of the manuscript, as determined by philologists from the various illustrations contained in it, essentially overlaps with the subdivision obtained in section 2.3 with our methods. Moreover, we verify the existence of a semantic continuity between consecutive pages belonging to the same section.
The great percentage of hapax in the manuscript leads us to morphological observations: in fact, it suggests the language of the manuscript to be particularly inflected, like Latin.

In particular, by the research of consecutive hapax sequences we can plausibly indentify some proper nouns.
Then, we decide to go further in the study of VMS's language morphology; for this purpose we create in section 2.5 a "linguistic" graph, essencially based on the Hamming distance; comparing the topology of these nerworks for different language vocabularies, we observe that the VMS one differs for his remarkable connectedness and density; furthermore, we find out that a large set of words are obtained just by adding one letter at the beginning or at the end of a common root. We then extract those roots.
Finally, we show that there exist a moderate correlation between the semantic relevance and the morphological one, this latter seen as centrality in the linguistic graph.
In conclusion, the strong evidences in favour to the presence of an information content in the text suggest that this one was actually written in a real language. However, because of the very simple morphological costruction rules, it doesn't look like a known natural language, but rather an artificial one, created specifically for this text.

Introduzione

Questa tesi si propone di investigare, mediante un approccio puramente quantitativo, il contenuto informativo e la morfologia della lingua del manoscritto di Voynich (VMS), noto per essere redatto in un alfabeto sconosciuto e tuttora non decodificato.
Per prima cosa, si definiscono e sperimentano alcune tecniche matematiche per lo studio dell'informazione linguistica.
A partire dal concetto di entropia, sviluppato nel contesto della teoria della informazione, si costruisce una misura del contenuto informativo di un testo (misura di Montemurro-Zanette), con la conseguente estrazione delle parole chiave (keywords).
Nel caso di lingue naturali conosciute, verifichiamo nella sezione 1.3 che il contenuto informativo è invariante per lemmatizzazione e, nella sezione 1.4, che i nomi contengono più informazione delle altre parti del discorso, e dunque l'estrazione di keywords può essere eventualmente perfezionata con l'intervento di algoritmi di Pos-tagging. Viene inoltre presentato, nell'ultima sezione del capitolo 1, un'altro metodo per estrarre keywords, basato sull'intermittenza del segnale linguistico (burstiness); verifichiamo che vi è una certa correlazione tra i risultati ottenuti coi due metodi.
Nel capitolo 3, invece, osserviamo come varia il contenuto informativo nel caso di un testo sottoposto a traduzione in altre lingue, sia questa operata da un uomo o da una macchina (traduzione automatica): in particolare si osserva che, in quest'ultimo caso, il contenuto informativo tende ad abbassarsi.
D'altra parte, i profili di occorrenza delle keywords sui diversi livelli testuali (pagine, capitoli, finestre di parole) ci consentono di rappresentare gli stessi in uno spazio vettoriale, e perciò di confrontarli e raggrupparli sotto l'aspetto semantico sfruttando opportuni algoritmi di clustering, presentati in appendice C.
Questo approccio, applicato nel capitolo 2 al VMS, ci permette di indagare la struttura tematica del manoscritto e le relazioni tra i suoi contenuti; a questo proposito, abbiamo mostrato che la classica suddivisione tematica del manoscritto, operata dai filologi a partire dalle numerose illustrazioni presenti in esso, combacia sostanzialmente con la suddivisione ottenuta nella sezione 2.3 tramire i nostri metodi. Inoltre abbiamo verificato che esiste una continuità semantica tra pagine consecutive appartenenti a una stessa sezione.
La grande quantità di hapax nel manoscritto ci porta poi a considerazioni di tipo mor-

fologico: suggerisce infatti che la lingua del manoscritto sia particolarmente flessiva, al pari del latino. La ricerca, in particolare, di sequenze di hapax consecutivi (sezione 2.4), ci porta a identificare -verosimilmente- alcuni nomi propri.
Proprio per approfondire la morfologia della lingua si costruisce nella sezione 2.5 un grafo linguistico basato sostanzialmente sulla distanza di Hamming; confrontando la topologia di questi grafi per alcune lingue e per la lingua del VMS si osserva che quest'ultimo si distingue per maggiore densità e connessione, e si trovano ampi insiemi di parole ottenute semplicemente aggiungendo o togliendo un breve suffisso a una radice comune.
Infine si verifica che esiste una certa correlazione tra la rilevanza semantica e quella morfologica, intesa come centralità nel grafo suddetto.
Traendo le conclusioni, i forti indizi a favore della presenza di un contenuto informativo nel testo confermano l'ipotesi che questo sia scritto in una vera lingua. Tuttavia, data la notevole semplicità delle regole di costruzione morfologiche, a nostro parere non sembra assimilabile ad una lingua naturale conosciuta, ma piuttosto ad una artificiale, creata appositamente per questo testo.

Ringraziamenti

Desidero innanzitutto ringraziare Mirko Degli Esposti, relatore di questa tesi, e Giampaolo Cristadoro, che ne è correlatore, per avermi mostrato per primi quanto la matematica possa dire e fare in un ambito, che può apparire distante, come il linguaggio e l'informazione scritta.
Proprio loro mi hanno consigliato il corso di Fabio Tamburini, che ringrazio sentitamente, il quale mi ha insegnato alcuni approcci computazionali alla risoluzione di problemi linguistici nel suo corso di NLP, e un ringraziamento speciale va poi a Dario Cardamone, mio collega e frequentante del suddetto corso, insieme al quale ho sviluppato il relativo progetto d'esame, di cui una parte è presente in appendice C del presente elaborato.
Non posso evitare, poi, di rivolgere un grande 'grazie' a Marco Ruffino, che mi ha fornito gli strumenti di base e il senso critico necessario all'analisi dei networks sociali, il cui corso alla facoltà di informatica di Bologna consiglierei davvero a chiunque.
Inoltre rivolgo i miei ringraziamenti a Marcelo Montemurro, della Faculty of Life Sciences di Manchester, sulle cui idee è basato molto del lavoro svolto in questo elaborato.
Per quanto riguarda il lavoro svolto per la tesi, ricordo anche l'aiuto in extremis che mi hanno dato Alex Casella e Mariagiulia De Maria, miei cari amici matematici, revisionandomi l'introduzione.
Dedico poi un ringraziamento del tutto speciale al mio amico e professore Alberto Parmeggiani, i cui consigli e avvertimenti in questi cinque anni sono stati per me realmente preziosi.
Sento il dovere di rivolgere i miei ringraziamenti e le mie scuse anche a tutto il personale amministrativo, ma particolarmente ad Alice Barbieri, che ho disturbato a ritmi regolari durante questi cinque anni e -inutile dirlo- mi ha sempre dato una mano, con grande pazienza e prontezza.
Infine, vorrei ringraziare tutti i miei amici e parenti che in questi anni mi hanno sostenuto, economicamente ed emotivamente, perché il loro aiuto è stato grande e indispensabile; tra questi voglio ricordare soprattutto i miei genitori, Matteo e Simona, mio fratello Davide e la mia fidanzata Wenting.

Chapter 1

L'entropia come misura dell'informazione in un testo

1.1 Teoria dell'informazione: l'entropia e la mutua informazione

L'unità di misura dell'informazione è il bit: un bit è definito come l'incertezza di un evento che si verifica con probabilità $\frac{1}{2}$ o, equivalentemente, come l'informazione che si ottiene nel conoscere l'esito di tale evento.
Per misurare l'informazione contenuta in un messaggio si utilizza l'entropia; formalmente, l'entropia $H(X)$ di una variabile aleatoria discreta X con distribuzione di probabilità $p(x)$ è la misura della quantità di incertezza, o informazione, associata al valore di X [10]:

$$H(X) := -\sum_{x \in X} p(x) \log_2 p(x).$$

Un'altra misura di fondamentale importanza nella teoria dell'informazione è la mutua informazione tra due variabili aleatorie X e Y, definita come segue:

$$M(X,Y) := \sum_{x \in X, y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)},$$

dove $p(x,y)$ è la distribuzione di probabilità congiunta di X e Y, mentre $p(x)$ e $p(y)$ sono, rispettivamente, le distribuzioni di probabilità marginali delle due variabili. La mutua informazione ci dice quanto la conoscenza di Y riduce l'incertezza di X, dunque quantifica l'informazione "mutua", ovvero condivisa da X e Y. Infatti possiamo scrivere:

$$M(X,Y) = H(X) - H(X|Y),$$

dove $H(X|Y)$ è l'entropia condizionata di X dato Y, ovvero:

$$H(X|Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x|y).$$

In realtà, la mutua informazione può vedersi come un caso particolare della entropia relativa $D(p//q)$, così definita:

$$D(p//q) := \sum_{x\in X} p(x)\log_2 \frac{p(x)}{q(x)},$$

dove $q(x)$ è una arbitraria distribuzione di probabilità su X; Infatti risulta:

$$I(X,Y) = D(p(x,y)//p(x)p(y)).$$

L'entropia relativa viene spesso interpretata come la "distanza" tra le distribuzioni $p(x)$ e $q(x)$, seppure in verità non lo sia: se da una parte è sempre non negativa (ed è uguale a 0 se e solo se $p = q$), non è simmetrica e non rispetta la disuglianza triangolare.

1.2 L'algoritmo di Montemurro-Zanette

E' stato mostrato in diversi studi [4] che la disomogeneità del profilo di occorrenza di una parola in un testo è un indicatore della sua rilevanza semantica: in effetti, la presenza di una parola solo in certe zone di un testo significa che essa è legata ad un particolare contesto o "sezione", e ne è perciò rappresentativa.
Basandosi su questa assunzione, l'algoritmo di Montemurro Zanette [2][3][4] si propone di quantificare il contributo di informazione che ciascuna parola da a quei contesti in cui essa appare: tale indice è proprio la differenza tra la mutua informazione di Shannon valutata sul nostro testo (ovvero tra le parole del testo e i contesti in cui lo si è suddiviso) e su una versione randomizzata dello stesso, ottenuta mescolando a random tutte le sue parole; in altri termini, misura quanto ciascuna delle parole si discosti dall'avere un profilo di occorrenza omogeneo.
Supponiamo ora di suddividere il testo in sezioni contigue di uguale lunghezza: si è giunti a mostrare sperimentalmente che esiste una particolare lunghezza che massimizza il valore del nostro indice (e dunque massimizza l'informazione), e che tale massimo è unico; questa lunghezza varia a seconda del genere letterario e della lingua del testo, ma si aggira tipicamente intorno al migliaio di parole.
Entriamo ora nel dettaglio, e deriviamo la scrittura esplicita dell'indice di informazione. Consideriamo un testo composto da N parole, che utilizza un vocabolario di K parole differenti, e una partizione P del testo in intervalli $A_1, ..., A_P$ di uguale lunghezza $s = \frac{N}{P}$ (la lunghezza è misurata in numero di parole).

Sia ora ω una parola che appare n volte nel testo e n_j volte nell'intervallo A_j; possiamo allora definire la probabilità condizionata di incontrare la parola ω nell'intervallo A_j, e sarà $p(\omega|A_j) = \frac{n_j}{s}$. Allora la probabilità di ω in tutto il testo è

$$\frac{n}{N} = p(\omega) = \sum_{j=1}^{P} p(\omega|A_j)p(A_j),$$

dove $p(A_j) = \frac{1}{P}, \forall j$.
Possiamo allora usare la regola di Bayes e calcolare la probabilità $p(A_j|\omega)$; otteniamo:

$$p(A_j|\omega) = \frac{p(\omega|A_j)p(A_j)}{p(\omega)} = \frac{n_j}{n},$$

che ci dice quanto è probabile trovarsi nell'intervallo j-mo una volta che ci si imbatta nella parola ω.
Possiamo ora usare queste quantità per calcolare la mutua informazione di Shannon tra le sezioni del testo e la distribuzione delle parole, come segue:

$$M(P,\Omega) = \sum_{A_j,\omega} p(A_j,\omega) \log_2 \frac{p(A_j,\omega)}{p(A_j)p(\omega)}$$

Si vuole ora sottrarre a questa quantità la mutua informazione che otterremmo in un generico testo random: così, infatti, possiamo ottenere una misura del contributo informativo delle parole, inteso appunto come tendenza delle parole a distribuirsi in maniera disomogenea nel testo.
Sia dunque $< \tilde{M}(A_j,\Omega) >$ la mutua informazione media tra tutte le possibili permutazioni del testo; possiamo allora definire il nostro indice di informazione come segue:

$$\Delta I(P,\Omega) := M(P,\Omega) - < \tilde{M}(P,\Omega) > . \tag{1.1}$$

Osserviamo che, se indichiamo con $\tilde{p}(A_j|\omega)$, $\tilde{p}(\omega)$ e $\tilde{p}(A_j)$ le probabilità calcolate sul testo random e notiamo che $\tilde{p}(A_j) = p(A_j)$ e $\tilde{p}(\omega) = p(\omega)$, la misura (1.1) può essere scritta nei termini di una differenza di entropie calcolate, rispettivamente, sul nostro testo e sulla sua versione randomizzata:

$$\Delta I(P,\Omega) = M(P,\Omega) - < \tilde{M}(P,\Omega) > =$$

$$= H(P) - H(P|\Omega) - (< \tilde{H}(P) > - < \tilde{H}(P|\Omega) >) = < \tilde{H}(P|\Omega) > - H(P|\Omega) =$$

$$= \sum_{\omega=1}^{K} p(\omega) \sum_{j=1}^{P} [p(A_j|\omega) \log_2 p(A_j|\omega) - < \tilde{p}(A_j|\omega) \log_2 \tilde{p}(A_j|\omega) >]$$

$$= \sum_{\omega=1}^{K} p(\omega) \left[< \tilde{H}(P|\omega) > - H(P|\omega) \right].$$

Ora resta solo da ricavare la scrittura esplicita di $< \tilde{H}(P|\omega) >$. D'altra parte, su un testo random suddiviso in P parti di uguale lunghezza, con ω parola che occorre n volte nel testo e n_j volte nell'intervalo A_j, otteniamo:

$$\tilde{H}(P|\omega) = -\sum_{j=1}^{P} \frac{n_j}{n} \log_2 \frac{n_j}{n}.$$

Allora l'entropia media su tutte le possibili realizzazioni random del testo è data da:

$$< \tilde{H}(P|\omega) >= - \sum_{\substack{m_1+...+m_P=n, \\ m_j \leq \frac{N}{P} \forall j}} \left[p(m_1, ..., m_P) \sum_{j=1}^{P} \frac{m_j}{n} \log_2 \frac{m_j}{n} \right] \tag{1.2}$$

dove $p(m_1, ..., m_P)$ è la probabilità di trovare m_j occorrenze di ω nella parte j. Se spezziamo l'espressione 1.2 rispetto a ciascuno dei P termini della seconda somma, otteniamo:

$$- \sum_{\substack{m_1+...+m_P=n, \\ m_j \leq \frac{N}{P} \forall j}} \left[p(m_1, ..., m_P) \frac{m_1}{n} \log_2 \frac{m_1}{n} \right] - ... - \sum_{\substack{m_1+...+m_P=n, \\ m_j \leq \frac{N}{P} \forall j}} \left[p(m_1, ..., m_P) \frac{m_P}{n} \log_2 \frac{m_P}{n} \right] =$$

$$= -P \sum_{m_1=1}^{min\{n, \frac{N}{P}\}} \sum_{m_2+...+m_P=n-m_1} p(m_1, ..., m_n) \frac{m_1}{n} \log_2 \frac{m_1}{n} =$$

$$= -P \sum_{m_1=1}^{min\{n, \frac{N}{P}\}} p(m_1) \frac{m_1}{n} \log_2 \frac{m_1}{n}$$

Riassumendo, possiamo scrivere l'entropia media mediante la seguente formula:

$$< \tilde{H}(P|\omega) >= -P \sum_{m=1}^{min\{n, N/P\}} p(m) \frac{m}{n} \log_2 \frac{m}{n},$$

dove $p(m)$ è la probabilità di trovare esattamente m volte la parola ω in uno degli intervalli, ovvero:

$$p(m) = \frac{\binom{n}{m} \binom{N-n}{N/P-m}}{\binom{N}{N/P}}.$$

Abbiamo ottenuto una scrittura esplicita di ciascun termine del nostro indice.

1.3 Confronto tra l'informazione di un testo e della sua versione lemmatizzata

Il corpus da noi considerato in questo capitolo è composto da 10 testi di diverso genere, nella loro versione in lingua inglese: romanzi, testi scientifici e trattati filosofici (tabella 1.1); indichereremo questo corpus con il simbolo Γ. Si tratta dunque di un corpus di modeste dimensioni, che nondimeno ci permetterà di illustrare alcune interessanti proprietà e proporre possibili spunti di riflessione.
Per prima cosa abbiamo testato l'algoritmo MZ, calcolando le curve di informazione (ovvero i grafici della funzione $\Delta I(P, \Omega)$ di un determinato testo, al variare della lunghezza s degli intervalli) ed estratto le parole di maggiore contenuto informativo (keywords) dai nostri testi, comparando questi risultati con quelli ottenuti su una versione lemmatizzata del nostro Corpus: lemmatizzare un testo significa sostituire ciascuna parola col suo lemma, ovvero con la sua forma non flessa; ad esempio:

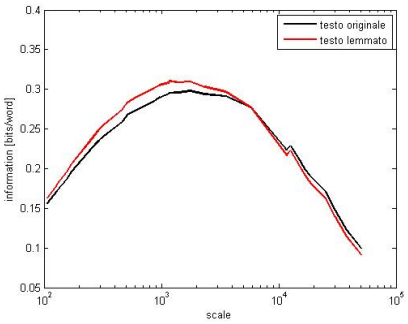
$$\begin{gathered} \text{I} \rightarrow \text{I} \\ \text{was} \rightarrow \text{be} \\ \text{reading} \rightarrow \text{read} \\ \text{some} \rightarrow \text{some} \\ \text{papers} \rightarrow \text{paper} \end{gathered}$$

Per lemmatizzare i nostri testi ci siamo serviti di un annotatore grammaticale probabilistico che fa uso di alberi decisionali: il Tree Tagger, sviluppato da Helmut Schmid e disponibile gratuitamente sul web [7] (si tratta anche e principalmente di un annotatore per parti del discorso, di cui parleremo -entrando anche nel merito dell'algoritmo- nella sezione dedicata 1.4.2 e nell'appendice B).
Abbiamo dunque impiegato il Tree-Tagger e generato, per ciascun testo di Γ, una sua versione lemmatizzata: indichereremo con Λ il corpus dei testi così generati.
Per prima cosa, abbiamo osservato che la cardinalità dei vocabolario nei testi di Γ e Λ cambia, diminuendo nel caso dei testi lemmatizzati di circa il 20/25 % (tabella 1.1).
Lemmatizzando un testo si hanno dunque due grandi benefici: prima di tutto, poiché i lemmi radunano in se intere classi di flessioni, si eliminano le ridondanze nel vocabolario. Il secondo vantaggio è la riduzione della sparsità dei dati, problema sempre presente nello studio dei testi in lingua naturale dal momento che, se si ordinano le parole in senso decrescente rispetto alla frequenza, quest'ultima decresce con l'andamento di una power law di esponente ~ -1 (legge di Zipf). Allora, lemmatizzando, si riduce la cardinalità del vocabolario e si ha, conseguentemente, una crescita della frequenza media delle parole.

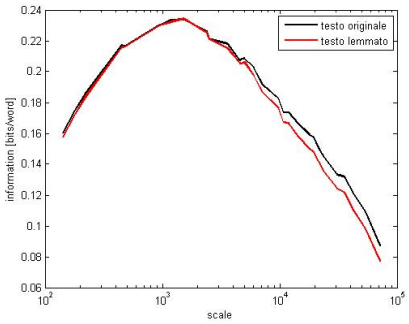
Short name	Title	Author	Text length	Vocabulary dimension	Vocabulary dimension (corpus Λ)
moby	Moby Dick	H.Melville	215939	17548	13148 (74,9%)
darwin	On the Origin of species	C.Darwin	151079	6983	5397 (77,3%)
quixote	Don Quixote	M.Cervantes	402964	14876	10916 (73,4%)
wrnpc	War and Peace	L.Tolstoy	565159	18048	13342 (73,9%)
larepubblica	La Repubblica	Platone	119479	7345	5408 (73,6%)
lincoln	Papers and Writings	A.Lincoln	552817	16009	12233 (76,4%)
hobbes	Leviathan	T.Hobbes	213971	9641	8369 (86,8%)
kant	Critique of Pure Reason	I.Kant	209792	6584	5078 (77,1%)
ulysses	Ulysses	J.Joyce	265304	29986	24893 (83,0%)
freud	The Interpretation of Dreams	S.Freud	54302	5966	4661 (78,1%)

Table 1.1: Corpus Γ. Tutti i testi considerati sono nella loro versione in lingua inglese.

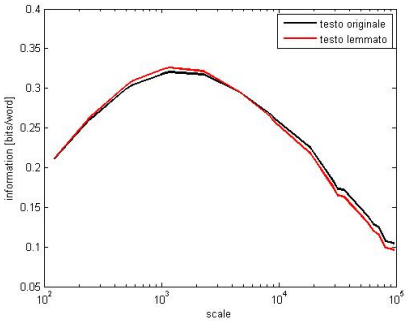
Applichiamo allora l'algoritmo di Montemurro-Zanette ai testi in Γ e ai rispettivi lemmati in Λ, per confrontare i grafici scala/entropia (figura 1.1). Osserviamo che le curve risultanti dalle due versioni di un medesimo testo finiscono quasi per sovrapporsi. Sebbene sarebbe necessario verificare questa proprietà su un corpus più ampio e vario al fine di validarla, i nostri esperimenti suggeriscono appunto che l'informazione contenuta da un testo e misurata dall'indice di M-Z sia invariante per lemmatizzazione del testo. Ciò è interessante se si riflette da un punto di vista qualitativo su cosa sia per noi l'informazione: se proviamo a leggere un testo lemmatizzato, osserviamo che esso è per noi comprensibile in quasi tutto il suo contenuto. Il testo lemmatizzato può contare sui contributi informativi di un numero inferiore di vocaboli, eppure la somma di queste quantità è la medesima ottenuta dal testo originale: dunque l'informazione dei lemmi va a ridistribuirsi sulle forme flesse.
In appendice A svolgiamo la stessa indagine su una serie di testi in italiano, lingua più flessiva dell'inglese; anche in quel caso i risultati confermano la nostra tesi, seppure in maniera meno netta (le curve non combaciano esattamente, ma si mantengono in stretta prossimità).
Effettuiamo ora, per Moby Dick, la keyword extraction usando il valore di P ottimale (tabella 1.2): Si nota che la classifica data dal testo lemmatizzato non presenta ridondanze (c'è 'whale', ma non 'whales'). A parte ciò, il risultato è simile.



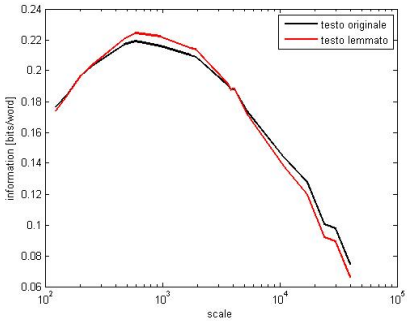
(a) darwin



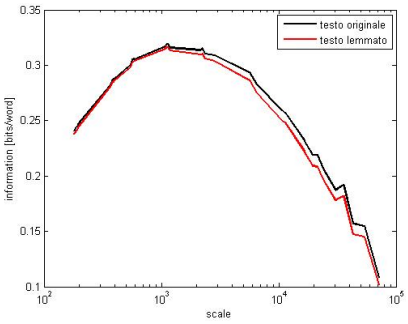
(b) moby



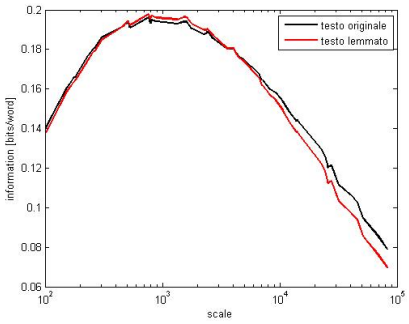
(c) wrnpc



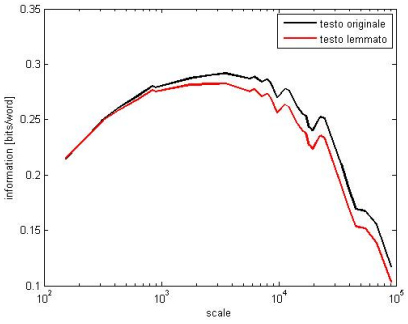
(d) larepubblica



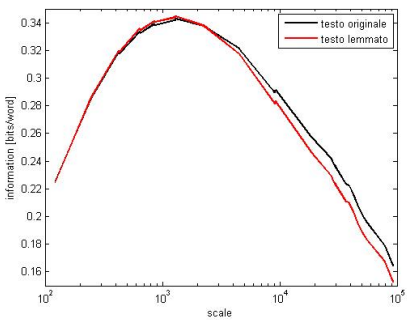
(e) hobbs



(f) quixote



(g) joyce



(h) lincoln

Figure 1.1: grafici scala/entropia

Prime 30 keywords da "moby"	Prime 30 keywords da "moby" (testo lemmatizzato)
I	I
whale	whale
you	you
ahab	ahab
is	ye
ye	queequeg
queequeg	boat
me	me
thou	thou
he	he
of	of
the	captain
captain	the
n't	n't
's	his
boat	my
his	stubb
my	jonah
stubb	him
was	starbuck
jonah	her
him	sir
whales	bildad
starbuck	sperm
her	white
sir	say
bildad	pip
sperm	peleg
white	do
pip	we

Table 1.2: prime 30 keywords da testo originale e lemmatizzato.

1.4 Estrazione di parole chiave in combinazione al POS-tagger

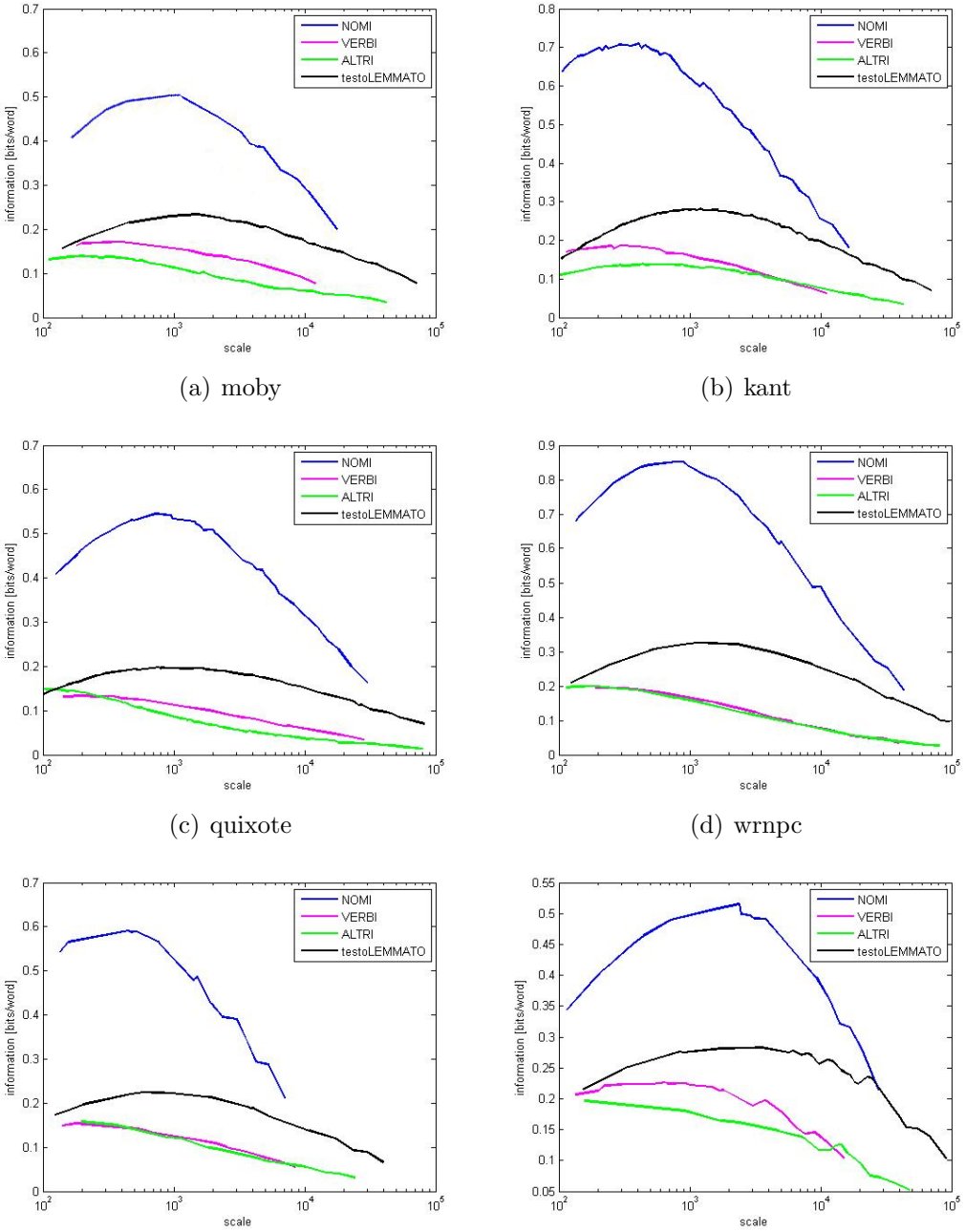
Come abbiamo potuto notare dalle estrazioni di keywords effettuate sul testo "moby", il risultato è compromesso dalla presenza di pronomi, preposizioni e verbi. Infatti, è la categoria dei nomi quella che raccoglie in se il cuore del contenuto semantico di un testo, che ne stabilisce gli eventuali personaggi, ne scandisce i temi. Abbiamo dunque pensato di scremare la nostra estrazione provando a comporre in maniera opportuna un analizzatore per parti del discorso all'algoritmo MZ.
Nell'appendice B vengono introdotte le idee fondamentali che stanno alla base della realizzazione di un POS-tagger, con due esempi di algoritmi.

1.4.1 Combinazione MZ e POS-tagging

Da ogni testo di Λ, il nostro corpus lemmatizzato, consideriamo ora il "testo" composto dalla sequenza ordinata di tutti i suoi verbi; allo stesso modo consideriamo il "testo" composto dalla sequenza dei nomi, e infine quello composto da tutte quelle parole che non sono né verbi, né nomi; abbiamo così tre nuovi corpus, che indicheremo, rispettivamente, con i simboli $\mathbb{N}$,$\mathbb{V}$ e $\mathbb{A}$.
Abbiamo già notato come, soprattutto quando si estraggono parole chiave da un romanzo, vi sono molti pronomi (anche alcuni verbi e avverbi). Dunque, per "migliorare" la nostra estrazione, abbiamo provato ad effettuare la keyword extraction sui tre corpus $\mathbb{N}$,$\mathbb{V}$ e $\mathbb{A}$.
Per prima cosa abbiamo stampato i grafici scala/entropia di tali testi, paragonandoli coi grafici dei testi di Λ (figura 1.2): si nota che $\mathbb{N}$,$\mathbb{V}$ e $\mathbb{A}$ formano anch'essi delle curve con dei chiari valori massimi, in generale non coincidenti né tra loro, né col massimo del testo lemmatizzato.
La presenza di curve di informazioni analoghe a quelle già osservate nei testi integrali, ma tra loro differenti, è interessante: infatti mostra che l'informazione viene raccolta maggiormente su alcune categorie morfosintattiche, soprattutto in quella dei nomi.
Utilizziamo ora i valori di massimo trovati per effettuare la Keyword Extraction da $\mathbb{N}$,$\mathbb{V}$ e $\mathbb{A}$ rispettivamente. Si noti che, a priori, non è detto che una estrazione effettuata a partire da questi "testi" dia qualche risultato, non trattandosi, per l'appunto, di veri e propri testi letterari. Tuttavia il risultato è molto positivo, come possiamo valutare dalla tabella 1.3 in cui è presentato il risultato dell'estrazione per i tre testi tratti da "moby". In particolare, nella colonna dei nomi possiamo ritrovare ai primi posti tutti i principali personaggi della narrazione, mentre nella seconda colonna rileviamo la presenza di verbi caratteristici del romanzo, quali 'cook','swim','say','tell', 'die','pull','sink','look'.

keywords da $\mathbb{N}$	keywords da $\mathbb{V}$	keywords da $\mathbb{A}$
whale	say	i
ahab	get	you
queequeg	pull	his
ye	cry	he
jonah	think	the
boat	do	my
thou	be	me
captain	will	of
bildad	cook	him
peleg	look	n't
stubb	have	we
starbuck	go	her
sir	tell	a
sperm	feel	they
head	come	old
line	die	's
steelkit	kick	your
fish	sleep	what
bed	can	or
thee	consider	in
harpooneer	hear	as
landlord	sit	us
god	turn	that
sailor	would	here
leg	want	there
gentleman	sink	from
ship	seem	white
coffin	start	our
carpenter	let	no
white	swim	to

Table 1.3: prime 30 keywords estratte, rispettivamente, dalle versioni $\mathbb{N}$,$\mathbb{V}$,$\mathbb{A}$ del testo "moby"



(a) moby (b) kant

(c) quixote (d) wrnpc

(e) larepubblica (f) joyce

Figure 1.2: grafici scala/entropia per le versioni $\mathbb{N}$,$\mathbb{V}$ e $\mathbb{A}$ di alcuni testi del nostro Corpus. Le curve nere rappresentano le curve delle versioni lemmatizzate dei testi

1.5 Burstiness: l'intermittenza del segnale come metodo per estrarre keywords

Diversi sistemi naturali presentano un comportamento "intermittente"; ovvero, in essi si alternano lunghi periodi di inattività a fasi di iper-attività: un esempio classico sono i terremoti, ed anche nei testi letterari è possibile ravvisare intermittenza nei profili di occorrenza di alcune parole.
In generale, per visualizzare questo fenomeno si costruisce un segnale (una stringa binaria), in il cui elemento i-mo è uguale a 1 se l'evento considerato si verifica al tempo i, 0 altrimenti. Il pattern che si presenta è random (processo di Poisson) se la probabilità di un evento è indipendente dal tempo; in tal caso il tempo di ritorno tra due eventi consecutivi, τ , segue la distribuzione esponenziale. Si può invece parlare di burstiness (o intermittenza) quando, e nella misura in cui, il pattern che si presenta è lontano dall'essere random. Ci interessa dunque avere un indicatore efficace che possa misurare tale tendenza.
Siano σ_τ e μ_τ, rispettivamente, la deviazione standard e la media di $P(\tau)$; allora il coefficiente di variazione, definito come $\frac{\sigma_\tau}{\mu_\tau}$ è un semplice ed efficace modo di caratterizzare la deviazione da un segnale Poissoniano. Infatti assume valore 1 per un segnale Poissoniano e tende a $+\infty$ per segnali di distribuzione $P(\tau)$ con varianza infinita e media finita. Preferiamo però un indicatore che prenda valori all'interno di un intervallo finito, e dunque definiamo l'indice di burstiness come [13]:

$$B = \frac{\frac{\sigma_\tau}{\mu_\tau} - 1}{\frac{\sigma_\tau}{\mu_\tau} + 1} \tag{1.3}$$

L'indice B prende valori in $(-1, 1)$, e i segnali di maggior burstiness sono quelli che più si avvicinano al valore 1.
Abbiamo detto che le parole di un testo possono avere un comportamento intermittente: sperimentalmente si è osservato che tali parole tendono ad essere quelle di maggiore rilevanza semantica. Allora vogliamo usare l'indice B proprio per calcolare la burstiness di una parola ω in una stringa di testo $w_1, ..., w_n$; per farlo, trasformiamo quest'ultima in una stringa binaria sostituendo w_i con 0 qualora $w_i \neq \omega$, e con 1 qualora $w_i = \omega$.
Se, in questo modo, calcoliamo $B(\omega)$ per ciascuna $\omega \in V$, vocabolario della stringa di testo, che occorre almeno 10 volte nel testo, otteniamo una distribuzione di valori per B nell'intervallo $(-1, 1)$; le parole dal valore di B più elevato saranno le nostre keywords.
Vediamo, in figura 1.3, la distribuzione di B in tre testi di lingua inglese. I profili tendono a concentrare il loro massimo intorno allo 0, prendendo valori dentro l'intervallo (-0.4,0.8); una situazione diversa da quella random, poiché la maggior parte delle parole presentano un valore di burstiness maggiore di 0.

Procediamo dunque all'estrazione delle keywords (tabella 1.5): in grossetto sono evidenziate quelle che compaiono pure tra le prime trenta parole estratte col metodo MZ.

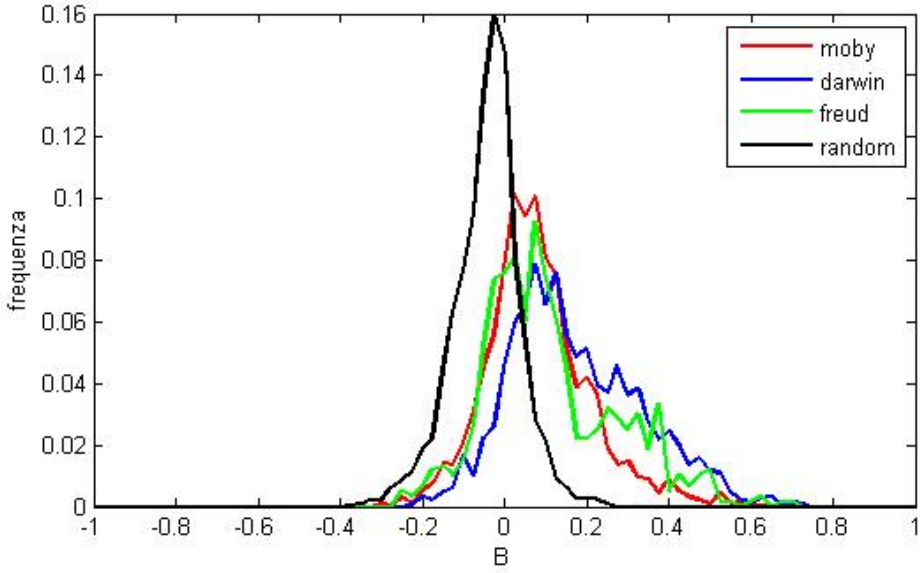


Figure 1.3: Distribuzione di frequenza dell'indice B in tre testi di lingua inglese e in un testo random ottenuto mescolando tutte le parole di "moby".

Il risultato è molto buono, e in questo caso non è "sporcato" dalla presenza di pronomi e articoli. Tuttavia, se da una parte vediamo comparire nuove keywords, molte di quelle parole significative che avevamo visto comparire utilizzando l'altro metodo non sono più presenti.
Perciò abbiamo deciso di visualizzare le parole di "moby" in un grafico indice MZ/ indice B (figura 1.4) per vedere come si dispongono su tale piano e poter studiare un criterio che ci permetta di pesare opportunamente il contributo di ciascun indice nella scelta delle nostre keywords. Il grafico mostra infatti che le parole più rilevanti (ad esempio queequeg, whale, ahab) sono quelle che superano una certa soglia per ciascuno dei due indici: hanno una buona entropia ed anche un comportamento intermittente. D'altro canto, le parole 'of','is','the','i', che posseggono molta informazione secondo l'indice MZ, hanno una burstiness prossima a 0. Dunque, scegliendo una opportuna intersezione tra quelle parole che superano una certa soglia di informazione, e quelle che superano un certo livello di intermittenza, potremo ottenere una keyword extraction più valida.

Questo approccio ha senso nella misura in cui i due indici misurano effettivamente cose diverse: evidentemente ci sono delle somiglianze, poiché una parole distribuita in maniera random otterrà un valore prossimo a 0 per entrambe le misure. Tuttavia, se ritorniamo ad analizzare la figura 1.4 vediamo che molte parole di informazione prossima allo 0 hanno però una burstiness che varia nell'intervallo (-0.4,0.4).
Allora, per avere una stima della diversità tra le due misure, possiamo calcolare la correlazione tra le coordinate dei punti del grafico; otteniamo una correlazione positiva

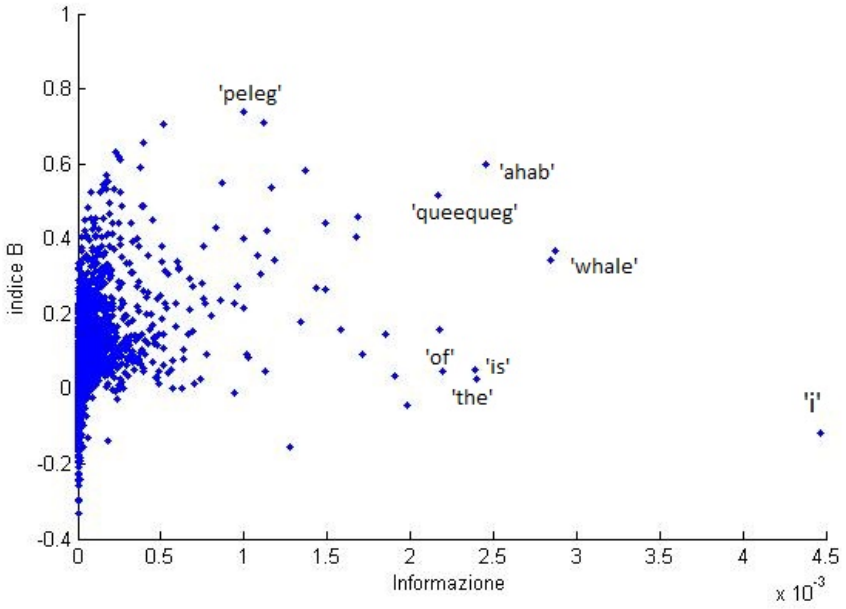


Figure 1.4: Grafico informazione/indice B; i punti blu rappresentano le parole del vocabolario del testo "moby".

presente ma piuttosto bassa, $\rho_{X,Y} = 0.28$. Questo valore potrebbe variare a seconda del testo su cui stiamo lavorando, e dunque non è affidabile come stima generale; Perciò abbiamo provato a calcolare la correlazione su diversi testi del nostro corpus (tabella 1.5), ottenendo in effetti risultati piuttosto diversi. Ad esempio nel testo "darwin" si ha una correlazione di circa 0.5, e dunque possiamo aspettarci che in questo caso i due indici tendano a misurare lo stesso fenomeno: detto altrimenti, burstiness e informazione sono piuttosto correlate in questo testo.
Dunque ci si potrebbe aspettare che, in un testo dove vi è alta correlazione, le keywords estratte con i due metodi siano le stesse: ciò potrebbe non verificarsi, ad esempio nel caso in cui la correlazione fosse tra parole con bassa informazione e bassa burstiness, che in un testo sono la maggior parte. A questo proposito, vediamo nella tabella1.5 che in "darwin", dove la correlzione è alta, si estraggono keywords diverse con i due metodi.

	moby	darwin	larepubblica	hobbes	kant	freud	lincoln
$\rho_{X,Y}$	0,28	0,52	0,22	0,35	0,40	0,37	0,21

Table 1.4: correlazione tra informazione e burstiness delle parole, in diversi testi

prime 30 keywords da "moby"	prime 30 keywords da "darwin"
peleg	instinct
bildad	wax
steelkilt	slave
landlord	hybrid
lakeman	slaves
radney	cells
gabriel	**sterility**
ahab	**hybrids**
whiteness	**instincts**
jonah	stripe
perth	formations
steak	bees
bunger	male
cook	**fertility**
ambergris	pollen
starbuck	fauna
guernsey	masters
parsee	floated
doubloon	miles
octavo	seed
blacksmith	beds
spade	silurian
queequeg	diagram
hussey	island
moby	value
dick	**groups**
ginger	land
streets	breeds
pulpit	embryo
captain	pupae

Table 1.5: Le prime 30 keywords estratte per burstiness. In grassetto sono evidenziate le parole presenti anche tra le prime 30 keywords estratte con l'algoritmo MZ.

Chapter 2

Il manoscritto di Voynich: contenuto informativo del testo e strutture morfologiche della lingua

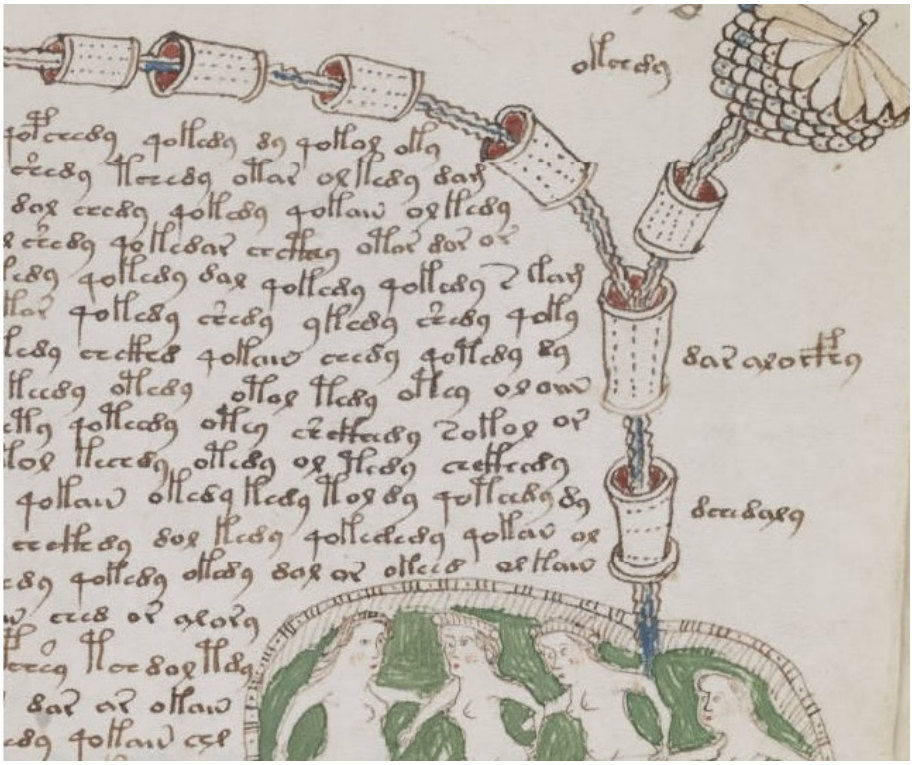


Figure 2.1: dettaglio di una pagina del manoscritto di Voynich

2.1 Introduzione

Il manoscritto di Voynich (VMS) è un manoscritto su pergamena di circa 240 pagine, datato mediante il test al radiocarbonio tra il 1404 e il 1438 d.C.. Contiene numerose illustrazioni di piante non chiaramente identificate, di costellazioni e astri, di ninfe, e di strani oggetti di natura non comprensibile. Tuttavia, l'aspetto più misterioso di questo manoscritto è il suo testo, scritto in un'elegante alfabeto sconosciuto, e che fino ad ora ha resistito ad ogni tentativo di decodifica portato avanti attraverso i secoli.
Sono stati fatti, ad ora, molti studi sulla natura di questo testo, e le spiegazioni che vengono date sulla dubbia natura del documento possono essere raccolte in tre correnti fondamentali [11]:
La prima è che si tratti di un vero testo letterario scritto in una lingua naturale che è però stato cifrato. Molti storici, in virtù delle origini del manoscritto, suppongono che tala lingua possa essere il tedesco o il latino.
La seconda è che si tratti di un testo scritto in un linguaggio artificiale inventato appositamente.
Infine vi è la possibilità che si tratti di un vero e proprio "scherzo". Ovvero che il VMS non contenga un testo di senso compiuto. A questa corrente appartiene anche chi pensa che le parole del testo non provengano da alcun linguaggio naturale.
Tuttavia la sequenza di parole del manoscritto ha dimostrato di possedere molte delle proprietà tipiche dei linguaggi naturali: di grande rilevanza è il fatto che il VMS rispetta la legge di Zipf. Questa è una condizione necessaria ma non sufficiente per provare che effettivamente si stia parlando di una lingua naturale. Eppure sembra molto difficile immaginare come, nel 1400, quando non si aveva ancora cognizione di questa ed altre proprietà dei linguaggi, si sia potuta generare una sequenza arbitraria che le rispettasse.
Resta il fatto che, ad oggi, il testo non è ancora stato decifrato; sono state proposte molte teorie e alcuni sostengono di aver effettivamente decifrato alcuni termini. Quello che può fare la matematica è, a nostro parere, cercare di andare sempre più a fondo nello studio della struttura linguistica e grammaticale di questa misteriosa lingua, fornendo utili indicazioni a chi, eventualmente, potrà un giorno decodificare il vero contenuto del VMS.

2.1.1 Alfabeto adottato

Come detto, non conosciamo la lingua in cui è scritto il manoscritto; tuttavia, per via della forma dei paragrafi e degli spazi in cui sono collocate le illustrazioni, è possibile dedurre che il senso della scrittura sia da sinistra verso destra [16]; del resto si può solo osservare come alcune lettere assomiglino a quelle dell'alfabeto romano, mentre altre ricordino i numeri arabi.
Per poter lavorare sul testo con strumenti informatici, è stato necessario agli studiosi trovare un insieme ragionevole di caratteri alfabetici per poterlo trascrivere. Ad ora

disponiamo di diverse possibili trascrizioni: l'ultima per ordine di tempo, chiamata EVA, permette di rappresentare il 99,86% del testo utilizzando 26 caratteri [16]. Tuttavia per ottenere una copertura totale del testo servirebbero altri 72 caratteri aggiuntivi, per via di diversi simboli che appaiono spesso solo una volta in tutto il manoscritto.
Altre trascrizioni molto adottate sono [14]: Currier, FSG, Bennet and Frogguy. In ogni caso, però, il problema sostanziale nel trascrivere il VMS è il saper identificare esattamente i "singoli caratteri" della lingua: infatti vi sono dei segni che possono avere la funzione di collegare un carattere a quello precedente e successivo, oppure brevi spazi tra due sequenze di caratteri che potrebbero (o no) indicare la separazione tra due parole. Ciascuna delle trascrizioni si caratterizza per scelte differenti in questo senso.
Nella presente tesi adotteremo la trascrizione di Takeshi Takahashi [15] , che è basata sostanzialmente sull'EVA, anche se con qualche semplificazione: in particolare si adotta un alfabero di 23 caratteri.

2.2 Curva dell'informazione MZ e parole chiave del testo

Proviamo ora ad applicare l'algoritmo MZ al manoscritto di Voynich [9].
Osserviamo che la curva dell'informazione (figura 2.2) presenta lo stesso aspetto di quelle che siamo abituati a vedere per gli altri testi letterari, ovvero una parabola il cui massimo -in questo caso- è raggiunto alla scala di circa 1300 parole, cui corrisponde una informazione media per parola di circa 0.34 bits/word. Questo risultato dimostra senz'altro che il manoscritto non è composto da una sequenza random di simboli, e la cosa che più ci interessa è la possibilità di classificare le parole in esso presenti in base alla rilevanza semantica: ovvero, possiamo estrarre keywords (tabella 2.1).

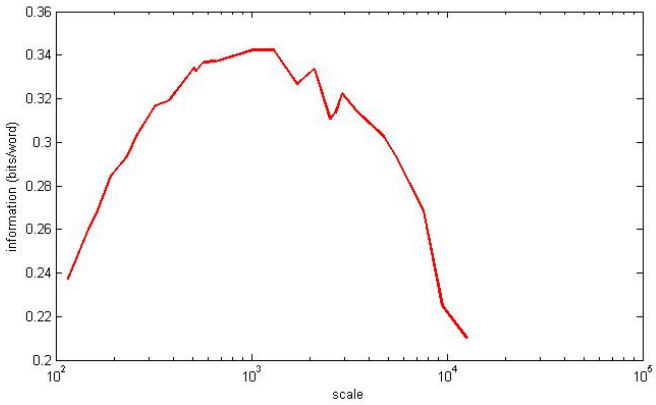


Figure 2.2: curva dell'informazione del VMS

prime 20 keywords da "voynich"
shedy
qokeedy
qokain
daiin
qokedy
chedy
qol
ol
chor
qokeey
cthy
chol
s
qokal
al
qokaiin
lchedy
dy
aiin
sho

Table 2.1: Le prime 20 keywords estratte dal manoscritto mediante l'algoritmo MZ.

Per scorgere le relazioni semantiche che esistono tra le keywords, applichiamo ora delle tecniche di clustering: il clustering consiste nel raggruppare un insieme di dati in base a una distanza stabilita tra essi: può essere utile, nell'ambito dello studio del linguaggio, per formare clusters (gruppi) di parole che condividano significati o funzioni linguistiche (in Appendice C sono presentati e testati i principali algoritmi).
Nel caso ora considerato vogliamo applicare il clustering k-means con la distanza coseno (cfr. Appendice C) ai profili di occorrenza delle prime 50 parole estratte; per profilo di occorrenza di una parola intendiamo il vettore normalizzato delle frequenze della stessa su una partizione del testo in intervalli di uguale lunghezza (in questo caso la lunghezza corrisponde alla scala in cui la curva dell'informazione raggiunge il massimo). Questo approccio ci permette di identificare tre gruppi di parole semanticamente simili:

cluster A: {al, aiin, ar, okeey, dal, otaiin, okeol, or, oteey, chdy, ytaiin, lkaiin}
cluster B: {shedy, qokeedy, qokain, qokedy, chedy, qol, ol, qokeey, qokal, qokaiin, lchedy, otedy, okedy, shey, qokey, oteedy, qotedy, otain, okain, qoteedy, qokar, chey}
cluster C: {daiin, chor, cthy, chol, s, dy, sho, chy, shol, dain, shor, cthor, shy, cthol, qotchy, dol}

Nonostante il kmeans sia un algoritmo iterativo, il risultato appena trascritto si è dimostrato molto stabile (segno dell'effettiva consistenza delle relazioni trovate): vediamo che le parole che appartengono a uno stesso gruppo condividono una notevole somiglianza morfologica (si veda ad esempio la radice "aii" nel cluster A, la radice "qo/qok" nel cluster B, e "ch/cth" nel cluster C). A tal proposito si veda anche la figura 2.3, in cui è presentato il hierarchical clustering delle prime 20 keywords e in cui è possibile apprezzare più agevolmente le relazioni che intercorrono tra i termini.

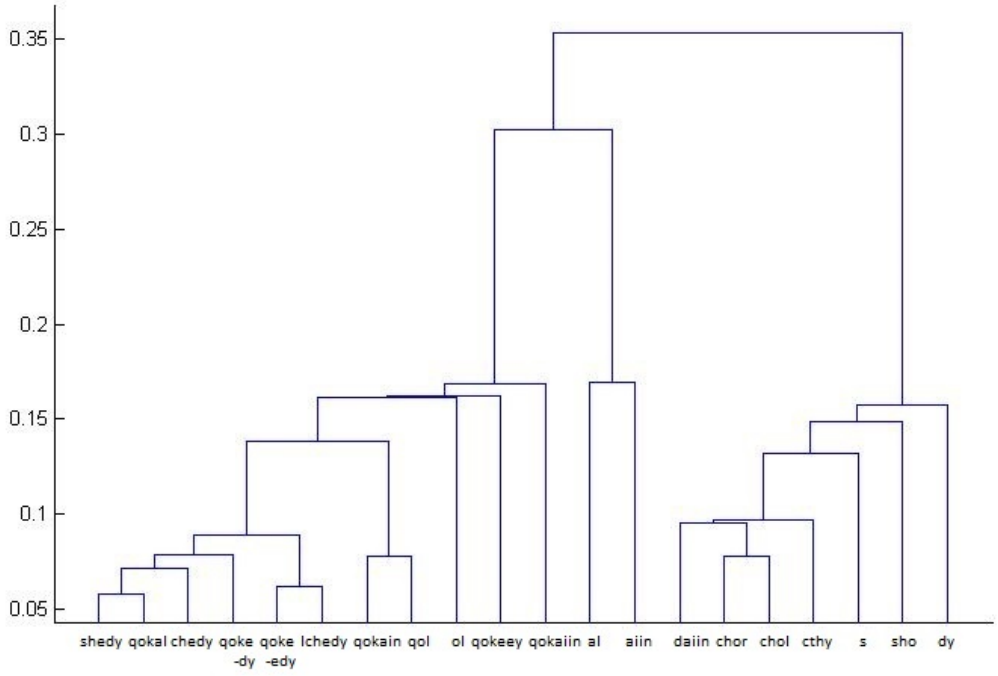


Figure 2.3: hierarchical clustering sulle prime 20 keywords del VMS. Si osservi come l'albero risulti sostanzialmente suddiviso in due grandi diramazioni.

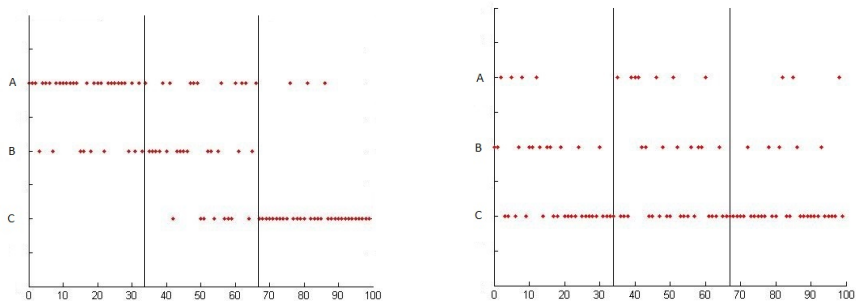
Può darsi, dunque, che molte di queste parole siano declinazioni di un medesimo lemma, e ne condividano il significato. Dato che si tratta di parole dall'alto contenuto semantico, possiamo anche ipotizzare che molte di queste appartengano alla categoria sintattica dei nomi: se il ragionamento fosse fin qui corretto, dovremmo ammettere che la grammatica della lingua che stiamo analizzando ammette i casi (ad esempio nominatico, genitivo, dativo, accusativo, ablativo ecc...). Infatti la molteplicità delle declinazioni che vediamo non possono essere giustificate semplicemente supponendo che i nomi abbiano il singolare e il plurale.

Altrimenti, un'altra possibilità è che le parole provenienti dalla medesima radice non provengano però dallo stesso lemma; allora dovremmo concludere che nel linguaggio che stiamo analizzando esiste un forte legame tra morfologia e semantica. Questo spiegherebbe infatti l'appartenenza di queste parole ad un medesimo cluster di significato.
Quelle appena espresse sono mere speculazioni, che però troveranno maggiori riscontri in seguito alle analisi delle prossime sezioni, in particolare nella 2.5., in cui costruiremo un network per studiare la morfologia della lingua del manoscritto.

2.3 Verifica della tradizionale suddivisione in sezioni del manoscritto

Basandosi principalmente sulle illustrazioni del VMS, i filologi lo hanno suddiviso in 5 sezioni, che differirebbero per il tema trattato. Lo scopo di questa sezione sarà valutare la presenza di relazioni semantiche tra pagine consecutive e verificare la validità di tale suddivisione.
Precisamente, il nostro approccio sarà il seguente: estrarremo le prime 50 keywords del VMS, e poi considereremo le pagine del manoscritto separatamente, associando a ciascuna di esse il vettore di frequenza di queste keywords in tale pagina. A questo punto compareremo questi profili tra loro mediante clustering: per prima cosa, ci aspettiamo che pagine consecutive si ritrovino tendenzialmente in un medesimo cluster, mostrando perciò continuità nella trattazione di un argomento. Inoltre, se la divisione tradizionale del VMS fosse corretta, dovremmo poterla ricostruire approssimativamente con questo metodo.
Ma, prima di testare questo metodo sul VMS, lo proviamo sulla Divina Commedia, testo di cui è ben nota la ripartizione in canti e cantiche: in particolare, le unità testuali qui considerate saranno i canti, che tenteremo di raggruppare in cantiche.

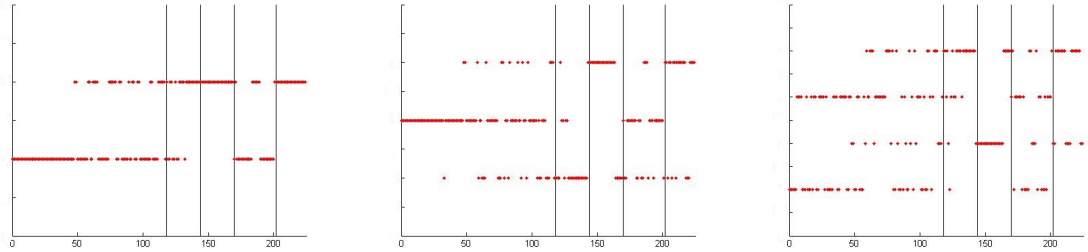
Nella figura 2.4(a) possiamo osservare i risultati: Se identifichiamo, rispettivamente, i cluster A, B, C con Inferno, Purgatorio, Paradiso, osserviamo che il 69% dei canti ha ottenuto una corretta attribuzione. Invece 2.4(b) mostra il risultato che si ottiene associando ad ogni canto i profili di occorrenza di 50 parole casuali nel testo, invece delle keywords: in questo caso non è possibile riconoscere le tre cantiche, in qualunque modo si identifichino i clusters otteniamo al più il 40% di attribuzioni corrette (si noti che persino il clustering banale che riunisse tutte le 100 cantiche in un solo cluster otterrebbe il 34%); si tratta di una classificazione random, e la ragione per cui il cluster C ha cardinalità maggiore, è dovuto al fatto che molti profili, essendo nulli, vengono associati ad un medesimo cluster. Tornando al grafico (a), possiamo osservare che il "rumore" (le oscillazioni di pagine consecutive tra diversi clusters) si concentra nella cantica del



(a) (b)

Figure 2.4: kmeans con distanza coseno sui canti della Divina Commedia, k=3. Le barre verticali indicano le separazioni corrette nelle cantiche. Sull'asse x abbiamo i canti, lungo l'asse y i tre clusters. Nel grafico (a) i profili dei canti sono dati dalle occorrenze delle prime 50 keywords; invece, in (b), sono dati dalle occorrenze di 50 parole estratte a caso dal testo.

Purgatorio, mentre nelle cantiche dell'Inferno e del Paradiso la quasi totalità dei canti trova giusta collocazione; questo è comprensibile se si pensa che nei canti del Purgatorio si trovano sia riflessioni relative alla dannazione che alla salvezza, e che come cantica presenta dunque tratti meno autoreferenziali.
La validità del nostro metodo deriva sostanzialmente dal fatto che le keywords, per come le intende Montemurro nel suo modello, non sono altro che quelle parole che danno la più alta informazione di quali siano le sezioni tematiche che compongono un testo.
Passiamo ora al VMS e, come detto, effettuiamo il clustering sui vettori-profilo delle pagine del manoscritto. I risultati permettono di scorgere alcuni temi separati nel testo,

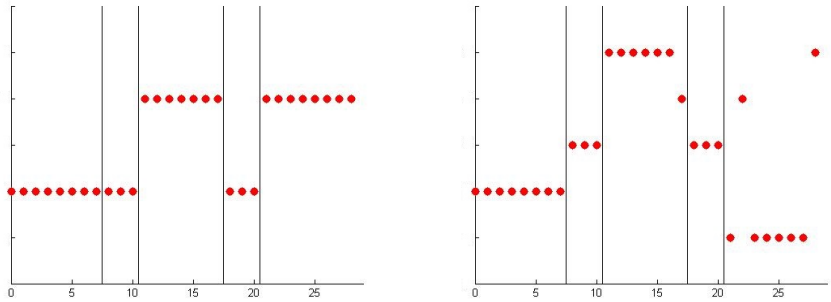


(a) 2 clusters (b) 3 clusters (c) 4 clusters

Figure 2.5: suddivisione delle pagine del VMS in clusters

per esempio quando si effettua un kmeans con k=2 o k=3 (figura 2.5); infatti in questi casi è possibile riscontrare lunghe sequenze di pagine consecutive appartenenti a un medesimo cluster (il che appunto è l'ennesima prova a sostegno dell'ipotesi che il manoscritto

possegga un contenuto linguistico), e si ritrovano delle suddivisioni in sezioni compatibili con quella classica, rappresentata dalle linee verticali nel grafico. Tuttavia, se si tenta di riconoscere quattro o cinque clusters, il risultato peggiora sempre di più, e la suddivisione in clusters è quasi random.
La difficoltà nell'ottenere un risultato preciso all'aumentare del numero di clusters è dovuto al fatto che le pagine del manoscritto, ricche di illustrazioni, hanno lunghezze molto variabili: non è così per i canti della Divina Commedia, tutti approssimativamente della stessa lunghezza; inevitabilmente tale disparità crea rumore. Ci troviamo, infatti, con diverse pagine il cui profilo è un vettore quasi ovunque nullo, e d'altra parte vettori con patterns piuttosto complessi.
Quindi abbiamo pensato di considerare blocchi più grandi di testo e di lunghezza omogenea; invece di pagine, finestre di parole di lunghezza arbitraria fissata. Dato che l'algoritmo MZ ci fornisce una lunghezza ideale per il riconoscimento delle sezioni tematiche del testo, abbiamo deciso di utilizzare proprio quella, che per il VMS è di circa 1300 parole.



(a) 2 clusters (b) 5 clusters

Figure 2.6: suddivisione delle finestre di testo del VMS in clusters.

Nel grafico 2.6(a) vediamo i risultati; la ricerca di due clusters combacia perfettamente, e ci permette di intuire non solo la suddivisione tematica del testo, ma anche le relazioni tra le sezioni, mostrando continuità tra la prima, la seconda e la quarta, e tra la terza e la quinta.
Vista la regolarità del risultato abbiamo tentato di identificare tutte e cinque le sezioni (figura 2.6 (b)): il risultato è molto soddisfacente, anche se la somiglianza tra seconda e quarta sezione le riconduce ad un medesimo cluster, lasciando di fatto uno dei cluster pressoché vuoto.
In definitiva possiamo sostenere che la suddivisone classica del manoscritto resta sostanzialmente confermata dai nostri esperimenti; la novità è che servendoci di un approccio a priori (ovvero senza ricorrere all'analisi filologica e delle illustrazioni del testo) siamo in

grado di sostenere che vi è una probabile corrispondenza di significato tra le illustrazioni del manoscritto e il suo contenuto testuale. Un ulteriore elemento a favore della tesi che il VMS contenga un effettivo messaggio linguistico.

2.3.1 Relazioni tra le sezioni del VMS

Dopo aver identificato le parole più rilevanti del manoscritto, abbiamo indagato le relazioni che intercorrono tra esse. Ora faremo la medesima cosa con le 5 sezioni del VMS: come sono legate tra di loro? quali differiscono di più?
Consideriamo ciascuna delle sezioni e vi associamo il solito vettore dato dai profili di occorrenza delle parole chiave ed effettuiamo il hierarchical clustering (questa volta useremo le prime 100 parole chiave; abbiamo comunque osservato che si ottiene il medesimo albero con le prime 30 o 50 chiavi).
L'albero (figura 2.7) ci informa del fatto che le nostre sezioni sono suddivise sostanzial-

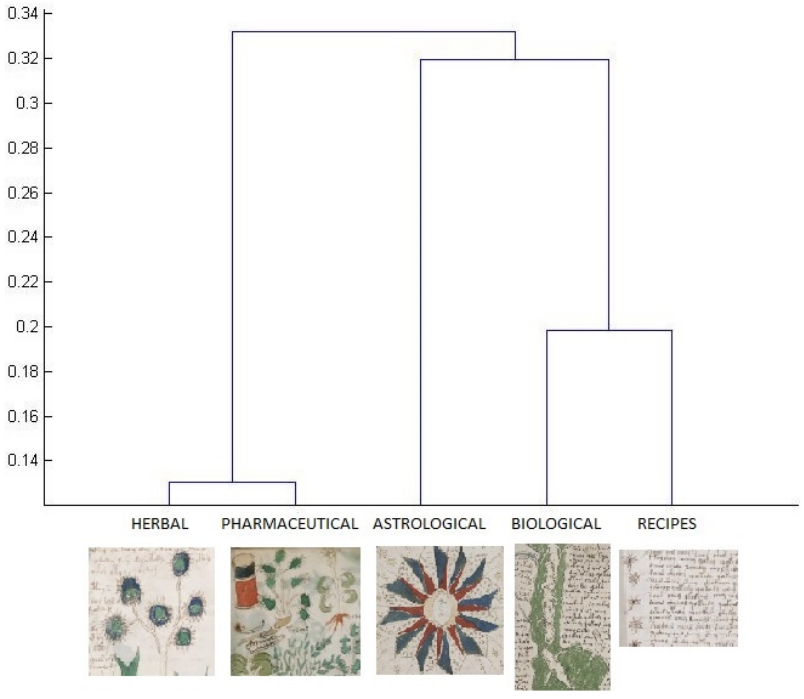


Figure 2.7: Hierarchical clustering sulle sezioni del VMS

mente in tre blocchi: il primo è formato dalla sezione 1, contenente immagini di piante, e dalla sezione 4, detta pharmaceutical per via delle rappresentazioni di strani rimedi e fiori. Tale accostamento (il più forte nell'albero) sembra essere dunque coerente a livello

intuitivo. Il secondo blocco è composto dalla sezione 3, definita "biological" poiché ricca di rappresentazioni di corpi umani, e dalla sezione 5, classificata come "recipes": dunque il tema delle ricette è legato al tema del corpo.
Un ultimo blocco, più o meno equidistante da tutti gli altri, è occupato dalla sezione numero 2, dedicata all'astrologia. Un argomento autonomo, dunque; anche ciò è piuttosto verosimile.

2.4 Sequenze di hapax: la presenza di "elenchi" nel manoscritto di Voynich

Un hapax legomenon (o hapax) è una parola che occorre una sola volta in un testo (o in un Corpus di testi). La quantità di hapax nel vocabolario di un testo dipende da diversi fattori, primo fra tutti la lingua. In effetti, una lingua molto flessiva -come per esempio il greco o il latino- tende ad avere una percentuale più alta di hapax; Viceversa, lingue meno flessive ne hanno un numero inferiore.
Ad esempio, nel Brown Corpus per la lingua inglese si osserva che circa la metà dei vocaboli sono hapax, e in generale la quantità si aggira tra il 40 e il 60 % nei testi di questa lingua.
Abbiamo cercato prima di tutto di fare qualche semplice considerazione statistica sulla quantità di hapax presenti nel VMS, anche per poter fare alcune ipotesi sulla struttura morfologica della lingua. Di seguito, nella tabella, sono riportati i valori percentuali sul manoscritto e su altri testi in diverse lingue naturali.

Short name	Title	Author	Language	Text length	Vocab. length	Hapax (%)
voynich	**The Voynich Manuscript**	**Unknown**	**Unknown**	**37813**	**8065**	**68,8**
aeneid	Eneide	Virgilio	LAT	64145	16850	56,03
alice	Alice's Adventures in Wonderland	L. Carrol	ENG	26667	2628	43,3
beagle	The Voyage of the Beagle	C. Darwin	ENG	208375	12669	40,48
commentari	Commentarii	G.G.Cesare	LAT	20520	5675	60,8
divina	Divina Commedia	A.Dante	ITA	101605	12832	57,27
drn_I	De Rerum Natura (Liber I)	Lucrezio	LAT	7281	2713	67,6
drn_II	De Rerum Natura (Liber II)	Lucrezio	LAT	7679	2976	68,82
drn_III	De Rerum Natura (Liber III)	Lucrezio	LAT	7441	2910	70,38
drn_IV	De Rerum Natura (Liber IV)	Lucrezio	LAT	8652	3326	67,5
drn_V	De Rerum Natura (Liber V)	Lucrezio	LAT	9534	3837	67,73
drn_VI	De Rerum Natura (Liber VI)	Lucrezio	LAT	8553	3333	67,39
ecloga	Ecloga	Virgilio	LAT	5841	2815	72,25
georgicon	Georgicon	Virgilio	LAT	14227	6882	71,19

inferno	Inferno(cantica)	A.Dante	ITA	34142	6504	61,95
jungle	The Jungle	U.Sinclair	ENG	151300	10105	43,43
mattia	Il fu Mattia Pascal	L.Pirandello	ITA	76831	10918	56,31
missisipi	Life on the Mississippi	M.Twain	ENG	146769	12242	44,97
moby	Moby Dick	H.Melville	ENG	215939	17548	44,56
origin	On the Origin of Species	C.Darwin	ENG	156770	7327	32,44
orlando	Orlando furioso	L.Ariosto	ITA	279771	19198	47,33
paradiso	Paradiso(cantica)	A.Dante	ITA	33410	6202	62,51
pinocchio	Le avventure di Pinocchio	C.Collodi	ITA	40660	6005	54,9
pride	Pride and Prejudice	J.Austen	ENG	122194	6412	38,46
proarchia	Pro A.Licinio Archia Poeta Oratio	M.T.Cicerone	LAT	3186	1529	71,94
proquinctio	Pro P.Quintio Oratio	M.T.Cicerone	LAT	8720	2876	65,54
prosestio	Pro P.Sestio Oratio	M.T.Cicerone	LAT	17041	5786	68,9
purgatorio	Purgatorio(cantica)	A.Dante	ITA	34053	6389	62,0
quixote	Don Quixote	M.Cervantes	ENG	402964	14876	36,7
saggia	Il Saggiatore	G.Galilei	ITA	84501	12854	58,32
sawyer	The Adventures of Tom Sawyer	M.Twain	ENG	71180	7324	48,53
sposi	I promessi sposi	A.Manzoni	ITA	223765	19603	51,55
ulysses	Ulysses	J.Joyce	ENG	265304	29986	54,49
wrnpc	War and Peace	L.Tolstoy	ENG	565159	18048	34,01
zeno	La coscienza di Zeno	I.Svevo	ITA	145487	13796	51,5

Osserviamo che il VMS ha un valore percentuale alto, ma compatibile con quello dei testi in lingua latina. Questo dunque ci suggerisce che la lingua del VMS deve essere flessiva.

Ma andiamo più nel dettaglio, e vediamo come si distribuiscono gli hapax nelle cinque

	Herbal	Astrolog.	Biological	Pharmac.	Recipes	VMS
Percentuali	15.56	25.42	9.74	16.61	13.95	14.67

Table 2.3: Rapporto percentuale tra numero di Hapax e lunghezza del testo, nelle varie sezioni del manoscritto e nel manoscritto per intero.

aree tematiche in cui si è tradizionalmente suddiviso il testo (tabella 2.3): vediamo che il numero di hapax è piuttosto disomogeneo. In particolare rileviamo il valore massimo nella sezione dedicata a temi presumibilmente astronomici (riguardo a questo picco faremo, tra poco, un'analisi più approfondita), mentre la sezione chiamata "farmaceutica" ha la percentuale più bassa. Per le altre sezioni, invece, il valore oscilla di poco intorno alla percentuale dell'intero testo.

Un altro fenomeno interessante legato agli hapax, si ha quando questi si presentano consecutivamente nel testo: di certo, se queste sequenze superano le 5 o 6 parole, si tratta di situazioni linguistiche straordinarie, di cui è abbastanza difficile enumerare esempi. Vediamo dunque, nel grafico 2.8, le code delle distribuzioni di queste sequenze in alcuni

testi di lingua naturale e nel VMS.
Notiamo che in più di un testo vi è un caso di sequenze di 10 o 11 hapax consecutivi:

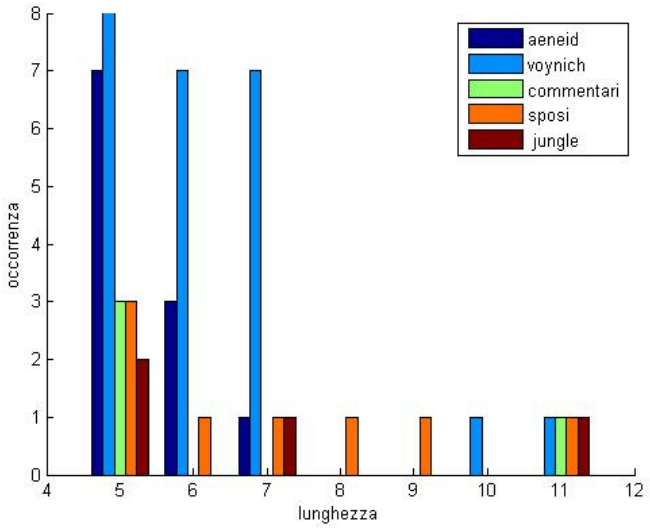


Figure 2.8: distribuzione di frequenza delle sequenze di almeno 5 hapax in cinque testi di lingua naturale e nel VMS (sull'asse x vi è la lunghezza della sequenza misurata come numero di HAPAX consecutivi nel testo.)

per comprendere meglio questo fenomeno andiamo a leggerle direttamente nei testi.
Guardiamo per esempio la sequenza di 11 hapax nei Commentarii di Cesare. Esse sono, nell'ordine:

'Tarbelli', 'Bigerriones', 'Ptianii', 'Vocates', 'Tarusates', 'Elusates', 'Gates', 'Ausci', 'Garumni', 'Sibusates', 'Cocosates'.

Si tratta di un elenco di nomi di personaggi (preceduti nel testo dall'espressione "quo in numero fuerunt") che, evidentemente, non vengono nominati altrove nel testo.

Vediamo la sequenza di 11 hapax nei Promessi Sposi:

'error', 'conditio', 'votum', 'cognatio', 'crimen', 'cultus', 'disparitas', 'vis', 'ordo', 'ligamen', 'honestas'

Si tratta di una frase in latino all'interno di un testo in lingua italiana. Vi è una sequenza di 11 hapax consecutivi anche nel testo The jungle:

'biednam', 'matau', 'paskyre', 'teip', 'aukszcziausis', 'jog', 'vargt', 'ant','svieto', 'reik', 'vienam'

Si tratta di una misteriosa canzone in una lingua sconosciuta che viene intonata in un passaggio del libro.
Ecco dunque che tali sequenze di 10 o più hapax sembrano rispondere a necessità linguistiche piuttosto specifiche; ciò è prevedibile, se si pensa che una tale sequenza di parole non si serve della struttura grammaticale presente nel resto del testo. Per semplicità, da ora in avanti chiameremo elenchi le sequenze di almeno 10 hapax.
Andiamo, infine, a leggere i due elenchi presenti nel VMS. Il primo è il seguente:

'opcholdy', 'dfar', 'oeoldan', 'ytoaiin', 'yfain', 'okadar', 'qotoear', 'dchodar', 'ysaldal', 'ytodal'.

Esso si trova, non a caso, nel capitolo dedicato all'astronomia (quello che aveva presentato una più alta percentuale di hapax), precisamente nel foglio 67: tale elenco potrebbe essere la lista dei nomi degli astri raffigurati nella pagina (figura 2.9). Tale ipotesi è confermata da alcuni studi filologici che vedono in queste parole la sequenza dei 7 pianeti tolemaici [14].

Anche il secondo elenco si trova nella stessa sezione, giusto nella pagina seguente (il

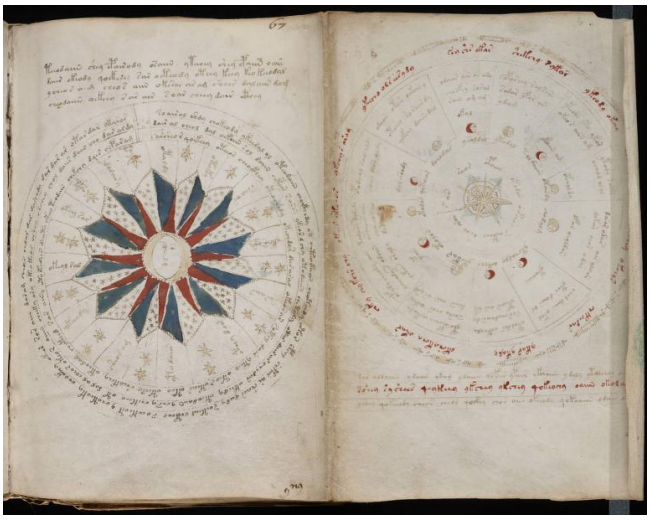


Figure 2.9: foglio 67 del VMS (fronte)

retro del foglio 67 - figura 2.10), in corrispondenza di quelle che sembrano proprio essere immagini di costellazioni:

'okaralet', 'olkao', 'okydseog', 'oafoaly', 'ocs', 'ekais', 'okolarm', 'qokoaiis',

'ocfhhy', 'dcsesor', 'ochepalain'.

Dunque ogni cosa lascia pensare a veri e propri elenchi di parole.
Questo risultato sembra confermare la presenza di un contenuto linguistico nel manoscritto; queste parole non sono state scelte in maniera casuale, ma potrebbero trovarsi lì per fornire una informazione.
Quello che possiamo concludere dalla nostra analisi, con una buona dose di certezza, è che tali hapax siano effettivamente nomi propri.

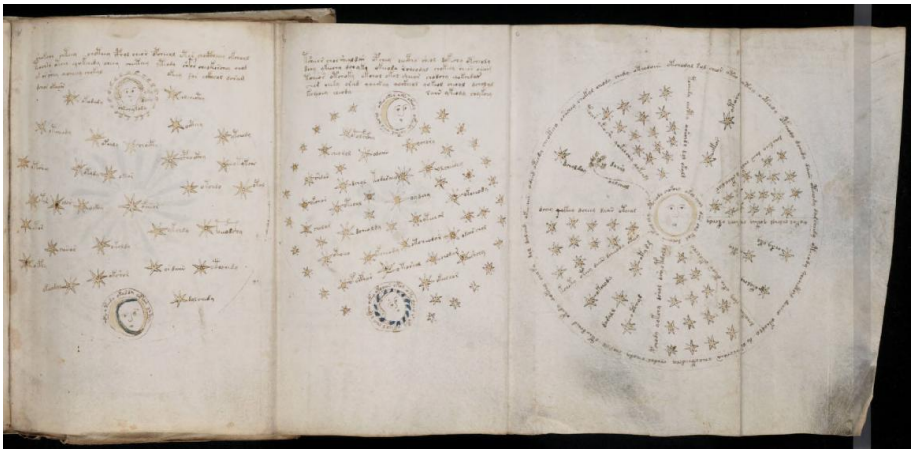


Figure 2.10: foglio 67 del VMS (retro)

2.5 Un network per la lingua del Voynich Manuscript

Proponiamo ora un grafo per il Voynich Manuscript, basato sulla Hamming distance tra due stringhe, estesa per adattarsi a un vocabolario con stringhe di lunghezze differenti.
Lo scopo è quello di fare un po' di ordine del vocabolario del manoscritto, classificando le parole in base alla loro somiglianza morfologica; in tal modo speriamo di poter fornire un aiuto nel difficile compito di distinguere le flessioni e le radici della lingua.
Dato un vocabolario, costruiamo un grafo G=(V,E) dove i nodi corrispondono ai vocaboli, ed esiste un legame tra due nodi w_i e w_j se e soltanto se possiamo ottenere w_i da w_j modificando un singolo carattere (i.e. la hamming distance tra w_i e w_j uguale a 1), o aggiungendo/togliendo un carattere all'inizio o alla fine di w_j [12].
A titolo esemplificativo, mostriamo in figura 2.11e 2.12 una componente connessa del network ottenuto a partire dal vocabolario del testo "divina". In figura 2.12 possiamo invece osservare una componente connessa del Network del VMS.

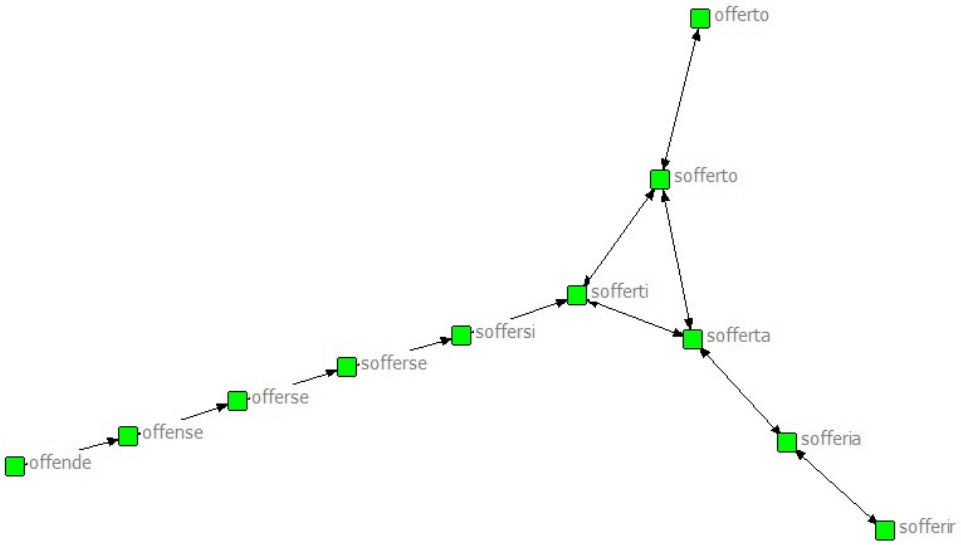


Figure 2.11: Una componente connessa del network tratto da "divina"

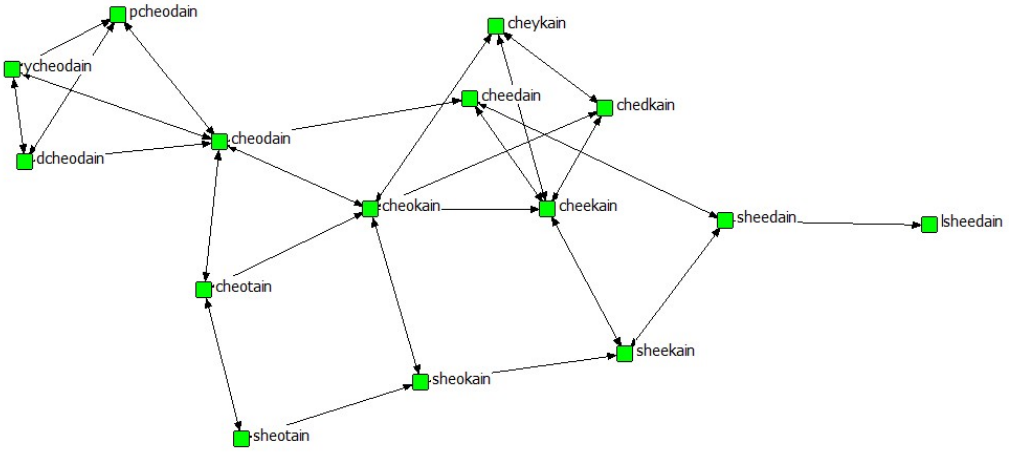


Figure 2.12: Una componente connessa del network del VMS

2.5.1 Confronto tra il network del VMS e quello di 3 testi in lingue naturali

Per prima cosa, vogliamo confrontare il network del VMS con quello ottenuto da testi in altre lingue naturali, andando a effettuare alcune misure statistiche elementari sui network stessi.
In particolare, consideriamo i seguenti testi: "aeneid" in lingua latina, "jungle" di lingua inglese, e "divina" di lingua italiana.

A partire dal vocabolario di questi testi potremmo già creare i rispettivi network, ma

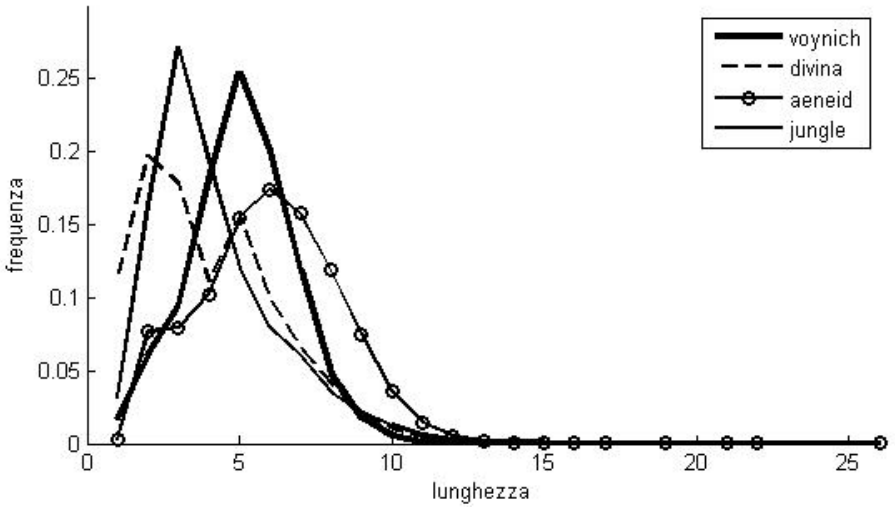


Figure 2.13: Distribuzione delle lunghezze dei vocaboli nei testi considerati

in tal modo le misure statistiche che faremo non sarebbero cofrontabili perché la dimensione dei vocabolari è differente. Dunque, abbiamo deciso di estrarre da ogni testo un vocabolario di lunghezza fissata, che corrisponde alla lunghezza del vocabolario del VMS: ovvero, 8065 vocaboli. Questi vocabolari non sono stati estratti a random; mano a mano che si percorre un testo, il numero di parole del suo dizionario cresce a una determinata velocità (legge di Heap). Abbiamo dunque ottenuto i vocabolari percorrendo il testo fino al raggiungimento della soglia stabilita.
Prima di effettuare delle misure su questi network, è necessario fare una considerazione preliminare: infatti, sappiamo che la distanza Hamming tra due stringhe dipende dalla lunghezza delle stringhe medesime (se a e b sono stringhe tali che $|a| = |b| = k$, allora la massima distanza tra a e b è k), e dunque la distribuzione delle lunghezze delle parole di una lingua può influenzare la topologia del network corrispondente. Analizzando in figura 2.13 queste distribuzioni per i vocabolari dei testi considerati, si osserva che la curva per il VMS si pone in una posizione intermedia tra le altre lingue. Questa con-

siderazione risulterà tanto più rilevante nel seguito, quando ci accorgeremo che questo network invece si distingue dagli altri, per maggior "connessione" e densità.
Effettuiamo dunque alcune misure statistiche di base: quantità di nodi non isolati (un nodo isolato è un nodo privo di connessioni), il numero delle componenti connesse e la cardinalità delle due componenti più grandi. Poiché siamo interessati a comprendere la struttura morfologica delle lingue in questione, una misura di densità può venirci in soccorso. La densità è infatti la misura di quanto un network sia distante dall'essere completo: un network ha densità 1 se ogni nodo è collegato ad ogni altro, e densità 0 se tutti i suoi nodi sono isolati. Precisamente, essa è il rapporto tra numero di connessioni effettive e numero di connessioni potenziali. Se M è il numero degli edges del network e N il numero dei nodi, allora la densità è:

$$\delta = \frac{M}{N * (N-1)/2}$$

Dunque se per assurdo il nostro network avesse densità uguale a 1, saremmo di fronte ad una lingua in cui la morfologia è banale: ogni parola si può ottenere dall'altra cambiando un singolo carattere. Invece quanto più sarà bassa la densità, tanto più la morfologia sarà complessa.
Presentiamo i risultati ottenuti sui network considerati (tabella 2.4): osserviamo, in sostanza, che il network del VMS è il più "connesso": pochi nodi isolati, poche componenti di cardinalità molto elevata e, infine, una densità che ha un'ordine di grandezza in più degli altri networks. D'altra parte, gli altri tre networks mostrano una grande somiglianza gli uni con gli altri, rispetto a queste statistiche: eppure si tratta li lingue naturali molto differenti. Inoltre vediamo che il network del testo "aeneid" è il meno denso: era proprio quello che ci aspettavamo osservando il grafico 2.13, dove appunto la curva del network di "aeneid" ha una coda lunga, e parole mediamente più lunghe degli altri network; invece non era affatto prevedibile che il network del VMS raggiungesse la densità più alta, trovandosi in una posizione intermedia nel grafico 2.13.
Cosa rende la lingua del VMS tanto diversa sotto l'aspetto morfologico e fonetico? Quello che ci dicono queste misure è che si tratta di una lingua con una morfologia molto elementare, persino ripetitiva, basata su poche radici e suoni che danno vita a numerose parole.

	n. of nodes	n.of non-isolated nodes	n.of connected components	main component length	second comp. length	density (δ)
voynich	8065	6613 (82,0%)	132	6064	106	$1,36 * 10^{-3}$
divina	8065	5224 (64,8%)	757	2865	26	$0,55 * 10^{-3}$
aeneid	8065	4261 (52,8%)	858	1636	26	$0,31 * 10^{-3}$
jungle	8065	4332 (53,7%)	792	2105	41	$0,41 * 10^{-3}$

Table 2.4: Statistiche dei networks

Per andare più a fondo nello studio dei networks, abbiamo calcolato la media e la varianza nella distribuzione dei degrees (tabella 2.5; ricordiamo che il degree di un nodo è il numero dei suoi edges). Queste misure ci consentono di riflettere sulla forma e compattezza del network: vediamo subito che il network del VMS ha un degree mediamente più elevato (circa il doppio rispetto alle altre reti), il cui valore massimo è davvero impressionante: 43 edges significa che ben 43 parole differenti della lingua si possono ottenere cambiando un solo carattere del nodo.

	voynichNET	**divinaNET**	**aeneidNET**	**jungleNET**
μ	6,69	3,45	2,39	3,11
σ^2	29,28	8,97	3,53	8,86
max	43	22	16	21

Table 2.5: Medie, varianze e massimo nella distribuzione del degree nei 4 networks

I nodi di degree più elevato sono per noi molto importanti: infatti, se è vero che il nostro network può individuare e raggruppare le principali "radici" morfologiche, allora questi gruppi saranno centrati sulle parole che hanno più alto degree, proprio in virtù della natura stessa del network. Dunque raccogliamo in una tabella la classifica dei primi 30 nodi del VMS, ordinati rispetto al degree (tabella 2.6).
A questo punto, se consideriamo il nodo w, con w una delle parole della classifica, possiamo estrarre il subnet formato da tutti quei nodi che hanno distanza 1 da w (tale subnet è detto EGONET di w): vediamo, ad esempio, l'EGONET della parola "chol" (figura 2.14). Quello che scopriamo guardando a tutti questi subnets è, appunto, che moltissime parole della lingua del VMS si ottengono semplicemente cambiando un solo carattere a w, o molto spesso aggiungendo un carattere all'inizio di w. Si noti a tal proposito la sequenza di parole:

tchol, cchol, fchol, rchol, lchol, pchol, ochol, schol, dchol, kchol

ottenute da "chol".
Se si è interessati particolarmente ad individuare tutti i casi in cui un simile fenomeno accade nella lingua, ovvero in cui la semplice aggiunta di una lettera ad un suffisso va a generare una numerosa serie di parole, è forse il caso di ricercare i k-cliques del network. Un k-cliques è un subnet completo di k nodi (per un algoritmo di estrazione delle cliques si veda [17]). Vediamo allora alcune cliques del network del VMS (figura 2.15 e figura 2.16): scorgiamo subito diverse radici della lingua, come **odaiin**, **chody**, **ain**, **sheo**,**oar**, **shor**.
Se proviamo ad estrarre questi k-cliques nei testi delle altre lingue non vediamo comparire mai parole così lunghe: si tratta di parole di 1 o due lettere, in cui è semplice ottenere

una parola dall'altra; oppure si tratta di numeri. Dunque una ipotesi potrebbe essere che questi cliques del voynich siano in realtà dei numeri, magari associando sequenze di più caratteri ad una singola cifra. Oppure si può ipotizzare che si tratti davvero di una lingua artificiale, con regole di composizione morfologica delle parole molto elementari.

nodo (parola)	degree
ol	43
chey	37
chedy	35
cheey	35
cheo	35
cheol	34
dal	33
oy	33
al	32
cho	32
chol	32
okar	32
or	32
dy	31
kar	31
okal	31
os	31
shey	31
cheor	30
dar	30

Table 2.6: Le prime 20 parole per degree, nel voynichNET

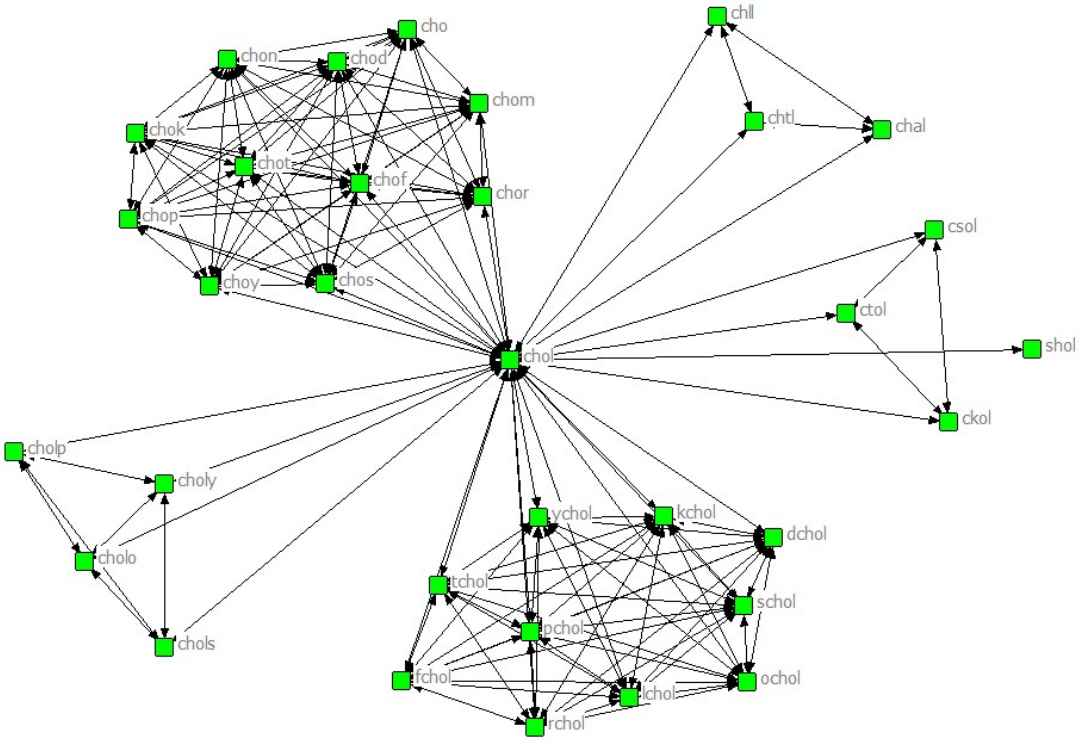


Figure 2.14: EGONET di 'chol'; in esso si possono osservare due cliques.

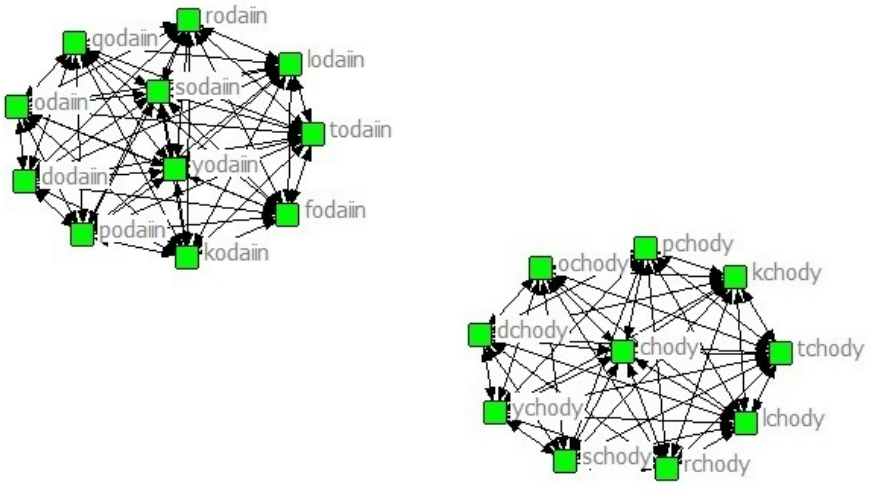
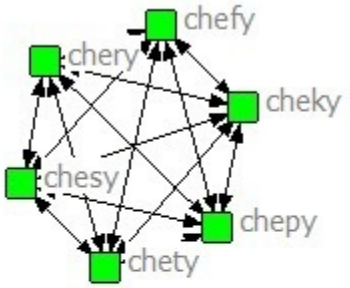
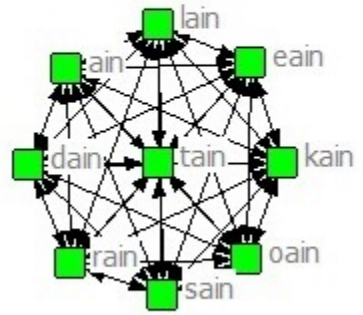
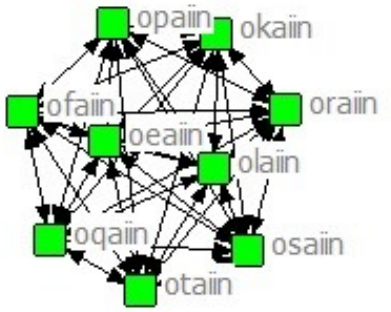
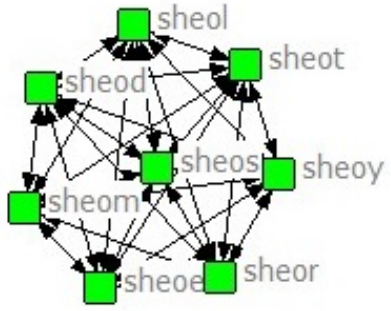


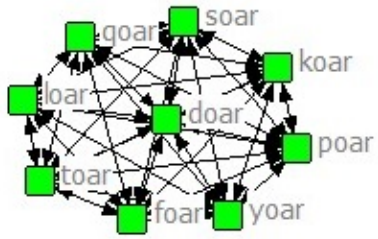
Figure 2.15: Alcuni 6-cliques e 7-cliques nel network del VMS











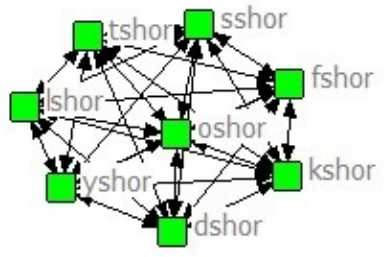


Figure 2.16: Alcuni cliques nel network del VMS

2.5.2 Correlazione tra degree e informazione nel network del VMS

Osservando l'elenco dei nodi di degree più elevato nel network del VMS, vediamo comparire alcune delle keywords del testo estratte nella sezione 2.2 (per esempio 'ol' e 'chol'). Dunque ci domandiamo se sussista una relazione tra il grado semantico delle parole e la loro centralità sotto il profilo morfologico/fonetico. Dunque se consideriamo per ogni parola w la coppia (E_w, D_w), dove la prima componente è il livello di informazione della parola w misurato dall'algoritmo MZ e la seconda ne rappresenta il degree nel nostro trasformation network, calcoliamo la correlazione tra queste componenti sulle parole del testo.
Vediamo i risultati sui testi considerati nelle nostre analisi (tabella 2.7): effettivamente il VMS si contraddistingue per un valore più alto dell'indice di correlazione.
In figura 2.17 è presentato il grafico degree/informazione per le parole del VMS: una certa correlazione è qualitativamente apprezzabile.

	voynich	**divina**	**aeneid**	**jungle**
$\rho_{X,Y}$	0,36	0,26	0,14	0,17

Table 2.7: correlazione tra la componente di informazione e quella del degree nelle parole dei testi considerati

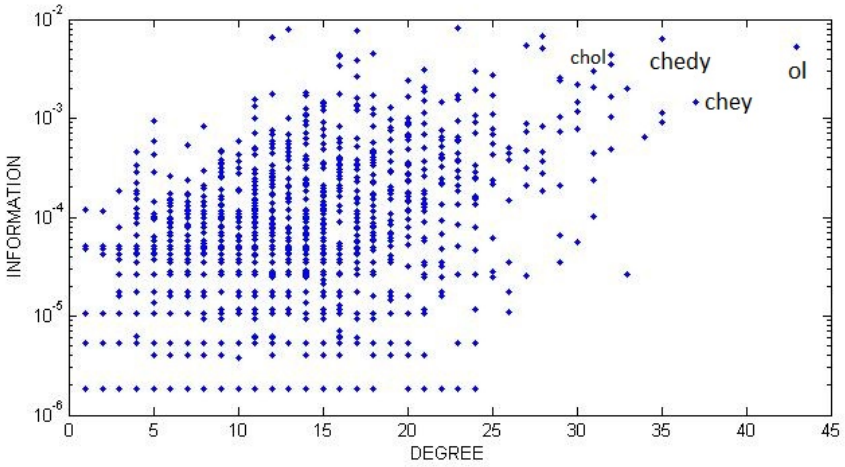


Figure 2.17: grafico degree/informazione per il VMS: i punti rappresentano le parole.

Chapter 3

Traduzione e informazione: alcune analisi su testi tradotti in diverse lingue

3.1 Machine translation e traduzioni "umane": curve dell'informazione a confronto

In questa sezione ci occuperemo di studiare come cambia il contenuto informativo di un testo, quando lo stesso è sottoposto a traduzione (nei casi che questa sia umana o automatica).
Consideriamo il testo "Il ritratto di Dorian Gray" di O.Wilde nella sua versione originale inglese e poi otto versioni del testo nelle seguenti lingue: polacco, finlandese, olandese, portoghese, spagnolo, tedesco, francese e italiano.
Per prima cosa vogliamo confrontare le curve dell'informazione di queste versioni; sappiamo che l'altezza delle curve dipende fortemente dalla lingua naturale del testo, tuttavia ci chiediamo se il massimo sia raggiunto alla stessa scala: infatti i testi raccontano le medesime vicende con i medesimi ritmi narrativi.
Nel grafico 3.1 possiamo osservare che, a parte l'eccezione della versione in Polacco (che raggiunge il massimo a circa 2000 parole), gli altri testi ottengono il massimo circa a 1000 parole. In tabella 3.1 possiamo poi verificare che le estrazioni di keywords, se non consideriamo il rumore creato da articoli, verbi, congiunzioni e preposizioni (rumore che può essere "pulito" utilizzando l'algoritmo MZ in combinazione con un POS-tagger - si veda il capitolo 1 della presente tesi), danno lo stesso risultato nelle varie lingue.
Consideriamo ora, in aggiunta alla versione originale inglese del testo, otto sue versioni tradotte mediante Google translate: si tratta di un software di traduzione automatica statistica, ovvero in cui le traduzioni vengono generate a partire da modelli statistici i cui parametri sono stimati da corpus bilingue. Riportiamo di seguito un frammento del

testo tradotto in italiano, così che sia possibile osservare la qualità della versione ottenuta.

"Lo studio è stato riempito con il ricco odore di rose, e quando la luce estate vento agitava tra gli alberi del giardino, ci è venuto attraverso la porta aperta il pesante profumo del lilla, o il più delicato profumo spina rosa fioritura."

Nel grafico 3.2 ripetiamo l'esperimento precedente, osservando ancora una volta che le curve raggiungono la massima informazione circa alla stessa scala; è possibile verificare che, anche in questo caso, le keywords corrispondono.
Se si osservano attentamente i due grafici 3.1 e 3.2 notiamo che, in quest'ultimo, le curve sono tendenzialmente più basse. Proviamo dunque a stampare i grafici, per ciascuna delle lingue considerate, della coppia di curve ottenute a partire dalla traduzione umana e automatica del testo in quella lingua (3.3). Osserviamo che, in ogni caso, la curva del testo tradotto automaticamente è più bassa, ovvero il suo contenuto informativo è minore.
Nel capitolo 1 avevamo osservato come la curva dell'informazione fosse invariante per lemmatizzazione del testo; ora vediamo che si perde informazione traducendolo automaticamente. In effetti, se, nel testo lemmatizzato, dobbiamo solo flettere opportunamente le parole per ricostruire un senso che resta comunque preservato, ora nella versione tradotta automaticamente dobbiamo colmare le lacune e correggere gli errori in maniera non banale, ovvero adoperando l'intuizione e non applicando semplici regole grammaticali. Possiamo dire che il grado di complessità del lavoro che dobbiamo svolgere per ricostruire il senso di un testo a partire da una sua versione trasformata, misura l'informazione persa mediante la trasformazione stessa. Dunque l'algoritmo MZ, di fronte a uno stesso testo su cui sono state operate trasformazioni linguistiche di qualsiasi tipo (lemmatizzazione, traduzione, o anche altre trasfomazioni che si possano ipotizzare, che mantengano più o meno invariata la lunghezza del testo), sembra poterci dire quanto lavoro sia necessario a ristabilire il contenuto del testo originale.

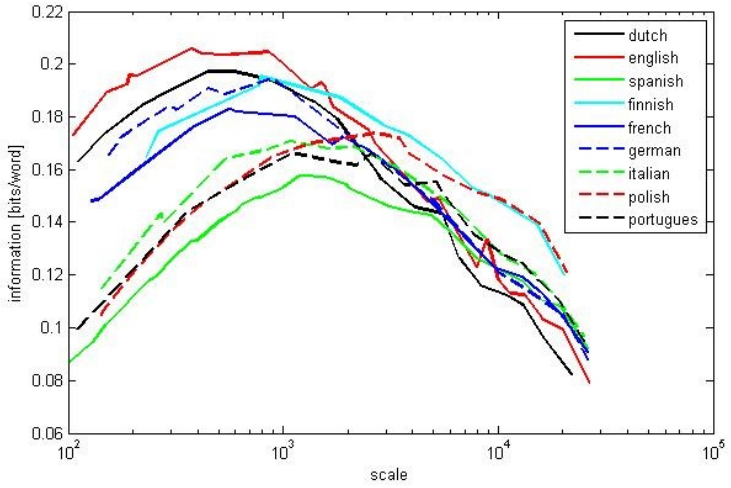


Figure 3.1: Curve di informazione per i testi "Il ritratto di Dorian Gray" in 9 lingue diverse.

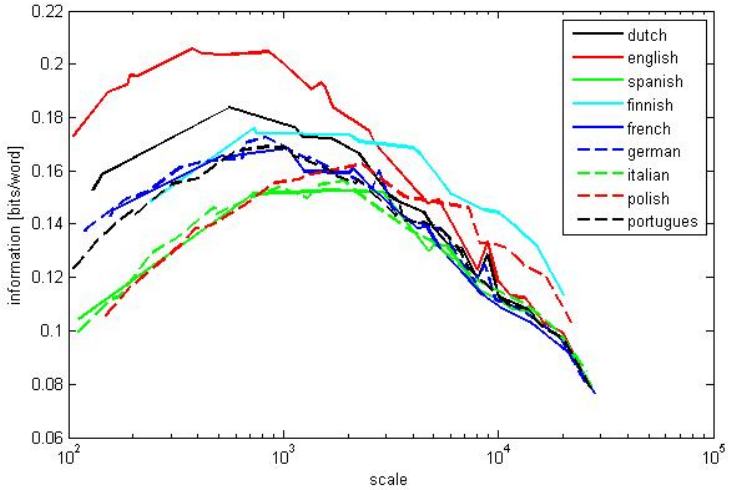
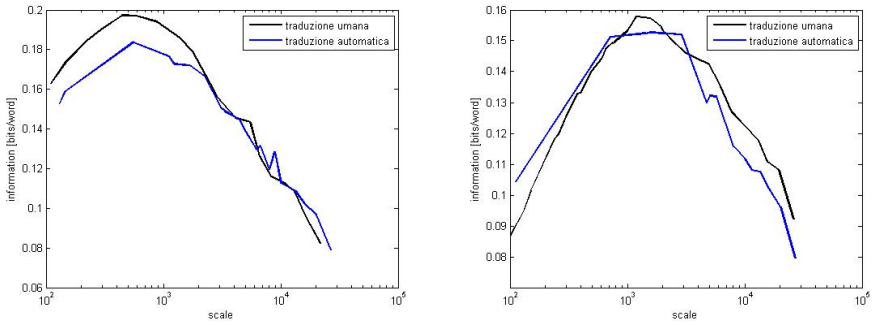


Figure 3.2: Curve di informazione per i testi "Il ritratto di Dorian Gray" in 9 lingue diverse, ottenuti con traduzioni automatiche.

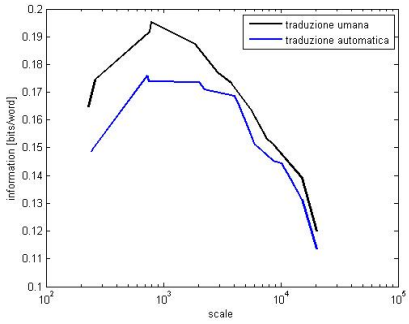
keys da spagnolo	keys da italiano	keys da francese	keys da olandese
me	mi	vous	je
te	è	je	ik
es	tu	elle	haar
habìa	non	il	zij
lord	era	nous	is
sibyl	aveva	des	was
no	**sybil**	me	u
usted	lei	avait	mij
de	**ritratto**	est	had
henry	**harry**	ne	zou
senor	**lord**	**lord**	hem
mi	**henry**	**harry**	mijn
su	ti	**sibyl**	de
harry	**basil**	**henry**	ze
duquesa	io	que	het
los	**dorian**	était	me
lo	**lady**	pas	zal
tu	**duchessa**	de	niet
retrato	**alan**	lui	**harry**
que	**signor**	m	**basil**
lady	mia	**portrait**	**henry**
le	me	**alan**	**portret**
y	avrebbe	**duchesse**	der
ha	voi	**basil**	**lord**
he	loro	la	ons
la	te	elles	bemìn
basil	di	votre	**sybil**
era	o	**lady**	den
o	vi	les	zijne
sus	sei	et	heb

Table 3.1: prime 30 keywords estratte in quattro lingue. In grassetto sono evidenziate quelle in comune (ignorando verbi, articoli, pronomi e congiunzioni).

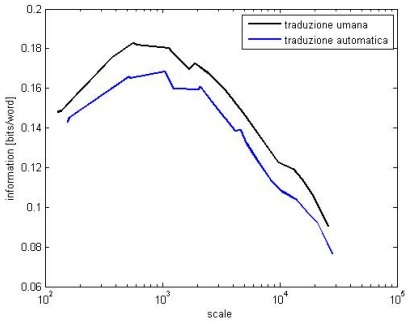


(a) olandese

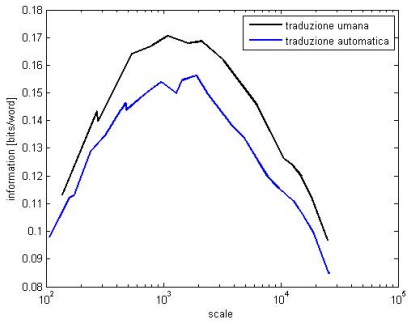
(b) spagnolo



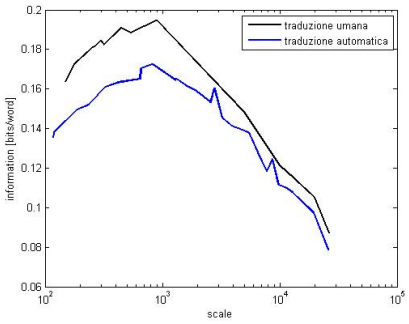
(c) finlandese



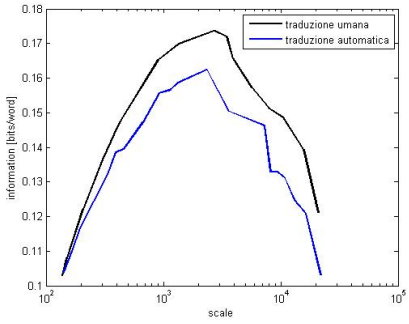
(d) francese



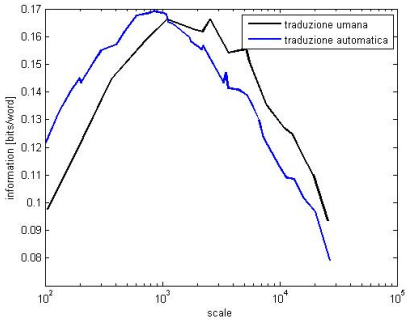
(e) italiano



(f) tedesco



(g) polacco



(h) portoghese

Figure 3.3: grafici scala/entropia in cui sono poste a confronto le traduzioni umane e automatiche del testo "il ritratto di Dorian Gray" nelle lingue considerate.

3.2 Machine translation: Analisi su sequenze di traduzioni

In questa sezione analizzeremo l'impatto della traduzione automatica utilizzando un indicatore diffente da quello di MZ.
Prenderemo come Corpus l'insieme della favole dei fratelli Grimm in lingua originale (inglese), disponibili in fortato elettronico sul sito "http://www.cs.cmu.edu/~spok/grimmtmp/".
Per prima cosa formalizziamo con una notazione utile l'operazione di traduzione:
Consideriamo l'insieme di indici $J = \{1, 2, ..., M\}$, dove ciascun indice rappresenta una delle M lingue che considereremo. Sia α una corrispondenza biunivoca tra gli indici:

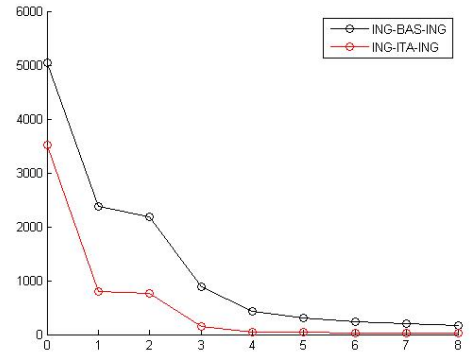
$$\alpha : J \longrightarrow J.$$

Allora $\alpha(i) = j$ significa che stiamo traducendo dalla lingua i alla lingua j. Invece se scriviamo $\alpha^{-1} \circ \alpha(i)$ indichiamo che stiamo traducendo un testo da una lingua $\alpha(i)$ ad una lingua i, dopo che questo è stato tradotto da una lingua i a una lingua $\alpha(i)$: dunque da un testo scritto in una determinata lingua, traducendo secondo $\alpha^{-1} \circ \alpha(i)$ otteniamo un nuovo testo scritto nella medesima lingua. Questo tipo di traduzioni le chiameremo "inverse".
Quello che faremo nella prossima sezione, sarà tradurre i testi del nostro corpus mediante $\alpha^{-1} \circ \alpha(i)$, e verificheremo quanto questi si modifichino: per un umano, esistono una quantità numerabile di traduzioni corrette di un libro, ma se si traduce in maniera automatica, sveleremo che esiste solo un grado limitato di elasticità, che va diminuendo mano a mano che si traduce e ritraduce un testo.
Per prima cosa definiamo una "distanza" tra i testi [5] che ci permetta di fare un'analisi opportuna dei nostri risultati:

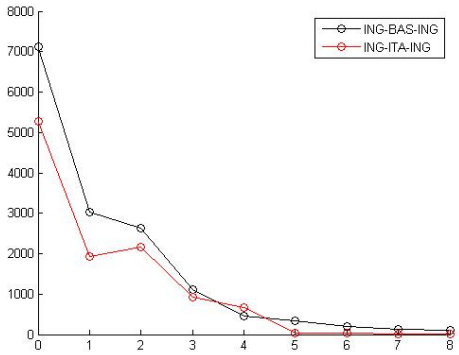
$$d_n^K(x, y) = \sum_{\omega \in D_n(y) \cup D_n(x))} \frac{(f_y(\omega) - f_x(\omega))^2}{(f_y(\omega) + f_x(\omega))^2}$$

Dove x e y sono due testi e $D_n(x)$,$D_n(x)$ sono i dizionari di n-grammi dei rispettivi testi, e f_x, f_y sono le frequenze nei rispettivi testi: per i nostri esperimenti utilizzeremo $n = 7$.
Osserviamo che si tratta in verità di una pseudo-distanza, in quanto non soddisfa la disuguaglianza triangolare; si è tuttavia dimostrata essere molto opportuna nello studio della similarità stilistica tra i testi ed un efficace strumento nella authorship attribution.
Per prima cosa consideriamo i testi del nostro corpus ed effettuiamo una traduzione "inversa"; partendo poi dai testi così ottenuti, effettuiamo una traduzione "inversa" di questi: ripetiamo una decina di volte questo procedimento e otterremo, per ciascun testo γ_i, una sequenza $\gamma_{i_1}, ..., \gamma_{i_{10}}$ di testi. Nella figura 3.4 vediamo i grafici della distanza tra γ_{i_j} e $\gamma_{i_{(j+1)}}$, al variare di j.
Quello che si può osservare è che la distanza tra due traduzioni inverse successive decresce. Il fatto che tale distanza vada a zero, significa che la traduzione perde elasticità, divenendo infine una mera relazione biunivoca tra due testi.

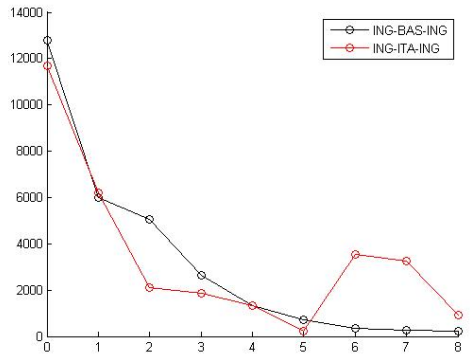
Osserviamo inoltre, che la traduzione inversa passante per l'italiano decresce più lentamente rispetto a quello passante per il basco; evidentemente la performance dell'algoritmo dipende dalla lingua in cui traduciamo. Se una traduzione è inefficace, vi è una perdita di informazioni (informazioni sintattiche, grammaticali, semantiche) e dunque quando si effettua una sequenza di traduzioni si arriva più rapidamente ad una stato di rigidità.



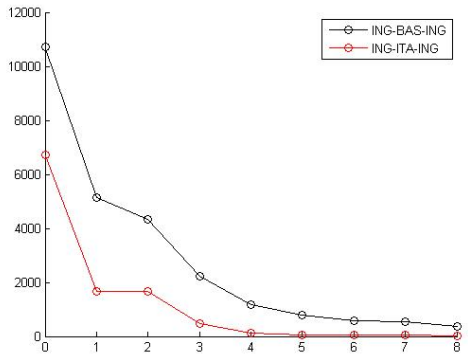
(a) grimm1



(b) grimm2

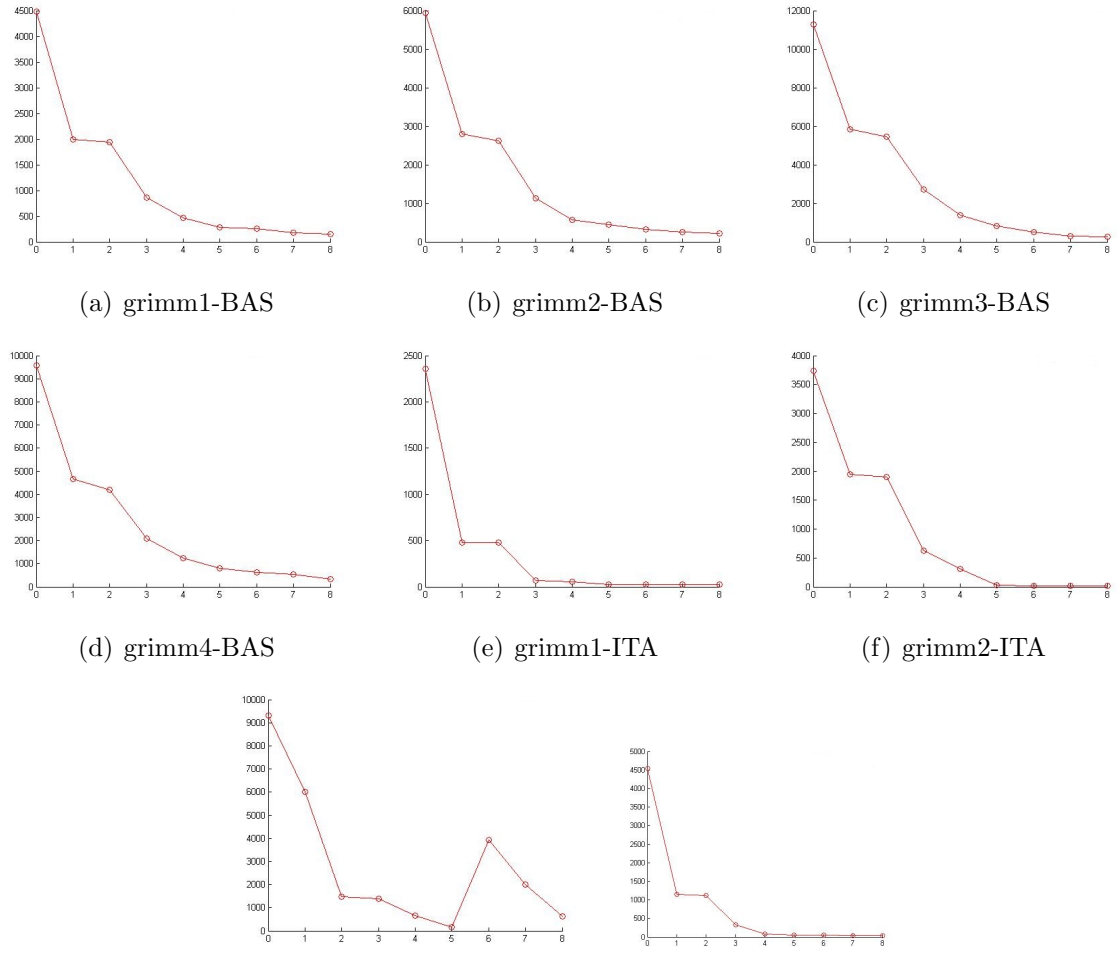


(c) grimm3



(d) grimm4

Figure 3.4: grafici di distanza testuale d_n^K tra step successivi di traduzione inversa; con le scritte $ING-BAS-ING$ e $ING-ITA-ING$ presenti nelle legende, indichiamo che ciascuna traduzione inversa è del tipo $\alpha^{-1} \circ \alpha(i)$, con i che rappresenta la lingua inglese (ING), e $\alpha(i) =: j$, che è il basco (BAS) o l'italiano (ITA).



(a) grimm1-BAS (b) grimm2-BAS (c) grimm3-BAS

(d) grimm4-BAS (e) grimm1-ITA (f) grimm2-ITA

(g) grimm3-ITA (h) grimm4-ITA

Figure 3.5: grafici di distanza testuale d_n^K tra step successivi di traduzione inversa; qui abbiamo considerato traduzioni del tipo $\alpha \circ \alpha^{-1}(i)$, con i la lingua italiana (ITA) oppure il basco (BAS), passando questa volta per la lingua inglese. Anche in questo caso vediamo generalmente una decrescita; si noti tuttavia che ciò non accade nel grafico (g).

Se fino ad ora abbiamo confrontato sequenze di testi tradotti e ritradotti in due lingue prestabilite i e $\alpha(i)$, ora, per confrontare l'efficacia delle traduzioni al variare della lingua, tradurremo i testi del nostro corpus mediante le traduzioni inverse $\alpha_1^{-1} \circ \alpha_1(i), ..., \alpha_{M-1}^{-1} \circ \alpha_{M-1}(i)$, facendo dunque variare la lingua j associata ad i: ovvero $\alpha_k(i) = j_k \neq j_{k+1} = \alpha_{k+1}(i), \forall k$. Ciò significa, in altre parole, che dato un testo del nostro corpus, lo tradurremo in ciascuna delle M lingue disponibili, e poi tradurremo i testi così ottenuti di nuovo in inglese.
Sia dunque γ un testo, e applichiamo il Hierarchical Clustering sui testi ottenuti mediante le traduzioni appena descritte (figura 3.6): Si osservi che quanto più le lingue sono diverse dall'inglese (lingua in cui è scritto γ), tanto più i testi ottenuti passando per una traduzioni in quelle lingue sono distanti dall'originale: ecco perché vediamo raggruppate insieme lingue molto simili per grammatica e vocabolario, mentre il latino, il cinese e il turco ci restituiscono testi in cui le strutture e i contenuti originali vengono stravolti. Facendo questo esperimento su diversi testi abbiamo sempre ottenuto risultati analoghi.

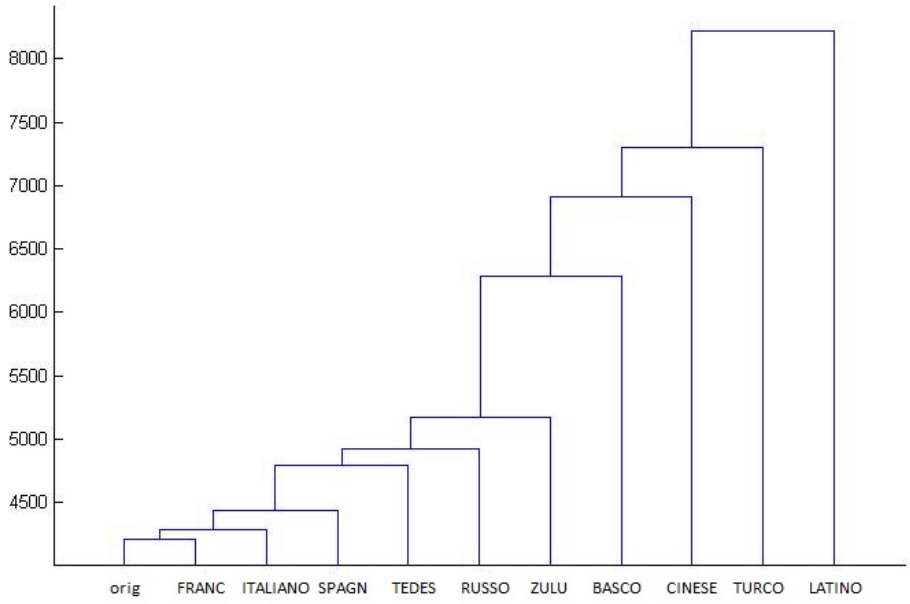


Figure 3.6: Hierarchical Clustering su distanze testuali (single linking), tra traduzioni inverse di grimm1 passando per 10 lingue diverse.

Appendix A

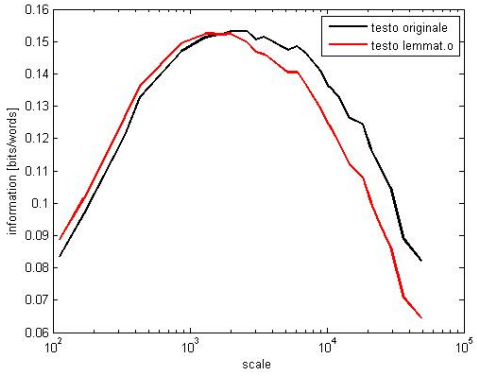
Invarianza per lemmatizzazione nella lingua italiana

In questa appendice vogliamo mostrare l'invarianza per lemmatizzazione dell'indice di informazione MZ, nel caso della lingua italiana; per farlo consideriamo un corpus composto da alcuni classici della nostra letteratura:

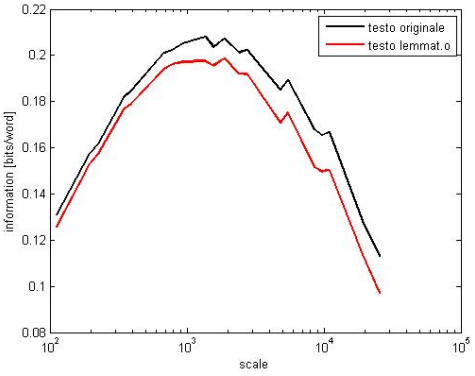
Short name	Title	text length	Vocabulary dimension	Vocabulary dimension (lemmatizzato)
mattia	Il fu Mattia Pascal	76831	10918	6621 (60,6%)
pinocchio	Le avventure di Pinocchio	40660	6005	3690 (61,4%)
saggia	Il saggiatore	84501	12854	9481 (73,7%)
sposi	I promessi sposi	223765	19603	10659 (54,3%)
zeno	La coscienza di Zeno	145487	13796	7023 (50,9%)

Table A.1: Corpus Γ. I testi elencati sono in lingua italiana.

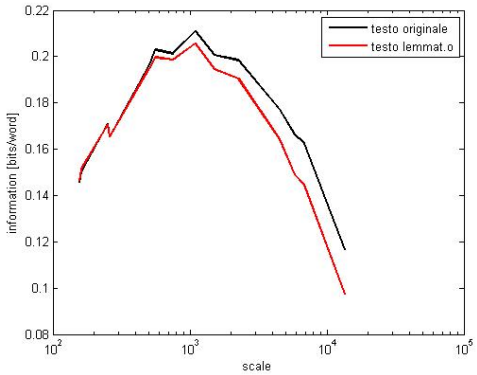
Nella tabella A.1 è indicata anche la cardinalità del vocabolario dei testi lemmatizzati: osserviamo che questo diminuisce di circa il 40 % rispetto a quello dei rispettivi testi originali, molto più che nella lingua inglese. Infatti l'italiano è una lingua molto più ricca di flessioni.
Nonostante, dunque, la lemmatizzazione in questa lingua consista in una trasformazione linguistica -per così dire- più invasiva, i grafici A.1 rivelano che la curva dell'informazione non risulta sensibilmente modificata, seppure le curve combacino meno rispetto allo stesso esperimento effettuato su testi in lingua inglese.



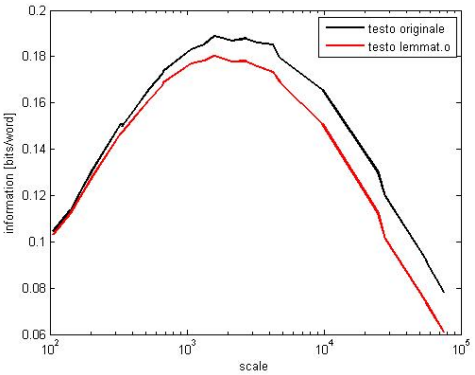
(a) zeno



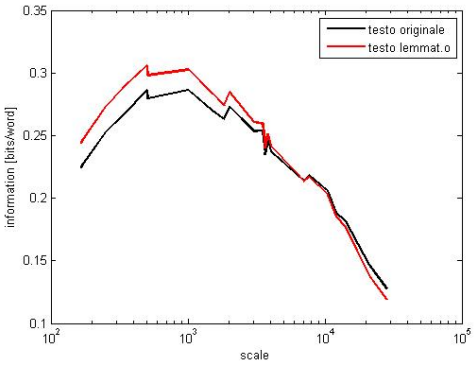
(b) mattia



(c) pinocchio



(d) sposi



(e) saggia

Figure A.1: grafici scala/entropia

Appendix B

Due algoritmi per il POS-tagging

Come è noto, una parte del discorso non dipende solo dalla parola stessa, ma anche dal contesto; dunque associare semplicemente ad ogni parola la sua categoria morfosintattica di maggior probabilità è un modo poco efficiente di fare POS-tagging. Nelle prossime sezioni illustreremo perciò due metodi stocastici, di cui uno basato sulle catene di Markov nascoste, e uno sugli alberi decisionali; in entrambi i casi, le probabilità saranno estratte mediante un approccio frequentista a partire da un corpus già annotato.

B.0.1 POS-tagging basato sui modelli di Markov nascosti(HMM)

I modelli di Markov nascosti (HMM) vengono utilizzati in numerose applicazioni nell'elaborazione del linguaggio naturale, come l'automatic speech recognition, il POS-tagging e altri pattern ricognicion [6][8]. Si tratta, sostanzialmente, di catene di Markov a cui si consente di emettere un "simbolo" ogni volta che si giunge in uno stato. Formalmente;
Sia $V = \{v_1, ..., v_n\}$ un insieme di simboli e S un insieme finito o numerabile di stati della catena di Markov con matrice di transizione A e distribuzione iniziale Π. Allora una HMM è una tripla $\Lambda = (A, B, \Pi)$ dove $(B)_{i,j}$ è la probabilità di generare il simbolo v_j quando si raggiunge lo stato s_i, ovvero $(B)_{i,j} = (P(y(t) = v_j | x(t) = s_i \in S)$, con $y(t) \in V$ è il simbolo generato al tempo t e $x(t) \in S$ è lo stato nascosto al tempo t.
In generale, quando si applica il modello delle HMM, si assume di conoscere il valore della sequenza dei simboli $y(0), ..., y(T)$ generati dal processo fino al tempo T, e si cerca la sequenza di stati che meglio la "spiega", ovvero quella che ne massimizza la probabilità; l'incognita è, dunque, $X^* = (x^*(0), ..., x^*(T))$ tale che

$$X^* := \underset{X}{\operatorname{argmax}} P(Y, X|\Lambda),$$

dove $X = (x(0), ..., x(T))$ è una qualsiasi sequenza di stati. A livello computazionale, X^* viene ottenuto mediante il noto algoritmo ricorsivo di Viterbi (si veda [6] per una

trattazione dettagliata dell'algoritmo).
A questo punto consideriamo un problema specifico, come quello di taggare per parti del discorso (POS-tagging) le parole di un testo di lingua naturale. Nel nostro caso, i simboli generati dal HMM sono le parole del testo (che indicheremo col simbolo w), mentre la sequenza di stati nascosta è proprio quella delle parti del discorso che indicheremo con τ.
Dunque, data una stringa di testo $w_1, ..., w_n$ il nostro compito si riduce a valutare:

$$T_{w_1,...,w_n} = \underset{\tau_1,...,\tau_n}{\text{argmax}} P(\tau_1, ..., \tau_n | w_1, ..., w_n) = (T.diBayes)$$

$$= \underset{\tau_1,...,\tau_n}{\text{argmax}} P(w_1, ..., w_n | \tau_1, ..., \tau_n) \frac{P(\tau_1, ..., \tau_n)}{P(w_1, ..., w_n)}$$

A questo punto possiamo eliminare il termine al denominatore, essendo costante per ogni sequenza di tags; otteniamo:

$$T_{w_1,...,w_n} = \underset{\tau_1,...,\tau_n}{\text{argmax}} P(w_1, ..., w_n | \tau_1, ..., \tau_n) P(\tau_1, ..., \tau_n) \tag{B.1}$$

Introduciamo ora delle ipotesi di Markov:

1. $P(w_i | w_1, ..., w_{i-1}, \tau_1, ..., \tau_i) = P(w_i | \tau_i)$
2. $P(\tau_i | w_1, ..., w_{i-1}, \tau_1, ..., \tau_{i-1}) = P(\tau_i | \tau_{i-2}, \tau_{i-1})$

Allora l'equazione (B.1) si traduce, applicando la chian rule, in:

$$T_{w_1,...,w_n} = \underset{\tau_1,...,\tau_n}{\text{argmax}} [\Pi_{i=1}^{n} P(w_i | \tau_i) \Pi_{i=1}^{n} P(\tau_i | \tau_{i-2}, \tau_{i-1})] \tag{B.2}$$

A questo punto si stimano le probabilità, e lo si fa a partire da un corpus annotato per parti del discorso, il quale può essere pensato come una sequenza di coppie (w, t) con w parola e τ il corretto tag associato a w nel suo contesto. Supponiamo che la dimensione (misurata in numero di coppie) del nostro corpus sia M, e che $f(w)$ e $f(\tau)$ siano, rispettivamente, le frequenze della parola w e del tag τ nel corpus. Allora calcoliamo le probabilità come segue:

1. $P(w, \tau) = \frac{f(w,\tau)}{f(\tau)}$
2. $P(\tau_i | \tau_{i-2}, \tau_{i-1}) = \frac{f(\tau_{i-2}, \tau_{i-1}, \tau_i)}{f(\tau_{i-2}, \tau_{i-1})}$

Nel caso $P(w|\tau)$ sia 0, ad esempio nel caso in cui w non sia presente o sia molto rara nel corpus annotato, si può porre:

$$P(w_i | \tau_i) = P(\tau_i | \tau_{i-2}, \tau_{i-1})$$

Infine si determina $T_{w_1,\ldots,w_n}$ utilizzando l'algoritmo di Viterbi.
Si noti che in una situazione reale $T_{w_1,\ldots,w_n}$ pone un problema numerico: può essere prodotto di quantità prossime allo 0; perciò -come per altro si tende a fare abitualmente quando si utilizza un modello stocastico- applichiamo il logaritmo, che essendo una funzione crescente non modifica l'argmax. In tal modo otteniamo:

$$T_{w_1,\ldots,w_n} = \underset{\tau_1,\ldots,\tau_n}{\operatorname{argmax}}[\sum_{i=1}^{n} \log(P(w_i|\tau_i) + \sum_{i=1}^{n} P(\tau_i|\tau_{i-2}, \tau_{i-1})] \tag{B.3}$$

Un POS-tagger basato su questo algoritmo tende a raggiungere un'accuratezza del 94 % circa.

B.0.2 POS-tagging basato su alberi decisionali (tree-tagger)

Come abbiamo accennato, uno dei principali problemi di un metodo probabilistico è che non riesce a gestire efficacemente il problema della sparsità dei dati; nel caso del linguaggio, per via delle legge di Zipf, tale problema è ben presente, e dunque se si dispone di un corpus di dimensioni limitate il POS-tagger non funzionerà bene.
Uno dei metodi che possono aggirare il problema è il Tree-tagger [7]; esso fa uso di un corpus annotato per parti del discorso per costruire un albero decisionale, che consente di ottenere buone stime delle probabilità di transizione.
Dunque, vogliamo determinare la probabilità che una parola abbia p.o.s. $\tau \in T$, con T l'insieme delle parti del discorso; allora consideriamo l'insieme dei trigrammi $[\tau, \tau_{-1}, \tau_{-2}]$, dove $\tau_{-1}, \tau_{-2} \in T$ sono le possibili p.o.s. che precedono τ e costruiamo l'albero decisionale procedendo ricorsivamente mediante dei test.
Ad esempio, un test è chiedersi se sia vero o no che τ_{-1} sia un nome; quindi si pone $\{\tau_{-1} = NN\}$ come primo nodo dell'albero, e si suddivide l'insieme dei trigrammi in quelli che verificano il test, e quelli che non lo verificano (analogamente anche l'albero è diviso in due sottoalberi). Si procede poi ricorsivamente con altri test τ_{-1} e τ_{-2} (si veda la figura B.1).
In generale, dunque, un test sarà una espressione della forma:

$$\tau_{-i} = \tau^*, i \in \{1, 2\}, \tau^* \in T.$$

La questione principale è stabilire, ad ogni passo, quale test scegliere.
Tra tutti i possibili test q, viene selezionato ad ogni iterazione quello che massimizza la nostra capacità di predire τ, ovvero quello che ci da la massima informazione su di esso. Dunque, se I_q è l'informazione che ci manca per conoscere τ una volta che è conosciuto

l'esito del test q, vogliamo trovare:

$$\underset{q}{\operatorname{argmin}} I_q = \underset{q}{\operatorname{argmin}} -P(C_+|C) \sum_{\tau \in T} P(\tau|C+) \log_2 P(t, C+) - P(C_-|C) \sum_{\tau \in T} P(\tau|C-) \log_2 P(t, C-) \tag{B.4}$$

dove C è il contesto al livello del nodo corrente (ovvero l'insieme dei trigrammi che soddisfano le condizioni dei nodi esplorando l'albero fino al nodo corrente) e C_+,C_- sono il contesto che risulta se il test ha risultato positivo e negativo rispettivamente; le probabilità sono stimate a partire dalle frequenze nel corpo annotato.
Per riassumere, dato l'insieme dei trigrammi, ad ogni iterazione vengono effettuati questi passi:

1. Si comparano tutti i possibili test q e si elegge quello che massimizza l'informazione su τ; questo test viene associato al nodo corrente.

2. Si espande l'albero all'altezza del nodo corrente nei due sottoalberi formati da quei trigrammi che soddisfano il test e che non lo soddisfano.

3. si verifica una condizione di arresto opportuna; ad esempio che la dimensione del più piccolo dei due sottoalberi superi una certa soglia minima.

4. infine si calcola la probabilità $P(\tau|C)$ dove C è il contesto corrispondente al nodo corrente; si associa questa probabilità al nodo corrente.

Infine, per stimare la probabilità di un dato trigramma, si segue il percorso corrispondente nell'albero fino al raggiungimento della foglia.

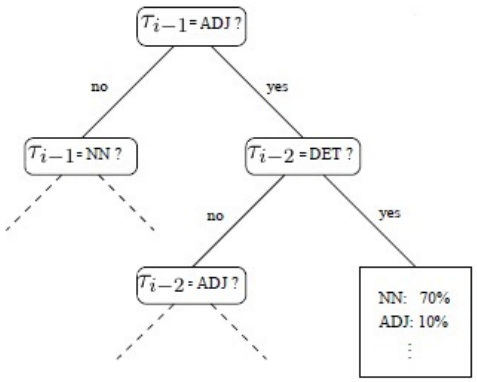


Figure B.1: esempio che illustra un ipotetico albero decisionale (tratto da [7])

Appendix C

Tecniche di clustering: una rassegna dei principali algoritmi e applicazione nei word spaces

Il clustering consiste nel raggruppare un insieme di dati in base a una distanza stabilita tra essi: può essere utile, nell'ambito dello studio del linguaggio, per formare clusters (gruppi) di parole che condividano significati o funzioni linguistiche. Infatti le parole di un testo possono essere rappresentate da vettori in uno spazio n-dimensionale, e possono dunque essere comparate e clusterizzate secondo una opportuna distanza.

C.1 Kmeans

il kmeans è un algoritmo di clustering iterativo non gerarchico, ovvero si basa sull'idea di trovare, iterativamente, una partizione ottimale del dataset in un numero di clusters prestabilito (k). Generalmente- ed è ciò che abbiamo fatto noi -come prima iterazione si considera una partizione random. Inoltre il kmeans è un algoritmo "forte", ossia assegna ad ogni vettore x_i uno e un solo cluster c_l di appartenenza. Precisamente:

$$c_l = \{x_i | d(x_i, \lambda_l) \leq d(x_i, \lambda_j), \forall j, 1 \leq j \leq k\} \tag{C.1}$$

dove d è la distanza usata -nel seguito adotteremo la distanza euclidea-, $x_1, ...x_m$ sono i vettori del dataset, mentre $\lambda_1, ..., \lambda_k$ sono i centroidi di ciascun cluster, ridefiniti ad ogni nuova iterazione come la media tra i vettori appartenenti al cluster considerato.
La complessità è O(n) e la convergenza del metodo si ha quando i clusters calcolati alla $i+1$-ma iterazione sono gli stessi che sono stati calcolati alla i-ma (in questo caso, convergenza al passo i): la convergenza tuttavia non è scontata, in quanto può accadere

che l'algortimo oscilli fra due soluzioni, dunque è oppurtuno aggiungere un controllo al fine di scongiurare il loop [6].

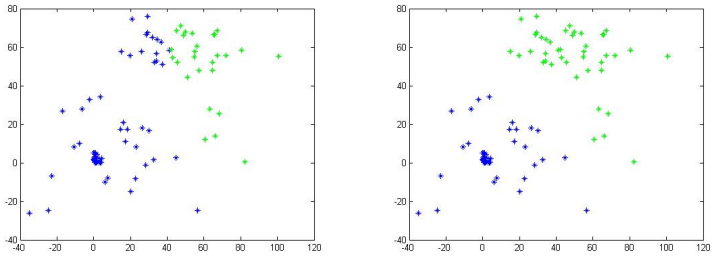
C.1.1 Esperimenti su dataset

Proviamo con un dataset di 100 numeri random che seguono tre diverse distribuzioni: due gaussiane di parametri

$$\mu_1 = (20, 13) \qquad \Sigma_1 = \begin{bmatrix} 900 & 0 \\ 0 & 400 \end{bmatrix}$$

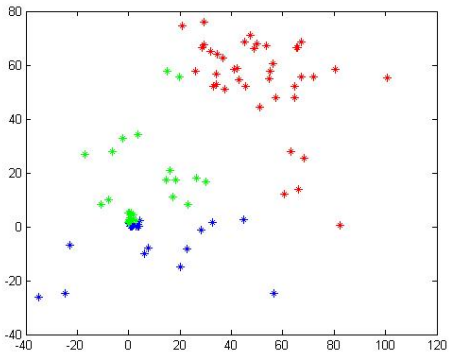
$$\mu_2 = (50, 60) \qquad \Sigma_2 = \begin{bmatrix} 324 & 0 \\ 0 & 49 \end{bmatrix}$$

e una esponenziale di parametro $\lambda = 0.5$.
Per prima cosa proviamo ad identificare due clusters (figura C.1): il risultato è soddisfacente, se consideriamo che i nostri dati si distribuiscono proprio su due aree dell'intervallo (in effetti la distribuzione esponenziale è centrata piuttosto vicino alla media di una delle due gaussiane, e dunque i suoi vettori ne sono stati inglobati, mentre la media dell'altra gaussiana è distante e la varianza bassa, dunque si formano due gruppi) che vengono riconosciute dal kmeans.
Ma vediamo cosa accade applicando nuovamente il kmeans, con k=3 (figura C.2): in questo caso vediamo che l'algoritmo genera un nuovo cluster che contiene alcuni dei vettori sparsi sulla destra (una coda della normale). Interessante osservare che, nonostante l'inizializzazione random abbia fornito alla prima iterazione una suddivisione davvero errata, l'algoritmo abbia raggiunto infine una soluzione soddisfacente, anche se con un maggior numero di iterazioni.

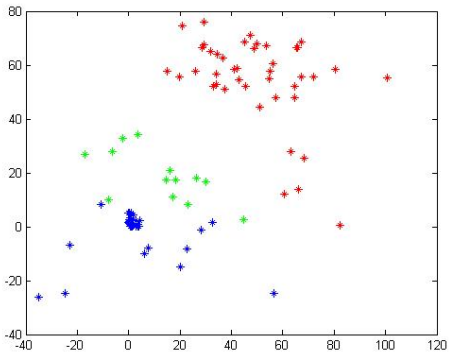


(a) iterazione 1 (b) iter 4: convergenza

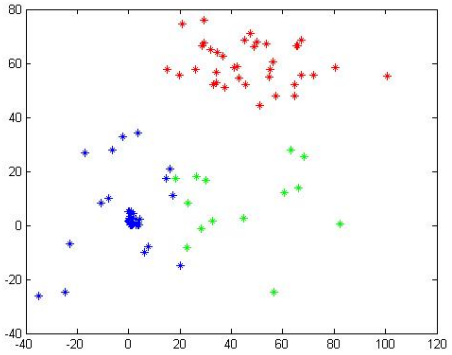
Figure C.1: kmeans con k=2



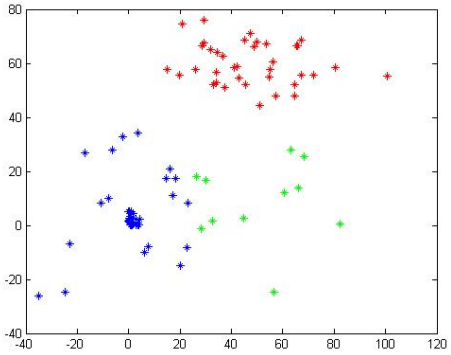
(a) iterazione 1



(b) iterazione 3



(c) iterazione 9



(d) iter 12: convergenza

Figure C.2: kmeans con k=3

C.2 EM

L'EM (Expectation-Maximization) è un algoritmo di clustering iterativo, di tipo non-gerarchico e debole: ciò significa che non associa ad ogni vettore un solo cluster di appartenza in maniera deterministica, bensì fornisce la probabilità del vettore di appartenere a ciascuno dei clusters.
Nella teoria questa idea è formalizzata supponendo l'esistenza di dati non osservabili z_j, le cui componenti z_{ji} assumono valore 1 se il dato j-esimo appartiene al cluster i-esimo. Assumiamo che i dati siano stati generati da un mix di distribuzioni gaussiane n-dimensionali, e si cerca di stimarne i parametri: i vettori media μ_i, le matrici di covarianza Σ_i e i coefficienti π_i:

$$p(x|\Theta) = \sum_{i=1}^{s} \pi_i p(x|\theta_i) \qquad \theta_i = (\mu_i, \Sigma_i)$$

Assumiamo che i vettori del set di dati siano indipendenti e siano generati dalla distribuzione $p(.)$ governata dai parametri di Θ. La probabilità che il set di dati X sia generato dal vettore di parametri Θ è:

$$p(\mathrm{X}|\Theta) = \prod_{j=1}^{N} p(x_j|\Theta) = L(\Theta|\mathrm{X})$$

$L(\Theta|\mathrm{X})$ è detta *Likelyhood*.

$$\Theta^* = T_{w_1,\ldots,w_n} = \underset{\Theta}{\mathrm{argmax}} L(\Theta|\mathrm{X})$$

L'algoritmo EM stima i parametri delle distribuzioni massimizzando la *Likelyhood*, più frequentemente la *Log-likelyhood* che agevola i conti da un punto di vista numerico.
Analizziamo più nel dettaglio i due step dell'iterazione dell'algoritmo:

E : calcola $h_{ij} = P(z_{ij}|\Theta) = \frac{P(x_i|n_j,\Theta)}{\sum_{k=1}^{s} P(x_i|n_k,\Theta)}$

M : determina il valore dei parametri che massimizzano h_{ij}

$$\mu_j = \frac{\sum_{i=1}^{n} h_{ij} x_i}{\sum_{i=1}^{n} h_{ij}}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} h_{ij}(x_i - \mu_j)^T(x_i - \mu_j)}{\sum_{i=1}^{n} h_{ij}}$$

$$\pi_j = \frac{\sum_{i=1}^{n} h_{ij}}{n}$$

Il vantaggio di basare la stima dei parametri sulla Likelihood è che l'algoritmo si dimostra sempre convergente anche se a volte non fornisce la il valore ottimale dei parametri perchè raggiunge un massimo locale.

C.2.1 Esperimenti su dataset

Abbiamo provato ad applicare l'algoritmo al nostro dataset misto: in questo caso notiamo subito che per raggiungere un risultato davvero ottimale è necessario fare più prove, affinché si ottenga un massimo locale più prossimo a quello globale: nella figura C.3(a) il risultato è meno soddisfacente rispetto a ciò che si vede nella (b) dove le nostre tre distribuzioni vengono riconosciute con notevole precisione (in figura C.4 rappresentazione con clusters di tipo forte). A dimostrazione di ciò, abbiamo osservato che i parametri restituiti dall'algoritmo approssimano bene quelli che abbiamo utilizzato per generare il dataset. Anche nel caso della distribuzione esponenziale, l'algoritmo ci restituisce una gaussiana che è piuttosto appropriata, avendo una media vicina allo zero e una varianza molto bassa.

parametri iniziali

$$\mu_1 = (20, 13) \qquad \Sigma_1 = \begin{bmatrix} 900 & 0 \\ 0 & 400 \end{bmatrix}$$

$$\mu_2 = (50, 60) \qquad \Sigma_2 = \begin{bmatrix} 324 & 0 \\ 0 & 49 \end{bmatrix}$$

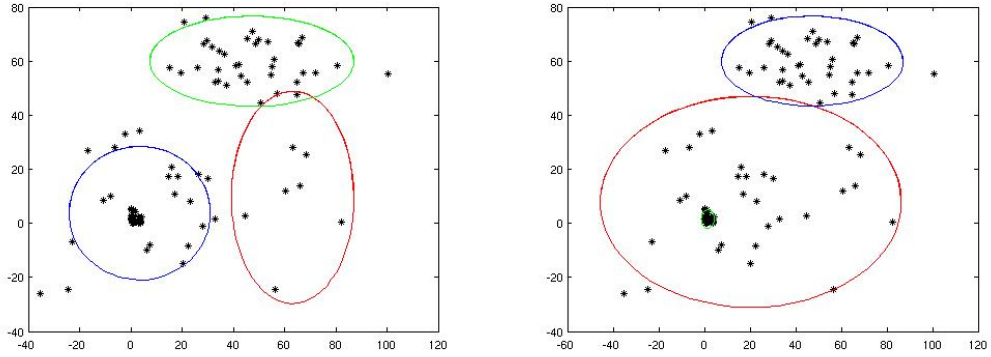
$$esponenziale, \lambda = 0.5.$$

parametri trovati

$$\mu_1 = (20.404, 7.967) \qquad \Sigma_1 = \begin{bmatrix} 899.204 & 101.533 \\ 101.533 & 316.462 \end{bmatrix}$$

$$\mu_2 = (47.241, 60.017) \qquad \Sigma_2 = \begin{bmatrix} 327.495 & -28.644 \\ -28.644 & 57.423 \end{bmatrix}$$

$$\mu_3 = (1.479, 1.611) \qquad \Sigma_3 = \begin{bmatrix} 1.682 & -0.665 \\ -0.665 & 2.319 \end{bmatrix}$$



(a) (b)

Figure C.3: EM con 3 clusters, due realizzazioni sul dataset misto

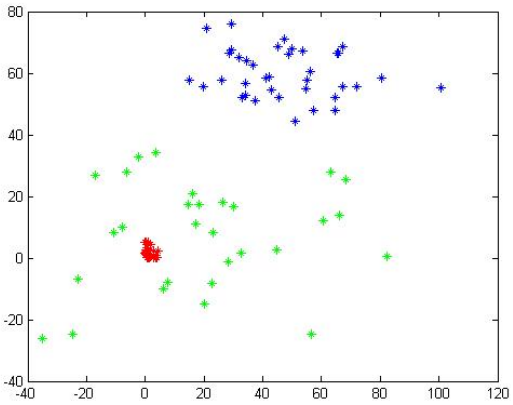


Figure C.4: figura C.3 (b) con clusters di tipo "forte"

C.3 Hierarchical clustering (bottom-up)

Gli algoritmi di tipo gerarchico hanno due caratteristiche principali: sono forti (formano classi tra loro disgiunte) e riconoscono una struttura, gerarchica, tra gli elementi all'interno di una classe.
Queste caratteristiche vengono lette graficamente con un dendrogramma ad albero, che ha come radice il cluster formato da tutti gli elementi e come foglie i clusters formati da singoli elementi.
L'algoritmo Bottom-Up è un esempio di clustering raggruppante, partendo dai singoli oggetti (le foglie dell'albero), l'algoritmo individua i due elementi maggiormente simili e li unisce in un unico cluster. Questo processo è iterato fino a che tutti gli elementi

appartengono ad un unico cluster.
Il concetto alla base dell'operazione di raggruppamento degli oggetti è la *similarità*. La similarità si può esprimere in termini di distanza:

$$sim(x, y) = \frac{1}{1 + d(x, y)}$$

con questa definizione riconduciamo la ricerca del massimo della similarità alla ricerca del minimo della distanza.
Il clustering gerarchico ha senso soltanto se la funzione di similarità che abbiamo definito è monotona, ovvero se dati C, C', C'' clusters vale:

$$min(sim(C, C'), sim(C, C'')) \geq sim(C, C' \cup C'')$$

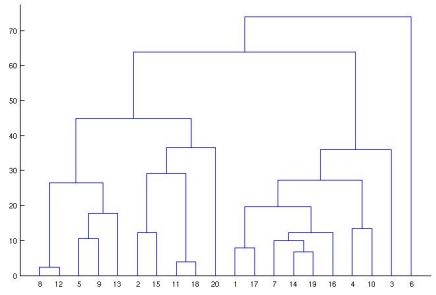
In effetti, se così non fosse, clusters che nell'albero si ritrovano lontani potrebbero divenire "simili" in raggruppamenti successivi, e l'albero stesso non sarebbe interpretabile.
Partendo dalla similarità tra oggetti, esistono diversi modi per estendere il concetto alla similarità tra clusters. Vediamone due molto usati:

- **single link**: $sim(C, C') = min_{x \in C\, y \in C'} sim(x, y)$
- **group average**: $sim(C, C') = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} sim(x_i, y_j)$

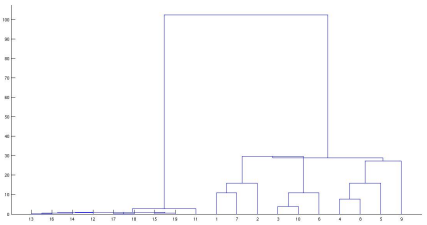
C.3.1 esperimenti su dataset

Abbiamo usato il group average su un dataset di 20 vettori random (distribuzione uniforme su [0,100]x[0,100]) e su un secondo dataset di 20 vettori, di cui 10 seguono la distribuzione uniforme nell'intervallo [50,100]x[50,100], e 10 la distribuzione esponenziale di parametro 1.

Comparando questi due dendrogrammi (figura C.6 (a) e (b)) ci accorgiamo che la loro forma è molto rappresentativa dei nostri set di dati: mentre nel primo (figura (a)) le distanze tra due clusters tendono ad aumentare linearmente al salire di livello nell'albero, il quale dunque mantiene una forma piuttosto compatta , invece nel secondo (figura (b)) notiamo una netta divisione in due clusters fondamentali, che possono essere visti a loro volta come due dendrogrammi di cui uno è più schiacciato (i dati sono vicini perché seguono una distribuzione esponenziale), e l'altro ha una forma riconducibile al dendrogramma random della figura 8(a).
Proviamo ora il single link sui medesimi set di dati (figura C.6).
Osserviamo che nel single link le distanze si accorciano: infatti l'altezza totale dell'albero è molto minore che nel group average. Inoltre il single linking tende ad accentuare le differenze esistenti tra i clusters nei primi livelli del dendrogramma (i.e. quelli più

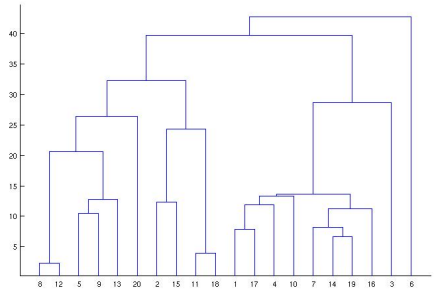


(a) dataset con distribuzione uniforme

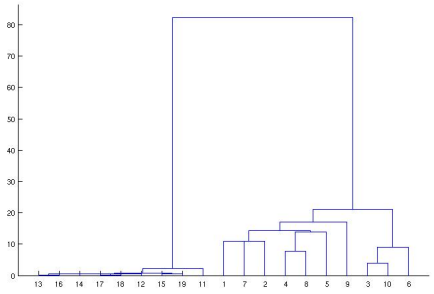


(b) dataset misto

Figure C.5: Hierarchical Clustering con Average link



(a) dataset con distribuzione uniforme



(b) dataset misto

Figure C.6: Hierarchical Clustering con Single link

prossimi alle foglie); si noti, infatti, che nella figura C.5(a) le distanze tra due clusters dei primi livelli dell'albero sono spesso simili tra loro, mentre sono più eterogenee nell'albero con single link in figura C.6(a).

C.4 Word spaces

Un word space è un modello in cui le parole vengono rappresentate come vettori in uno spazio n-dimensionale [6]. Tale spazio vuole rappresentare idealmente il luogo dei significati delle parole; non avrebbe però senso costruire word spaces se non assumessimo la cosiddetta "ipotesi distribuzionale", che afferma:

"Words with similar distributional properties have similar meanings." (Z.Harris)

Secondo Harris, dunque, la similarità semantica tra due parole è una funzione del grado di similarità del loro "ambiente linguistico", ovvero il grado in cui co-occorrono in con-

testi simili. Questa posizione è espressa in maniera molto concreta da Kilgarriff, che sostenne: "Where 'word senses' have a role to play in a scientific vocabulary, they are to be construed as abstractions over clusters of word usages".
Si possono riconoscere sostanzialmente due tipi di relazioni semantiche tra le parole: paradigmatiche e sintagmatiche.

- Le relazioni paradigmatiche riguardano parole che non co-occorrono nel testo e sono dette relazioni "in absencia": si hanno relazioni paradigmatiche tra due parole se queste sono "intercambiabili", ovvero se svolgono la stessa funzione linguistica (ad esempio articoli, verbi ecc...). Un modello distribuzionale ottenuto mediante una matrice di occorrenza parola per parola (WxW), ovvero in cui si contano per ciascuna parola le occorrenze in un intervallo [-a,a], contiene questo tipo di relazioni. (un esempio di relazione paradigmatica è quella tra le parole "you" e "we")

- Le relazioni sintagmatiche sono invece relazioni "in praesentia", ovvero dove le parole sono relazionate a tutte quelle che sono presenti nel medesimo contesto. Un modello distribuzionale ottenuto mediante una matrice parola per documento (WxD), ovvero in cui -dato un corpus di documenti- si contano le occorrenze di ogni parola in ciascun documento, contiene questo tipo di relazioni (un esempio è la relazione che lega "albero" a "bosco").

C.5 Cluster analysis sulle matrici di co-occorrenza delle keywords

In questa sezione vogliamo sfruttare i pattern di occorrenza delle parole chiave estratte mediante l'algoritmo MZ, e applicare poi algoritmi di clustering al fine di identificare la presenza e l'intensità delle relazioni esistenti tra queste parole.
La principale metrica per gli word spaces è quella data dalla distanza coseno:

$$d(x, y) = 1 - \frac{|xy|}{xy} = 1 - \cos\theta_{x,y}$$

In questa tesi, salvo diversamente specificato, ogni volta che faremo clustering adottiamo sempre questa distanza. Inoltre facciamo esperimenti solo su relazioni di tipo sintagmatico, utilizzando matrici del tipo WxD.
Proviamo a considerare come livello testuale quello dei nostri P intervalli ottimali dell'algoritmo MZ mediante i quali abbiamo estratto le keywords: nel caso del testo "darwin", per esempio, questi hanno una lunghezza di circa 2500 parole; se associamo a ciascuna delle prime 30 keywords un vettore normalizzato di occorrenza sugli intervalli, possiamo effettuare un kmeans per ottenere 3 clusters; vediamo comparire un risultato di questo tipo:

cluster A: {hybrids, fertility, sterility, pollen}
cluster B: {species, varieties, forms, genera, characters, groups, rudimentary}
cluster C: {the, of, in , and, to, islands, will, selection, a, be, i, we, breeds, seeds, have, plants, on ,bees, that}

Per prima cosa osserviamo che gli elementi che compongono i gruppi A e B hanno tra loro una evidente similarità semantica: si osservi per esempio la presenza di "fertility", "sterility" e "pollen" nel cluster A, o la presenza di diverse categorie di classificazione tipiche delle scienze naturali nel cluster B. Invece il terzo cluster è molto ricco di elementi e non può fornirci troppa informazione; in effetti ancora una volta ci ritroviamo di fronte al problema di una estrazione in cui sono presenti categorie morfosintattiche diverse dai nomi.
Proviamo, comunque (soprattutto a scopo illustrativo), a fare un hierarchical clustering, così da intendere la struttura delle relazioni sui diversi livelli di un dendrogramma (figura C.7). Si noti che parole come "will","he","i",we" (che si trovavano tutte nel cluster C trovato col kmeans), hanno tra loro relazioni fortissime; questi termini, come pure

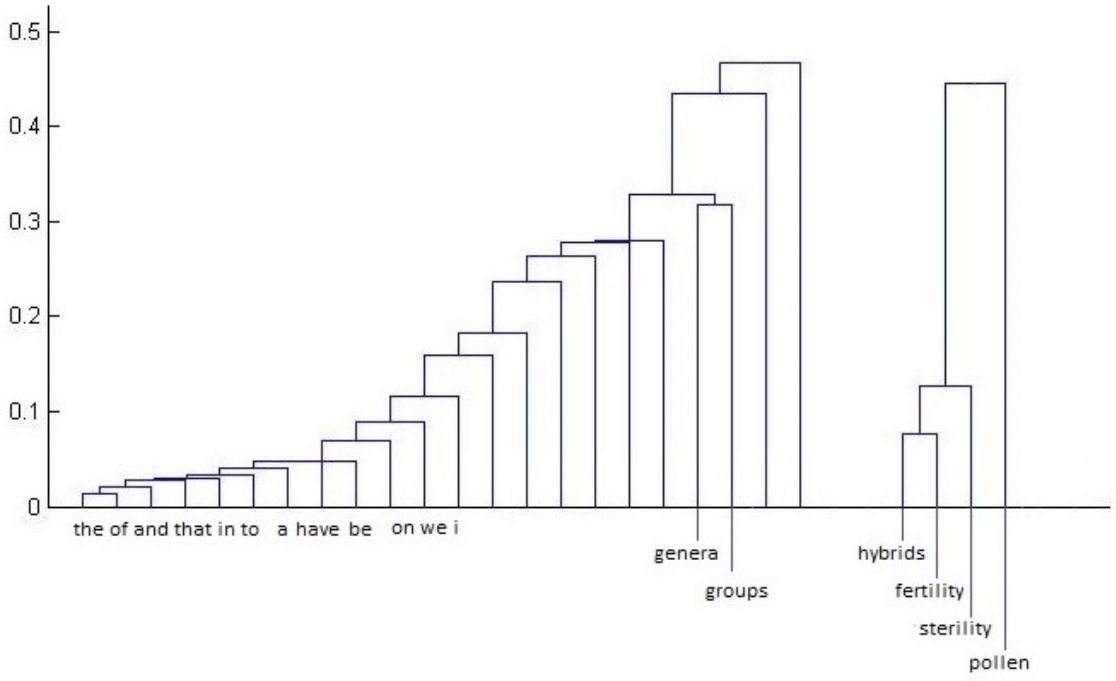


Figure C.7: hierarchical clustering sulle prime 30 keywords estratte da "darwin"

'the','of', 'and','in,'a' e 'that', pur avendo un alto valore informativo secondo l'indice di MZ, hanno in comune il fatto di essere molto frequenti nel testo, dunque non troviamo componenti nulle nei vettori che li rappresentano; probabilmente è questa la ragione per cui hanno la similarità più alta nel dendrogramma. Gli altri gruppi che si possono intravedere nell'albero sono proprio quelli riconosciuti precedentemente.

Bibliography

[1] D.R.Amancio, E.G.Altmann, D.Rybski, O.N.Oliveira, L.F.Costa, "Probing the statistical properties of unknown texts: application to the Voynich Manuscript"

[2] Montemurro, Zanette, "Towards the quantification of the semantic information encoded in written language", Advances in Complex Systems, Vol. 13, No. 2 (2010) 135–153 World Scientific Publishing Company

[3] Montemurro, Zanette, "The statistics of meaning: Darwin, Gibbon and Moby Dick", Significance, 12(2009), 165-169

[4] Montemurro, Zanette, "Quantifying the information in the long-range order of words", Cortex 55 (2014) 5-16

[5] C.Basile, D.Benedetto, E.Caglioti, M.D.Esposti, "An example of mathematical authorship attribution", Journal of Mathematical Physics 49, 125211 (2008)

[6] Manning, Shutze, "Fundations of statistical NLP", The MIT Press (2000)

[7] H.Schmid, "Probabilistic POS-tagging using decision trees", Proceedings of International Conference on New Methods in Language Processing, 1994, Manchester (UK)

[8] Appunti delle lezioni di F.Tamburini (Università di Bologna) su word-space models

[9] Montemurro MA, Zanette DH (2013) Keywords and Co-Occurrence Patterns in the Voynich Manuscript: An Information-Theoretic Analysis. PLoS ONE 8(6): e66344.

[10] Cover and Thomas, "Elements of Information Theory", 2nd edn. (J.Wiley, Hoboken, N.J., 2006)

[11] Schinner, Andreas(2007) 'The Voynich Manuscript: Evidence of the Hoax Hypothesis', Cryptologia, 31: 2, 95 — 107

[12] V.Batagelj, A.Mrvar, M.Zaversnik, "Network analysis of texts", University of Ljubljana, Inst. of Mathematics, Physics and Mechanics, Department of Theoretical Computer Science, 2002

[13] K.J.Goh, A.L.Barabàsi, "Burstiness and memory in complex systems", EPL, 81 (2008), 48002

[14] http://voynich.freie-literatur.de/index.php

[15] http://www.voynich.com/pages/index.htm

[16] G.Landini, "Evidence of linguistic structure in the VMS using spectral analysis" (2001) Cryptologia, 25:4, 275-295,

[17] P.M.Pardalos, "An exact algorithm for th maximum cluque problem", Operation Research Letters 9 (1990) 375-382