Università di Roma "La Sapienza"
Istituto Italiano di Studi Orientali
Dottorato in Civiltà Culture e Società dell'Asia e dell'Africa
Curriculum civiltà islamica: storia e filologia
XXV ciclo

Ḥadīṯ E ANALISI COMPUTAZIONALE

Strategie e strumenti per il trattamento automatico del testo arabo e l'estrazione d'informazione

Tesi di dottorato

Candidato:
Marco Boella

Tutore:
Prof. Giuliano Lancioni

Università di Roma "La Sapienza"
Istituto Italiano di Studi Orientali
Dottorato in Civiltà Culture e Società dell'Asia e dell'Africa
Curriculum civiltà islamica: storia e filologia
XXV ciclo

Ḥadīṯ E ANALISI COMPUTAZIONALE

Strategie e strumenti per il trattamento automatico del testo arabo e l'estrazione d'informazione

Tesi di dottorato

Candidato:
Marco Boella

Tutore:
Prof. Giuliano Lancioni

“Do or do not...
there is no try”
Yoda

Note sulla traslitterazione

Se inserito nel corpo del paragrafo, il testo arabo è riportato in traslitterazione. Nel caso di citazioni che costituiscono un paragrafo a sé il testo arabo può comparire in caratteri originari, in traslitterazione semplice oppure in traslitterazione computazionale.

La traslitterazione semplice utilizza le corrispondenze della tabella successiva, con le seguenti osservazioni: (1) la *hamza* in inizio di parola è omessa; (2) la ة è resa 'a' in forma pausale, a meno che l'intero contesto sia vocalizzato.

أ ء ئ ؤ ’	د d	ض ḍ	ك k
ب b	ذ ḏ	ط ṭ	ل l
ت t	ر t	ظ ẓ	م m
ث ṯ	ز ṯ	ع ‘	ن n
ج ǧ	س s	غ ġ	ه h
ح ḥ	ش š	ف f	و w
خ ḫ	ص ṣ	ق q	ي y

La traslitterazione computazionale rende il testo arabo così come presente nei listati originali dei programmi, nel testo di input e nel processamento informatico dei dati e corrisponde in modo rigoroso alla relativa grafia araba. La tabella di conversione è reperibile al titolo 3.5.2, insieme a un maggiore dettaglio di informazioni.

Elenco delle abbreviazioni

ad es.	ad esempio
ar.	arabo
cfr.	Confrontare
EI	Enciclopedia dell'Islam
EF	Espressioni Funzionali (per i *ḥadīṯ*)
ingl.	inglese
NLP	Elaborazione del linguaggio naturale (*Natural Language Processing*)
TA	Traduzione Automatica
TM	Memoria di traduzione (*Translation Memory*)
TO	Testo di origine (per la traduzione)
TT	Testo tradotto
qc.	Qualcosa
qn.	Qualcuno

Indice dei contenuti

0. Introduzione

0.1 Le *Digital Humanities*: utilità o *divertissement*?

A partire dagli anni '90 ha cominciato ad affermarsi una nuova area di studi conosciuta sotto il nome di *Digital Humanities*[1]. Il carattere marcatamente interdisciplinare e l'ampio ventaglio di situazioni alle quali questa espressione può essere applicata, ma anche l'incertezza su ciò che vi possa rientrare o ne sia escluso, rendono difficile una definizione univoca ed esaustiva, e forse il motivo è nell'operazione stessa di fusione di due settori disciplinari piuttosto distinti come le scienze informatiche e quelle umanistiche.

Prive di una definizione precisa su obiettivi e strumenti e senza un chiaro statuto disciplinare, le scienze umane digitali contrassegnano comunemente fenomeni tra i più diversi, dall'edizione digitale di un testo letterario all'organizzazione on-line di una biblioteca, dall'analisi di un soggetto umanistico con tecniche computazionali alla presentazione multimediale di un museo di arte contemporanea.

Nel tentativo di dare una definizione chiara e coerente, Kirschenbaum (2010, p. 2) suggerisce di consultare la relativa voce inglese su Wikipedia[2], che recita:

---

[1] In italiano si trovano anche le diciture 'informatica umanistica' (Celentano, Cortesi, & Mastandrea, 2004) 'scienze umane digitali', che nel diverso uso e successione delle parole che le compongono concentrano maggiormente l'attenzione su una disciplina o sull'altra Cfr. ad es. http://www.e-lib.ch/it/Attualita?d=201302 [consultato il 20 febbraio 2013].

[2] L'utilizzo in questo contesto della definizione di Wikipedia, non nasce tanto dall'ovvio motivo della facilità di reperimento in linea rispetto ad altre fonti, bensì dalla considerazione che, soprattutto per i termini relativi a discipline di studio e concetti scientifici (o comunque definizioni che non comportano necessariamente visioni personali, politiche, religiose), la 'lessicografia partecipata' offre due vantaggi: l'aggiornamento cronologico della definizione (e con esso il minore rischio di diventare vetusta) e la costruzione di un certo 'consenso' sulla definizione del termine stesso da parte della comunità scientifica

> "The digital humanities, also known as humanities computing, is a field of study, research, teaching, and invention concerned with the intersection of computing and the disciplines of the humanities. It is methodological by nature and interdisciplinary in scope. It involves investigation, analysis, synthesis and presentation of information in electronic form. It studies how these media affect the disciplines in which they are used, and what these disciplines have to contribute to our knowledge of computing." (Kirschenbaum, 2010, p. 2).

Kirschenbaum scriveva però nel 2010. Oggi, a tre anni di distanza, la definizione 'wikipediana', per sua natura estremamente volatile se messa alla prova dal tempo, dice invece:

> The digital humanities is an area of research, teaching, and creation concerned with the intersection of computing and the disciplines of the humanities. Developing from the field of humanities computing, digital humanities embraces a variety of topics ranging from curating online collections to data mining large cultural data sets. Digital Humanities currently incorporates both digitized and born-digital materials and combines the methodologies from the traditional humanities disciplines (such as history, philosophy, linguistics, literature, art, archaeology, music, and cultural studies), as well as social sciences, with tools provided by computing (such as data visualisation, information retrieval, data mining, statistics, text mining) and digital publishing. (Digital Humanities, voce di Wikipedia (tratto da http://en.wikipedia.org/wiki/Digital_humanities, [21 febbraio 2013]).

Secondo queste definizioni, è comunque evidente che le scienze umane digitali sono principalmente una prospettiva di metodo, un modo diverso e interdisciplinare di affrontare i soggetti delle tradizionali delle discipline umanistiche, piuttosto che una vera e propria scienza costituita e autonoma (Svensson, 2009). L'uso sistematico e quasi imprescindibile delle risorse digitali nel campo delle scienze umane ne è probabilmente l'aspetto quantitativamente più visibile e rilevante, a testimonianza dell'importanza che

(quasi a evocare, con un po' di scherzoso azzardo, l'*ijmā'* della giurisprudenza islamica). Poiché infatti le voci di Wikipedia sono liberamente modificabili dagli utenti e la traccia cronologica dei cambiamenti rimane associata al nome di chi li ha proposti, il fatto che certe definizioni sono perfezionate da alcuni tra i massimi studiosi del campo si verifica piuttosto frequentemente, non solo per i termini specifici quanto per quelli più generali quali le definizioni disciplinari, appunto.

l'elaborazione e l'organizzazione computazionale dei dati oggi hanno nella definizione di molti approcci metodologici 'umanistici'.

Se spostiamo l'attenzione da chi può utilmente fruire nel proprio lavoro di ricerca degli strumenti messi a disposizione dalle scienze umane digitali, a chi questi strumenti progetta ed elabora, quale potrebbe essere il suo profilo ideale? Chi è in altre parole l'attore principale di questa prospettiva metodologica, l'umanista digitale o l'informatico umanista? Le scienze umane digitali sono investigate e percorse da umanisti con competenze digitali o da informatici con interessi umanistici? La domanda è provocatoria e non ha probabilmente risposta, ma può servire ad affrontare una delle questioni principali di questa area interdisciplinare, vale a dire qual è lo scopo ultimo di integrare tecniche computazione e metodi di analisi tradizionale? Giova di più alle scienze umanistiche o a quelle informatiche o a entrambe? Un testo digitalizzato ed elaborato ad esempio con tecniche di analisi morfologica e semantica, è poi effettivamente utile per chi si occupa di ricerche testuali o letterarie o rappresenta piuttosto un semplice terreno per la verifica e l'applicazione di modelli teorici informatici?

Ciò molto dipende dalle motivazioni e dal bilanciamento di competenze di chi compie questo tipo di ricerche. Avere un'eccellente competenza sia in una disciplina umanistica sia nelle teorie informatiche e tecniche di programmazione e computazione è piuttosto raro. La maggior parte degli umanisti, di là delle competenze anche avanzate necessarie per utilizzare il computer e le sue varie applicazioni, difficilmente avrà le conoscenze informatiche per progettare e realizzare da sé i propri strumenti di analisi computazionale. Questi sono solitamente realizzati da informatici e programmatori con una formazione scientifica di tipo matematico-computazionale, che considerano le scienze umanistiche un campo come un altro per l'applicazione di propri modelli e la realizzazione di programmi e applicativi. Alla luce di queste considerazioni, chi si occupa di studi umanistici spesso considera le tecnologie informatiche essenzialmente come qualcosa di cui fruire nello stadio di elaborazione finale, un servizio già pronto ma anche già definito e limitato nei suoi scopi e possibilità di applicazione.

Questo pone un problema: il bisogno e la soddisfazione di un bisogno vengono da mondi diversi: l'umanistica si separa nuovamente dall'informatica, e chi deve proporre soluzioni informatiche, ad esempio per l'analisi di un testo, non può necessariamente avere l'ampiezza di visione del ricercatore umanistico, che conosce sicuramente in modo più approfondito l'oggetto di studio a cui proprio quegli strumenti si dovranno applicare.

Da un punto di vista informatico, ciò che conta è il modo di organizzare efficientemente i dati, gli algoritmi e i processi di elaborazione. Per un informatico trattare numeri, lettere o stringhe di testo è concettualmente uguale. Dal punto di vista di uno studioso di discipline umanistiche invece, ciò che conta è se il risultato faccia o meno progredire la ricerca, vale a dire quanto di nuovo e utile si può ottenere grazie alla strategia computazionale. Il processo informatico con cui lo si ottiene spesso non è ugualmente interessante per il ricercatore.

La questione chiave, comune a tutti i processi interdisciplinari è quindi, come fare a ottenere un certo equilibrio nell'interazione delle due discipline, in modo che gli strumenti costruiti siano efficacemente adeguati all'oggetto di ricerca ma abbiano soprattutto l'elasticità e l'apertura necessaria per trattare con argomenti umanistici, che per definizione, non hanno la rigorosa struttura della scienze fisiche, biologiche, matematiche.

Nella prefazione alla sua *Encyclopedia of Canonical Ḥadīth*, Gautier Juynboll a un certo punto parla della voluminosissima opera *Tuḥfa al-ashrāf bi-ma'rifa al-aṭrāf* di Yūsuf bin 'Abd al-Raḥmān al-Mizzī (m.1341 d.C.), una compilazione di tutti gli *isnād* contenuti nelle sei collezioni canoniche di *ḥadīṯ*, e scrive "I embarked up on reading all the thirteen volumes of the work" (Juynboll, 2007). Affrontare la lettura personale e integrale di un'immensa quantità di testo è ancora l'unico modo possibile per acquisire risorse utili per la propria ricerca?

Il senso di scegliere per questa tesi l'approccio metodologico tipico delle scienze umanistiche digitali è forse racchiuso in un'altra domanda: il computer ovviamente non può sostituire la complessità del ragionamento umano, ma può quel computer arrivare a fare qualcosa che lo studioso umanistico di per sé non è in grado di fare o impiegherebbe un'immensa quantità di

tempo e risorse per farlo? E soprattutto, possono essere i risultati elaborati in modo computazionale, davvero utili per far progredire la ricerca umanistica in un determinato settore oltre che per dimostrare la sofisticazione dell'approccio informatico?

0.2 La linguistica computazionale e l'elaborazione del linguaggio naturale

La linguistica computazionale è un settore interdisciplinare di studi che tratta della rappresentazione del linguaggio naturale attraverso modelli basati o su regole linguistiche o su algoritmi di tipo empirico-statistico.

Seppure in un panorama che vede la lingua inglese predominare sia come oggetto di studio che come mezzo di espressione scientifico dei risultati, la fondazione della linguistica computazionale può essere fatta risalire all'opera di uno studioso gesuita italiano, Roberto Busa, che nel 1949 inizia un pionieristico progetto di digitalizzazione e ipertestualizzazione dell'*Index Thomisticus* di Tommaso d'Aquino, che verrà completato solo nel 1980 con la pubblicazione dell'intero corpus di 56 volumi (Busa, 1974).

La linguistica computazionale oggi è una disciplina strutturata e pienamente matura, e si occupa dell'analisi e della proposizione di modelli astratti che possano rendere conto dei vari livelli della linguistica, in particolare quelli morfologico e sintattico e quello semantico-lessicale. Tali rappresentazioni vengono poi elaborate attraverso algoritmi e applicazioni informatiche che le traducono per permetterne l'applicazione ai registri orali e scritti delle lingue naturali.

Sovente le espressioni 'linguistica computazionale' ed 'elaborazione del linguaggio naturale' (*Natural Language Processing*, NLP) sono usati come sinonimi, anche se le tecniche di NLP prevedono un'interazione più stretta tra informatica, intelligenza artificiale e linguaggi naturali umani, senza necessariamente considerare le componenti linguistiche al centro del processo di elaborazione come avviene invece nel campo della linguistica computazionale. Due paradigmi caratterizzano le applicazioni di NLP, a seconda del grado con cui :

- ❖ riproducono alcuni fenomeni linguistici in relazione alla necessità di trasformarli o convertirli;
- ❖ riconoscono o analizzano l'input linguistico allo scopo di generare o sintetizzare nuovo output linguistico.

È quindi possibile applicare una categorizzazione bidimensionale ai principali settori di applicazione delle NLP, secondo gli assi di riproduzione-trasformazione e riconoscimento-generazione (cfr. figura 0.1).

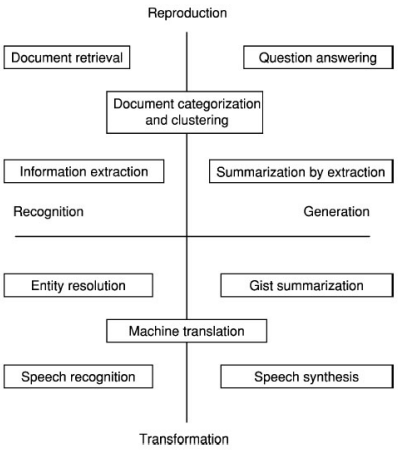


Figura 0.1: Principali applicazioni di NLP.
Fonte: (Jackson & Schilder, 2005, p. 505).

Secondo questa classificazione, tra le principali applicazioni di NLP possiamo quindi comprendere (Jackson & Schilder, 2005):

- ❖ categorizzazione di documenti (*document clustering*): applicazioni che cercando di ordinare i documenti secondo categorie e somiglianze basate sulla lingua dei documenti stessi.
- ❖ traduzione automatica (*machine translation*): è un processo di trasformazione che coinvolge sia il riconoscimento (della lingua da cui tradurre) che la generazione (nella lingua in cui tradurre),

cercando come obiettivo principale di eliminare l'ambiguità della produzione di più risultati concorrenti (Cardie & Mooney, 1999);

- estrazione di informazione (*information retrieval/extraction*): ha come scopo principale quello di localizzare, filtrare ed estrarre informazioni da un dato testo o da una collezione di essi. è il processo alla base del funzionamento ad esempio di tutti i principali motori di ricerca in linea (Baeza-Yates & Ribeiro-Neto, 1999);
- risposta a domande (*question answering*): prevede la generazione di risposte specifiche a domande inserite in input. Simile all'estrazione di documenti, restituisce però in output una sintesi d'informazioni provenienti da documenti diversi (Hirschman & Gaizauskas, 2001);
- riepilogo testuale (*summarization*): a seconda delle strategie utilizzate può prevedere la semplice ricomposizione di alcune sottounità testuali ritenute rilevanti oppure la generazione di nuovo riassunto testuale sulla base delle informazioni estratte dal testo originario (Mani & Maybury, 1997);
- estrazioni di *named entities:* prevede l'identificazione in un testo dei nomi indicanti referenti unici nella realtà esterna senza perdita di informazione o importazione di materiale estraneo (Rau, 1991);
- riconoscimento e sintesi vocale (*speech recognition/synthesis*): concerne le strategie utilizzate per la conversione dal registro orale a quello scritto e viceversa (Gibbon, Moore, & Winski, 1997).

Al di fuori di questa classificazione, possono esser elencati ancora alcuni settori ascritti alla linguistica computazionale e aventi come obiettivo principale se non esclusivo l'analisi e la modellazione linguistica. Questi settori possono poi essere a loro volta impiegati nello sviluppo di applicazioni NLP:

- linguistica dei *corpora*: prevede la costituzione di basi di dati organizzate e formalizzate fondate sulla collezione e digitalizzazione di ampie quantità di testi reali, che possono essere dotati di diversi gradi di annotazione linguistica (Hunston, 2005).

- ❖ semantica computazionale: consiste nella definizione di sistemi logici adeguati alla rappresentazione dei significati linguistici e alla identificazione delle reti di relazioni tra essi (Eijck & Unger, 2010).
- ❖ modellizzazione computazionale della grammatica: è lo sviluppo di modelli e teorie grammaticali che siano traducibili in linguaggio macchina e siano in grado di analizzare e/o generare le strutture morfo-sintattiche e semantiche di una lingua naturale (Baldridge & Kruijff, 2007).
- ❖ sviluppo di analizzatori sintattici (*parsing*): è l'analisi delle strutture sintattiche di una lingua viene compiuta attraverso la tecniche di computazione che impiegano grammatiche formali di regole oppure sfruttando metodi a base statistica basati su collezioni di frasi annotate sintatticamente (Grune & Jacobs, 1990).
- ❖ sviluppo di lemmatizzatori e analizzatori morfologici (*tokenizers, lemmatizers)*: si concentra sulle componenti morfologiche delle lingue naturali per la realizzazione di analizzatori in grado di riconoscere e segmentare i sotto-elementi costituenti di parola (Al-Sughaiyer & Al-Kharashi, 2004; Roark & Sproat, 2007);
- ❖ simulazione dell'evoluzione diacronica delle lingue e classificazione tipologica.

Si può infine notare come lo sviluppo delle linee di ricerca della linguistica computazionale sia fortemente asimmetrico, e dipenda essenzialmente da due fattori:

- ❖ la preponderanza di studi di linguistica computazionale e NLP riguarda le lingue europee e in misura particolare l'inglese. Ancora oggi, le altre famiglie linguistiche sono rappresentate in misura molto inferiore, sia dal punto di vista della letteratura sia da quello dello sviluppo di strumenti di analisi;
- ❖ l'enorme sviluppo di alcuni settori NLP rispetto ad altri più marcatamente linguistici dipende anche dalla possibilità di applicarli in ambiti non accademici ed extralinguistici. Molte delle ricerche

di estrazione di informazione sono favorite e finanziate da compagnie private che forniscono servizi di indicizzazione e reperimento risorse on-line, ad esempio Google Inc. Le ricerche su traduzione e riepilogo automatici possono disporre di notevoli risorse fornite oltre che da privati, anche da agenzie governative di *intelligence* e sicurezza nazionale, come il caso della Defense Advanced Research Projects Agency (DARPA) negli USA, che finanzia ambiziosi programmi di linguistica computazionale per il recupero di informazioni e la traduzione automatica multilingua da e verso il numero più alto possibile di lingue naturali.

0.3 Un approccio interdisciplinare allo studio del testo arabo

Nell'ambito delle teorie e delle applicazioni di NLP per il trattamento del testo, questo lavoro affronta l'analisi di una tipologia specifica di testi in lingua araba cercando di impiegare un ampio spettro di strategie computazionali di diversa natura. I testi scelti per l'analisi appartengono alle raccolte di *ḥadīṯ*[3], le tradizioni in lingua araba relative al profeta dell'Islam.

Le ragioni della scelta dei *ḥadīṯ* sono essenzialmente due:

- lo stato dell'arte delle tecniche NLP per l'arabo privilegia quasi esclusivamente l'approccio a testi contemporanei in Arabo Moderno Standard. Scegliere invece un testo fondante della storia culturale e religiosa ma anche linguistica del mondo arabo-islamico può costituire in potenza un'innovazione metodologica e fornire forse un modesto contributo allo sviluppo degli studi critici e letterari di settore;
- Le raccolte di *ḥadīṯ* presentano un grado relativamente esemplare di organizzazione e strutturazione interna del testo che ne consentono l'interpretazione computazionale e l'estrazione di informazione;

[3] Per alcune note relative all'uso del termine in questa dissertazione, cfr. titolo 2.1.1.

Il testo dei *ḥadīṯ*, non costituisce semplicemente un input come un altro per la sperimentazione dell'efficacia degli strumenti di analisi, ma diventa la chiave di volta di un approccio che tenta di definire gli strumenti di analisi a partire e in funzione delle caratteristiche interne del testo stesso. Il testo diventa quindi la ragione principale per cui certi strumenti vengono sviluppati, anche se essi mantengono alcuni tratti di trasferibilità e universalità che ne permettono l'utilizzo anche per altre tipologie testuali. Lo scopo del modello è di tendere a essere fattivamente interdisciplinare, attraverso il coinvolgimento a pari livello delle ragioni del testo letterario da una parte e dei metodi di trattamento computazionale dall'altra.

Il modello presenta le seguenti caratteristiche generali:

- modularità: il sistema è costruito come un insieme di moduli indipendenti in grado di operare un certo numero di elaborazioni sul testo al fine di ottenere progressivamente un'analisi completa ma sfaccettata;
- automaticità: ogni parte del modello è concepita cercando il più possibile di eliminare l'intervento manuale in sede di elaborazione e riservandolo alla valutazione e verifica dei risultati;
- ibridazione: l'utilizzo congiunto o sovrapposto di tecniche computazionali dall'opposta concezione teorica è esplorato e favorito laddove può portare al miglioramento dell'efficienza generale del sistema.

Il modello sviluppato è composto da una serie di strategie per l'analisi progressiva del testo arabo dei *ḥadīṯ* a partire dalla sua superficie, passando per alcune delle sue componenti più profonde fino a giungere al suo rapporto con altri testi. Tali strategie sono poi implementate attraverso un insieme di strumenti computazionali, appositamente ideati o modificati sulla base di applicazioni esistenti, che comprendono operazioni di segmentazione, ricerca interna di regolarità estrazione e rappresentazione di informazione, annotazione morfologica, comparazione con altri testi e allineamento bilingue.

0.3.1 Struttura e organizzazione della tesi

Questa dissertazione è organizzata in tre parti ciascuna comprendente un certo numero di capitoli.
Nella prima parte sono approfonditi gli aspetti e le implicazioni teoriche che innervano un possibile quadro di analisi del testo dei *ḥadīṯ*.

Il capitolo 1 contiene presenta e analizza alcune teorie e applicazioni di NLP specifiche per l'arabo, privilegiando quei settori che direttamente coinvolgono lo sviluppo di questa tesi (sono state per esempio tralasciate le tecniche di riconoscimento e sintesi vocale in quanto il piano fonetico/ fonologico non è particolarmente rilevante in questo contesto). Dopo un'introduzione sulle specificità di approccio che l'arabo richiede per il trattamento informatico, sono presentate alcune analisi di natura morfologica, che continuano a essere il focus della maggior parte degli approcci computazionali all'arabo. Gli studi sull'estrazione d'informazioni e la disponibilità di corpora e risorse lessicali precedono la parte dedicata al confronto tra più lingue, in particolare le tecniche di traduzione automatica, allineamento e definizione di memorie di traduzione. Il capitolo si chiude su alcune considerazioni riguardanti la prevalenza di studi sul trattamento dell'arabo contemporaneo rispetto alla variante classica, accompagnate da alcuni esempi invece di applicazioni NLP dedicate ad alcuni testi specifici del patrimonio classico letterario.

Il capitolo 2 indaga il genere letterario dei *ḥadīṯ*, presentando alcuni cenni storici relativi alle origini della Tradizione islamica, alla formazione del canone e alla costituzione delle raccolte più rappresentative di *ḥadīṯ*. Dopo una breve descrizione delle caratteristiche strutturali e compositive dei *ḥadīṯ*, un accenno è riservato alle scienze islamiche di studio, trasmissione, registrazione, critica e verifica della Tradizione, a cui segue una breve panoramica sugli studi della critica non arabo-islamica, in particolare quella europea e occidentale, con particolare riferimento agli studi relativi all'*isnād* e ai modelli di rappresentazione delle catene di trasmettitori.

La seconda parte affronta la descrizione delle strategie computazionali e degli strumenti di analisi progettati *ad hoc* per l'elaborazione automatica del testo dei *ḥadīṯ*.

Il capitolo 3 tratta di questioni metodologiche e dell'uso delle fonti per l'input. Dopo l'individuazione di alcuni parametri di metodo per la classificazione degli strumenti di analisi prodotti, sono riservati alcuni cenni alla descrizione dei linguaggi di programmazione informatica utilizzati. Le ragioni di quale edizione digitalizzata delle raccolte di *ḥadīṯ* è stata scelta come testo di input precedono alcune osservazioni generali e specifiche sui processo di codifica e traslitterazione dei caratteri arabi e la loro pertinenza.

Il capitolo 4 rappresenta il nucleo fondamentale del presente lavoro di ricerca, e tratta dell'analisi computazionale della struttura informativa di superficie dei *ḥadīṯ*. Dapprima si ipotizza l'esistenza di una sorta di 'valore computazionale originario' del testo, che permette di accostare la struttura di una raccolta di *ḥadīṯ* a quella di una moderna base di dati. Quindi lo strumento computazionale della sintassi delle espressioni regolari viene analizzato in dettaglio e proposto come chiave per l'interpretazione del testo. La parte successiva è dedicata alla descrizione del programma appositamente costruito per la segmentazione e l'estrazione di informazioni dalla raccolta di *ḥadīṯ*. Il capitolo si chiude con la presentazione di alcune tecniche per la rappresentazione multidimensionale delle informazioni estratte relative alle catene di trasmissione.

L'estensione dell'analisi dalla superficie del testo a una delle sue componenti linguistiche occupa la prima parte del capitolo 5. Il livello morfologico è esplorato attraverso la descrizione di un lemmatizzatore esistente e l'applicazione al testo degli *ḥadīṯ* di una sua versione opportunamente modificata e migliorata. La seconda parte del capitolo comprende la descrizione di alcuni programmi specifici creati per la ricerca testuale di stringhe in un testo e per l'identificazione di quali parti di un testo siano presenti in un altro, in particolare la presenza di citazioni coraniche nei *ḥadīṯ*. Viene anche presentata una strategia per l'appaiamento del testo arabo alla relativa traduzione in inglese, in modo da ottenere un esempio di memoria di traduzione allineata a livello di periodi.

La terza e ultima parte, costituita dal capitolo 6, presenta in modo dettagliato la valutazione quantitativa dei risultati ottenuti, in particolare la

segmentazione, l'analisi morfologica e l'allineamento bilingue. Delle conclusioni generali precedono infine alcune proposte inerenti le prospettive future di indagine. Una bibliografia completa delle opere citate e un'appendice contenente estratti di codice dei programmi e alcuni esempi di elaborazione dati chiudono la dissertazione.

Parte I:
TEORIE E TESTO

1. Lingua Araba e NLP: contesti, teorie, metodi

1.1 Il trattamento computazionale dell'arabo

L'applicazione delle tecniche di analisi computazionale alla lingua araba ha sin dalle origini affrontato una certa moltiplicazione delle difficoltà di processamento rispetto al trattamento di altre lingue, soprattutto europee (Soudi, van den Bosch, & Neumann, 2007, p. vii).

Il paradigma semitico di composizione morfologica attraverso la combinazione di radici e schemi (*roots and patterns*) rende particolarmente ardua l'applicazione di strategie analitiche a concatenazione di morfemi, in quanto i morfemi e le radici lessicali si combinano sia per successione (prefissazione, suffissazione), sia per interposizione (infissazione).

Un secondo ostacolo importante risiede nel carattere difettivo del sistema di scrittura arabo, laddove nella consuetudine corrente (eccettuati i testi sacri della tradizione religiosa, certa poesia classica e altri limitatissimi contesti) alcuni fonemi (vocali brevi, rafforzamenti consonantici, l'occlusiva glottale *hamza* in certi contesti) non vengono indicati esplicitamente in scrittura ma sono lasciati alla presupposizione del lettore.

Per l'arabo inoltre, il criterio dell'interpretazione degli spazi bianchi nel testo come discrimine tra le parole (qui intese come unità lessemiche o funzionali) è segnatamente messo in discussione. L'articolo, alcune preposizioni, congiunzioni e altri elementi funzionali come i pronomi sono direttamente prefissi alla parola successiva e sono cliticamente scritti insieme con essa senza soluzione grafica di continuità. Ad esempio, il processo di *tokenization*[4], vale a dire la riduzione di una stringa di testo in unità discrete sin-

[4] 'tokenizzazione' nel lessico terminologico italiano della linguistica computazionale.

tatticamente dotate di significato, solitamente ma non sempre parole, è un'operazione piuttosto semplice per l'inglese o l'italiano, siccome in queste lingue spazi bianchi sono marcatori univoci di separazione e gli elementi clitici. Applicando gli stessi presupposti interpretativi all'arabo, si otterrebbe però una segmentazione solo parziale, che non garantisce automaticamente l'ottenimento di tutti i lessemi, in quanto alcuni di essi presenteranno articoli o altri elementi clitici (congiunzioni, pronomi suffissi). La tokenizzazione dell'arabo quindi non può semplicemente basarsi sull'analisi superficiale del testo trattandolo come semplice successione di unità discrete, ma deve fare necessariamente ricorso a risorse lessicali o trattamenti morfologici in grado di aiutare la segmentazione corretta del testo (Elgibali, 2005).

1.2 La morfologia come *focus* privilegiato

Tra i vari piani di analisi interessati dallo sviluppo della linguistica computazionale applicata all'arabo, quello morfologico è stato indubbiamente il più percorso e analizzato. Questo è probabilmente dovuto a diversi fattori.

In primo luogo lato la necessità di supplire al carattere difettivo della scrittura e quindi ricostruire la pienezza fonologica dell'enunciato arabo porta naturalmente a privilegiare la morfologia come iniziale campo d'indagine. Soprattutto nel caso della lingua araba, i cui paradigmi di derivazione e flessione sono strutturati in un sistema piuttosto organico che ne preserva una certa 'leggibilità morfologica' fino allo stadio ultimo della parola derivazionalmente formata, soprattutto in termini di radice e schemi derivati (Al-Sughaiyer & Al-Kharashi, 2004).

Inoltre l'estrema chiarezza, almeno negli intenti, con la quale i grammatici arabi hanno costruito un edificio morfologico che potesse rendere conto in modo organico di ogni costruzione lessicale, sia regolare sia aberrante, è in certa misura accostabile a metodologie di tipo computazionale (Bohas & Guillaume, 1984). Si può pensare alla morfologia derivativa dell'arabo come

a un processo con molte analogie 'computazionali'[5]: le triplette di consonanti che costituiscono la base radicale del significato della gran parte delle parole arabe (accanto ad alcuni gruppi di coppie o quartetti consonantici) possono essere interpretate come input; l'applicazione alla radice degli schemi derivazionali e flessionali sarebbe un esempio particolarmente raffinato di *processing*; l'output finale consisterebbe nelle parole stesse che compongono l'inventario lessicale. In questo senso si potrebbe con una certa arditezza affermare che la morfologia dell'arabo sia una sorta di complesso programma informatico, che tratta radici e costruisce parole.

Passando a una breve panoramica dello sviluppo della morfologia computazionale applicata all'arabo, si possono identificare due paradigmi generali di analisi, tra loro alternativi. Il primo è costituito dagli approcci basati su una conoscenza linguistica pregressa (*knowledge-based*), l'altro riguarda metodi empirici e statistici, che tendono invece a trattare il dato testuale come semplice combinazione in stringhe di unità discrete a prescindere dal loro significato linguistico (Roberts, 2006).

1.2.1 Approcci basati sulla conoscenza

Un approccio computazionale basato sulla conoscenza (*knowledge-based*) tratta il dato testuale utilizzando ampie conoscenze di tipo linguistico, sotto forma sia di risorse (dizionari, repertori di annotazioni morfosintattiche e lessicali) sia di strategie di analisi (teorie linguistiche e paradigmi, modelli grammaticali formali). In tal senso il testo è trattato sin dall'origine tenendo conto di tutte le sue valenze grammaticali e linguistiche, a livello esplicito o implicito (Shaalan, 2010). La chiave di questo approccio generale consiste nella codificazione di regole e condizioni strettamente linguistiche per il riconoscimento e la classificazione della struttura interna di parola. Perché il modello funzioni è però necessario che tali regole costituiscano un sistema linguistico il più consistente possibile: se il sistema cioè non sarà in grado di

[5] Il tipico processo informatico prevede nella sua definizione più semplice una fase di inserimento dei dati (*input*), un successivo trattamento di questi attraverso algoritmi di elaborazione dei dati (*processing*), e una restituzione finale di un risultato (*output*), che rappresenta appunto la trasformazione del dato originario (McCarty, 2005, p. 9).

prevedere regole per ogni fenomeno e comportamento linguistico, in questi casi l'analisi restituirà un errore. L'efficacia dell'approccio basato sulla conoscenza dipende quindi essenzialmente dalla raffinatezza e dalla complessità dell'insieme delle regole di trasformazione utilizzate. Per l'arabo possono essere identificati quattro quadri di riferimento all'interno dei quali si situano gli studi e le applicazioni computazionali basate sulla conoscenza. (Soudi, van den Bosch, & Neumann, 2007, p. 4-6)

1.2.1.1 Morfologia sillabica

Nella morfologia sillabica (*Syllable-based Morphology*, SBM) le realizzazioni morfologiche sono definite in base alla struttura sillabica di parola, che è quindi considerata come costituita da una stringa lineare di sillabe dotate di valore morfologico attivo (Cahill L. , 1990). Sebbene questo modello sia stato applicato principalmente alle lingue europee, specie di ceppo germanico, alcuni studi dimostrano che esso è applicabile con successo anche all'analisi morfologica computazionale delle lingue semitiche, prima fra tutte l'arabo. (Cahill L. , 2007).

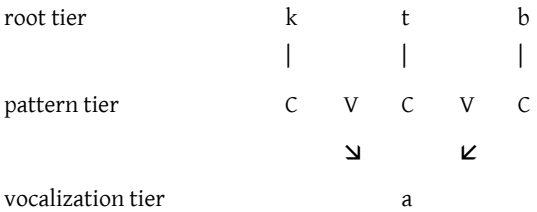
1.2.1.2 Morfologia radice-schema

La sintesi del paradigma radici-schemi (*root and pattern morphology*) proposta da McCarthy (1981) rappresenta il più utilizzato modello di riferimento per l'analisi morfologica computazionale dell'arabo. In tale concezione, i temi (*stems*) sono formati attraverso una combinazione derivazionale di un morfema radice e una melodia vocalica. Le radici sono quindi interpolate (*interdigitated*) e fuse con gli schemi (*patterns*) per formare i vari temi. Tale modello differisce però da quello di Harris (Harris, 1941) poiché le vocali sono visualizzate in un livello (*tier*) ulteriore rispetto a quello del pattern (*autosegmentation*) (Soudi, van den Bosch, & Neumann, 2007, p. 5):

Analisi segmentale di Harris:

root: k t b *pattern*: _a_a_ *stem*: katab ('ha scritto')

Analisi autosegmentale di McCarthy:



Aggiungendo un livello all'analisi, l'approccio autosegmentale costituisce una complessificazione di quello segmentale, e riesce a rendere conto di fenomeni come lo *spreading* consonantico (Beesley K. , 1998, p. 119-120).

L'approccio autosegmentale è adottato da parecchi sistemi computazionali che tentano di modellare la morfologia dell'arabo, soprattutto quelli cosiddetti a stati finiti. La morfologia a stati finiti è basata sull'assunto che la relazione tra forme di superficie e forme sottese, ad esempio le regole di variazione fonologica/ortografica o morfosintattica, può essere formalizzata seguendo la tecnologia dell'automa a stadi finiti (*finite state technology)*. Tale dispositivo astratto è caratterizzato dal presupposto di potersi trovare in ogni momento solo in uno specifico stadio (finito, appunto), e di poter cambiare stadio sotto il verificarsi di specifici eventi (input o condizioni) che mutano le condizioni iniziali[6] (Gill, 1962).

Tra i vari approcci all'arabo che hanno applicato questo modello (Beesley K. , 1996) (Kay, 1987) (Kiraz, 2000), quello sviluppato con la tecnologia a stadi finiti Xerox (Xerox, 2013) è uno dei più formalizzati (Beesley K. , 1998). In tale modello la digitazione di caratteri appartenenti alla radice e allo sche-

[6] Il classico esempio di una macchina a stadi finiti è quello che regola l'erogazione a pagamento delle bevande di un distributore automatico: All'inizio lo stato del sistema è 'chiuso, non permette cioè di accedere direttamente al prodotto. All'inserimento di una moneta il sistema verifica le nuove condizioni (è stata o no raggiunta la somma necessaria per pagare il prodotto?), se le condizioni sono soddisfatte cambia di stato ('aperto', erogazione della bevanda) altrimenti rimane nello stato precedente. Ogni volta che un'altra moneta viene aggiunta, il sistema verifica le condizioni fino a che esse non consentono di cambiare stato (il raggiungimento della somma necessaria per avere la bibita) che consentirà al distributore di erogare il prodotto (Wright, 2005).

ma è un processo di semplice intersezione. Le radici trilittere sono rappresentate dall'espressione *?*C?*C?*C?**, dove '*' denota zero o più concatenazioni di qualunque '?' (che indica 'qualunque simbolo'). La radice *d r s* ad esempio può essere rappresentata come *'?*d?*r?*s?** L'intersezione di questa rappresentazione con lo schema *CaCaC* produce *daras* , '(ha) studiato', che può essere a sua volta rappresentato come

(1) [drs & CaCaC][7]

oppure con auto-segmentazione delle vocali a livello lessicale astratto

(2) [drs & CVCVC] [a]

Per ciascuna coppia radice/schema una regola di tipo (1) è generata automaticamente e in seguito compilata nel corrispondente trasduttore (*transducer*)[8] che codifica la stringa *[drs & CaCaC]* in *daras*.

Si consideri ora il trattamento di un verbo concavo[9], ad esempio la voce verbale *qāla* ('lui ha detto'). Nel sistema Xerox, il livello inferiore (primo stato), contiene la voce *qwl* 'il concetto del dire', forma profonda della radice così come appare nel dizionario delle radici utilizzato dal modello, che riporta per ogni radice tutte le forme che essa può assumere. A tale voce viene applicato lo schema del tempo passato integrato con l'indicazione della flessione, che al livello superiore è *[qwl & CaCuC] + a,* in modo da ottenere la forma intersecata a livello intermedio *qawul+a*. Infine alcune regole di alternanza combinatoria mappano *qawul+a* in *qāla.* Sia la base lessicale che le regole sono compilate in un trasduttore, che le combina insieme attraverso un'operazione a stati finiti chiamata composizione. Nel caso di cui sopra, ad esempio, la *w* di *qawul+a* viene cancellata e sostituita da una *ā.* Il risultato è quindi un trasduttore a stati finiti a due livelli che codifica direttamente stringhe, e dove la forma intermedia scompare nella composizione:

[7] Le parentesi quadre indicano i confini del tema e il simbolo '&' l'operazione di intersezione.
[8] Sempre un'automazione a stati finiti ma che prende in considerazione due stringhe anziché una.
[9] La cui seconda radicale è la semiconsonante *w* o *y* che in determinati contesti possono assimilarsi alla vocale contigua o generare un nuovo corredo vocalico.

Livello superiore: *[qwl & CaCuC] + a*

Vocalizzazione piena: *qāla*

Il procedimento è bidirezionale, ed è cioè sia analitico (da *qāla* si può risalire a *[qwl & CaCuC] + a)* sia generativo (*[qwl & CaCuC] + a* produce *qāla*). Uno dei limiti di questo sistema e di altri analoghi resta però l'impossibilità di rendere conto di alcuni fenomeni linguistici come il sincretismo (o omonimia inflessionale sistematica) della flessione verbale e nominale.

L'approccio proposto da Al-Najem integra invece la formalizzazione multilineare di McCarthy alla possibilità di riconoscere dipendenze, sincretismi e altri fenomeni eccentrici della morfologia araba (Al-Najem, 2007). La sistematizzazione di questi fenomeni è realizzata sul piano computazionale attraverso l'uso del linguaggio di inferenza DATR, che rappresenta la conoscenza lessicale come un sistema di nodi costituito da temi e parole codificati da un set di attributi, in analogia con i sistemi ontologici descritti al titolo 1.5.

1.2.1.3 Morfologia basata sui lessemi

Diversamente dagli approcci precedenti, dove i vari livelli costituenti di parola quali radice, schema (*pattern*), vocalizzazione vengono trattati con pari dignità computazionale e appartengono ciascuno a una base lessicale differente che poi il modello elabora insieme per l'analisi o generazione, la morfologia basata sui lessemi (*lexeme-based morphology*) identifica il tema come unica unità morfologicamente rilevante per l'elaborazione computazionale, in quanto dominio fonologico delle regole di realizzazione. In questo approccio quindi, i temi e le operazioni sui temi sono il cuore del livello di rappresentazione, la base lessicale utilizzata è unica e solo liste di temi la compongono (Cavalli-Sforza & Soudi, 2007).

1.2.1.4 Lessico basato sui temi con specificazioni grammaticali e lessicali

Questo approccio si basa essenzialmente su basi di dati lessicali contenenti i temi, in cui però ogni voce è accompagnata da informazioni grammaticali e restrizioni semantiche. Questo basandosi sull'ipotesi che la considerazione

adeguata delle relazioni tra grammatica e lessico sia un criterio imprescindibile per l'analisi morfologica dell'arabo.

Alcune debolezze tipiche del modello di rappresentazione radice-schema sono superate utilizzando il modello basato sulla centralità del tema ma integrandolo con una definizione tratta da liste finite di specificatori morfosintattici di parola (*w-specifiers*). I sistemi *DIINAR.1 Dictionnaire Informatisé de l'Arabe* (Dichy, Braham, Ghazali, & Hassoun, 2002) e *SYSTRAN Arabic-English Translator* sono esempi di questo approccio che unisce ai database lessicali dei dispositivi di analisi morfologica per regole (Dichy & Farghali, 2003).

1.2.1.5 Analisi e identificazione lessicale

Si è già accennato al fatto che nella linguistica computazionale si definisce come *token* una sequenza di caratteri di una stringa caratterizzata dall'essere la minima unità sintattica possibile. Il processo di analisi e conversione del testo in una successione di tali unità è chiamato tokenizzazione, e costituisce uno dei pretrattamenti ritenuti fondamentali prima di qualsiasi elaborazione computazionale del testo arabo, in quanto gli spazi bianchi non sono un marcatore esclusivo e biunivoco dei confini tra un *token* e l'altro. Le principali strategie di tokenizzazione per l'arabo seguono generalmente tre impostazioni diverse.

La prima utilizza per la segmentazione in unità sintattiche un analizzatore morfologico fondato su regole linguistiche, che produce allo stesso tempo sia la riduzione in *token* sia la marcatura morfologica (Habash & Rambow, 2005). Ad esempio il trattamento della stringa وليشكر ('e per ringraziare') darà come risultato:

```
و+conj@ ل+comp@+verb+pres+sg @شكر
```

dove il simbolo @ indica il confine del *token*. Un tale sistema viola però il concetto di modularità che prevede di avere un dispositivo separato per ciascuna analisi. La modularità consente ad esempio il riutilizzo dei dispositivi ad esempio per l'analisi sintattica (*parsing*), che richiede l'esistenza di un tokenizzatore separato dall'analizzatore morfologico.

La seconda strategia azzera le regole di tipo linguistico e prevede l'utilizzo di un programma di predizione, che tenta di 'indovinare' la corretta sequenza di tokenizzazione secondo una lista data di clitici che possono essere prefissi o suffissi a un dato *token*. Il sistema è piuttosto robusto poiché non necessita di identificare i temi (per i quali sarebbe necessaria una corposa risorsa lessicale) ma solo gli spazi bianchi di parola e tutti i clitici possibili. La sua efficienza è però limitata dal numero piuttosto rilevante di soluzioni ambigue proposte che un sistema come questo, incapace di selezione, può produrre (Attia, Arabic tokenization system, 2007).

La terza strategia infine, combina la robustezza e la semplicità del tokenizzatore predittivo con un analizzatore morfologico indipendente, che però risolve le ambiguità del primo utilizzando un repertorio lessicale di temi per la selezione (Attia, 2006).

1.2.2 Approcci empirici

I modelli puramente basati sulla conoscenza richiedono che tutte le possibili strutture da codificare nel sistema siano conosciute dal linguista e possano essere formalizzate e sviluppate in modo consistente e non ridondante. Tali tentativi di sistematizzazione hanno in comune due inconvenienti principali, l'ambiguità e la copertura. Il sistema basato sulla conoscenza restituisce infatti i risultati in termini di vero o falso, cioè di semplice soddisfacimento binario di una condizione. Con questi presupposti è frequente trovare più soluzioni che soddisfano la stessa condizione. Tali soluzioni, giudicate come vere dall'elaboratore computazionale rimangono invece concorrenti o alternative dal punto di vista della comprensione linguistica del testo, generando un certo livello di ambiguità, risolvibile in termini di sistemi di regole solo con lo sforzo ideale di arrivare a una formalizzazione omnicomprensiva della lingua in questione (Schütze, 1997). La riduzione delle ambiguità è quindi connessa alla difficoltà intrinseca a estendere indefinitamente la copertura a livello teorico degli avvenimenti di una lingua. Per avere un sistema pienamente efficiente bisognerebbe non solo conoscere e prevedere tutti i fenomeni linguistici possibili, ma anche riuscire a sviluppare per ogni fenomeno una formalizzazione e applicazione che funzionino a livello di

elaborazione computazionale (Soudi, van den Bosch, & Neumann, 2007, p. 8-10).

I sistemi empirici capovolgono questo presupposto teorico. Invece di trattare i fenomeni linguistici cercando di spiegarli e formalizzarli, cercano di ottenere risultati affidandosi non a regole predefinite ma alla classificazione (*ranking*) e selezione computazionale dei dati basate su algoritmi di tipo statistico.

Lo sviluppo di metodi empirici per la linguistica computazionale risale ai primi anni '90 e prevede generalmente l'impiego di tecniche per l'istruzione del computer (*machine learning techniques*) in modo che esso estragga la conoscenza linguistica dal linguaggio naturale in modo automatico e diretto, senza dover fornire manualmente in input tale conoscenza al sistema stesso sotto forma di insiemi di regole (Cardie & Mooney, 1999). Sebbene agli inizi i modelli empirici fossero considerati antagonisti e in competizione con quelli basati sulla conoscenza, la tendenza più recente è piuttosto quella di integrarli in sistemi ibridi che prevedano un'elaborazione centrale di tipo statistico combinata con moduli di tipo normativo-linguistico (Magerman, 1995). Gli approcci empirici possono essere raggruppati secondo alcune questioni fondamentali sintetizzate come segue.

1.2.2.1 Automaticità e supervisione. Addestramento della macchina

Applicare un metodo empirico piuttosto che definire a priori un sistema di regole comporta anche una riflessione specifica sul grado di automazione del processo stesso, e cioè su quanto si debba fare ricorso a procedimenti manuali o a risorse più o meno corredate di informazioni linguistiche inserite manualmente in precedenza. Tali riflessioni sono alla base del modello presentato da Clark per analizzare i fenomeni di morfologia non concatenativa tipici dell'arabo, come il caso dei plurali fratti. Il metodo usa un set di trasduttori stocastici, di procedimenti cioè che agiscono sul piano probabilistico selezionando i risultati in output sulla base di sistemi di classificazione (ranking) e ordinamento dei possibili candidati presentati da un archivio a base statistica. L'addestramento della macchina è dapprima compiuto in modo supervisionato e in seguito con un algoritmo che identifica la morfo-

logia non concatenativa impostando un processo automatico che considera come variabile nulla il morfema flessivo del plurale (Clark, 2007).

1.2.2.2 Integrazione e modularità

Tra gli approcci puramente empirici e basati totalmente sull'istruzione statistica della macchina quello proposto da Diab, Hacioglu e Jurafsky considera la morfologia come parte integrale del più vasto problema della segmentazione e marcatura (*tagging*) delle varie parti del discorso. L'interesse si sposta pragmaticamente dall'analisi morfologica pura all'identificazione, segmentazione e marcatura semi-automatica degli elementi clitici (Diab, Hacioglu, & Jurafsky, 2007).

1.2.2.3 Trasferibilità e ibridazione

Un esempio di metodo empirico ibrido applicato all'analisi morfologica dell'arabo è quello dell'identificazione delle consonanti radicali di parola. Rispondendo all'interrogativo se sia possibile aumentare l'efficienza dei metodi statistici in accordo con un insieme di regole basato sulle conoscenze linguistiche codificate, il sistema proposto da Daya, Roth e Wintner costruisce un approccio che combina analisi statistica e sistemi di regole[10], e compara i risultati con quelli ottenuti da un analogo trattamento di un'altra lingua semitica come l'ebraico. Il problema del trattamento estensivo delle radici arabe trilittere viene risolto con l'uso di un insieme di classificatori in grado di predire statisticamente dire il valore delle consonanti radicali a partire dal tema di parola, combinato con una serie di semplici condizioni linguistiche che ne aumentano l'efficacia d'analisi (Daya, Roth, & Wintner, 2007).

[10] Un interessante approccio alternativo per l'estrazione delle radici arabe consiste nell'utilizzo della teoria delle catene markoviane, dove il passaggio da uno stato finito all'altro dipende esclusivamente dallo stato di sistema immediatamente precedente e non dal modo con cui si è giunti a tale stato (Bebah, Belahbib, Boudlal, Lakhouaja, Mazroui, & Meziane, 2011).

1.3 Corpora e risorse lessicali

Affinché gli algoritmi statistici di trattamento del testo possano essere realmente efficienti, è necessario che la base di dati usata per istruire il computer sia la più vasta possibile. La linguistica dei corpus supplisce a questa necessità, studiando la costruzione di sistemi di organizzazione formale di risorse testuali e lessicali. La presenza di risorse testuali e lessicali in lingua araba organizzate in corpora è piuttosto consistente, anche se privilegia decisamente la varietà moderna standard rispetto a quelle classica e letteraria. Una classificazione possibile delle risorse digitali le divide in esplicite o implicite, in dipendenza del livello di formalizzazione interna e della presenta di un intento lessicografico dichiarato da parte del compilatore o dell'editore.

1.3.1 Risorse digitali esplicite

Comprendono risorse testuali (raccolte organizzate di testi) oppure risorse lessicali (che organizzano l'informazione solitamente in voci singole o lemmi, ai quali corrisponde un certo contenuto informativo).

I corpora testuali possono essere ulteriormente distinti in dipendenza del loro grado di annotazione. Il grado zero consiste in semplici raccolte organizzate di testi reali. Il livello più basso di annotazione fornisce per ogni testo informazioni generali sulla tipologia, sul registro e sulla fonte di provenienza. Un testo parzialmente o totalmente annotato è strutturato in unità più piccole (periodi, frasi, sintagmi, parole), ciascuna delle quali può essere variamente integrata con informazioni di tipo morfologico, sintattico o semantico (Hunston, 2005; Maamouri, Bies, Buckwalter, & Mekki, 2004).

I corpora paralleli consistono in raccolte di testi nelle quali ogni brano in arabo è associato alla relativa traduzione in un'altra lingua o in più lingue. Vengono classificati a seconda del grado di segmentazione e allineamento interno, vale a dire la possibilità di recuperare la traduzione parallela non solo per il brano nella sua interezza, ma anche per le sue unità costitutive, come periodi, frasi, sintagmi o singole parole. In questi strumenti non sempre è però possibile conoscere la direzione della traduzione e quindi distin-

guere tra lingua d'origine e lingua di traduzione (Kutz, Normann, Mossakowski, & & Dirk, 2010). Concettualmente i corpora paralleli, conosciuti anche con l'espressione 'memorie di traduzione' (*translation memories*), sono opposti ai sistemi di traduzione automatica in quanto non generano la traduzione ma allineano traduzioni esistenti. I corpora paralleli sono una risorsa essenziale per la traduzione automatica di tipo empirico-statistico, che seleziona i risultati proprio in base a una classificazione di pertinenza degli allineamenti esistenti tra lingue diverse (Simard M. , 2003).
Le risorse lessicali propriamente dette comprendono i dizionari monolingua, bilingui e plurilingui, i database lessicali e terminologici. Molte di queste opere sono però concepite come sistemi chiusi e proprietari, utilizzabili liberamente solo per la consultazione ma non per l'importazione e la successiva elaborazione in processi NLP. Un'eccezione per l'arabo è costituita dalla digitalizzazione del dizionario arabo-inglese di Salmoné (Salmoné, 1889), della quale si parlerà specificamente al titolo 5.1.3.2 in merito al miglioramento della base lessicale di un analizzatore morfologico.

1.3.2 Risorse digitali implicite

Alcuni recenti studi di lessicografia e grammatica computazionale sfruttano come risorsa per la definizione lessicale repertori non originariamente costruiti per questo scopo, ma interpretabili computazionalmente come basi lessicali ordinate di dati (Pasternack, 2008; Cu, Lu, Li, & Chen, 2008). La versione araba di Wikipedia contiene circa 230.000 voci, che comprendono definizioni enciclopediche, lessicografiche e un certo numero di *named entities*. Il libero accesso ai file di codice contenenti l'intera enciclopedia, permette di avere a disposizione una grande quantità di materiale lessicale per successive applicazioni di NLP, e soprattutto consente di importare automaticamente come informazione la rete di relazioni multilingua che legano una voce araba alle corrispettive in versioni Wikipedia di altre lingue. Mantenendo le voci di lemma e trascurando le definizioni dei contenuti (che quasi mai corrispondono da una lingua all'altra), questa rete può costituire l'embrione precostituito di un dizionario multilingua esteso al riconoscimento di un certo numero di *named entities*.

1.3.3 Un elenco ragionato di alcune risorse[11]

RISORSA	TIP	LIN	CON	ANN	ACC
Graff, D. *Arabic Gigaword*, LDC, 2003 Corpus monolingua Tipo di testi: giornalistici (AFP, Al Hayat, AlNahar, Xinhua)	T	AMS	391	0	P
Arabic Data Set, ELDA; University of Essex Corpus monolingua Tipo di testi: giornalistici (Al Hayat)	T	AMS	18	1	P
Parkinson, D. *arabiCorpus*, 2012 http://arabicorpus.byu.edu/ Corpus monolingua Tipo di testi: giornalistici, letterari, saggistica, arabo colloquiale egiziano, testi classici e religiosi	T	AMS AEG	173	2 MOR	L
UNBIS Thesaurus, UN 1980-2013 http://lib-thesaurus.un.org/ Corpus parallelo *UNTERM*, UN http://unterm.un.org/ Database terminologico Tipo di testi: relativi alle NU	T/L	AR EN FR ES RU CH	<0,1	1	L
Sahr Part of Speech Tagged Corpus http://www.sakhr.com/arabicresources.aspx Corpus annotato	L	AR EN	7,2	3 MOR SIN	PAG
Maamouri, M, Bies, A., Buckwalter, T. & Jin, H, Penn *Arabic Treebank*, LDC & University of Pennsylvania, 2005 Database di rappresentazione sintattica Tipo di testo: giornalistico (AFP)	T	AR	1	3 MOR SIN	PAG

[11] La finalità di questo elenco non è presentare tutte le risorse esistenti, ma mostrarne alcuni esempi rappresentativi.

RISORSA	TIP	LIN	CON	ANN	ACC
Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J., , Beška, E. *Prague Arabic Dependency Treebank,* 2003 Database di rappresentazione sintattica Tipo di testo: giornalistico	T	AR	0,1	3 MOR SIN	PAG
Dukes, K. Quranic Arabic Corpus, Language Research Group, University of Leeds, 2009-2011 http://corpus.quran.com Corpus annotato Tipo di testo: Corano	T/L	AR EN	<0,1	3 MOR SIN SEM	OS
Fellbaum, C. (ed), ArabicWordnet, 2005 http://www.globalwordnet.org/AWN Database di reti semantiche a base ontologica	L	AR EN	<0,1	3 SEM	OS
Salmoné, H. An Advanced Learner's Arabic-English Dictionary. Beirut : Librairie du Liban 1889; edizione digitalizzata TEI 2010.	L	AR EN	0,05	2 SEM	OS
Meedan -translation memory, 2009 Memoria di traduzione http://www.meedan.com	T	AR EN	0,02	2	OS

Legenda: TIP(Tipologia): TES(testuale); LES(lessicale); LIN: lingua o lingue contenute; CON: Consistenza in milioni di parole. ANN: Grado di annotazione: 0 (non annotato); 1 (annotato a livello del brano); 2 (parzialmente annotato); 3 (annotato); Tipo di annotazione: MOR (morfologica); SIN(sintattica); SEM (semantica); ACC: Accesso: OS(Open Source); LIB (libero in consultazione); PAG (a pagamento).

1.4 Il confronto fra più lingue

1.4.1 Traduzione automatica: tecniche e tendenze

La traduzione automatica (o *machine translation*) è uno dei primi e più esplorati settori applicativi della linguistica computazionale. La complessità del suo scopo principale richiede la combinazione di analisi di tipo morfologico, sintattico, lessicale, semantico e testuale, e comporta un certo numero di problemi ancora insoluti, quali ad esempio il trattamento di *named entities*, espressioni idiomatiche, deissi testuale, elisione di informazione. Il grado di

efficacia e precisione della traduzione automatica (TA) è ancora lontano da poter essere paragonato a quello della traduzione umana, ciononostante questo settore continua a mostrare una vivacità di teorie e metodi che interessano in qualche misura gli altri campi di analisi della linguistica computazionale (Goutte, Cancedda, Dymetman, & Foster, 2009).

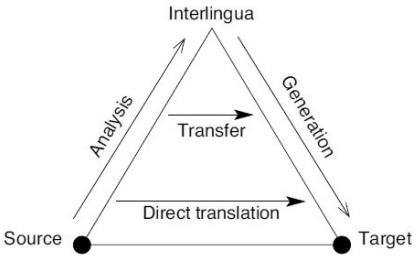


Figura 1.1: La piramide della traduzione automatica

In conformità paradigma computazionale che alterna i modelli basati su regole a quelli di tipo empirico-statistico, i tre principali approcci alla TA (cfr. figura 1.5) comprendono:

- sistemi di traduzione diretta: usano tipicamente gli approcci empirici e si basa sulla consultazione di grandi repertori preesistenti di testi originali allineati alle rispettive traduzioni. Un modello di inferenza statistica classifica tutte le traduzioni possibili e ne seleziona su basi statistiche quella considerata più pertinente. Come si vede in figura 1.1., questo processo non coinvolge minimamente la rappresentazione dei livelli linguistici: le traduzioni esistono già e il cuore del processo è il corretto allineamento di originali e traduzioni (Koehn, 2010). Molti servizi di traduzione disponibili in linea, *Google Translate* ad esempio, impiegano questi sistemi. Questo approccio richiede inoltre una grande sofisticazione nella valutazione quantitativa e qualitativa dei risultati (Papineni, Roukos, Ward, & Zhu, 2002).

- ❖ sistemi di rappresentazione: attraverso l'uso di modelli grammaticali e linguistici completi analizzano il testo nella lingua di origine, ne forniscono una rappresentazione a livello di interlingua grazie ai propri modelli teorici, e generano nuovo testo per la lingua di arrivo (Cardie & Mooney, 1999);
- ❖ sistemi di trasferimento: utilizzano tecniche che integrano sistemi ad hoc di regole morfologiche e sintattiche con selezione statistica delle traduzioni.

In merito alla TA da e verso l'arabo, si possono individuare due tendenze di sviluppo, una di natura universalistica che mira a estendere gli esistenti sistemi al trattamento dell'arabo, l'altra di tipo localizzativo che aspira alla progettazione di sistemi specifici basati sulle caratteristiche linguistiche peculiari dell'arabo (Guidère, 2002).

Il tipo estensivo è quello più percorso dagli approcci statistici, che non riferendosi a nessun modello linguistico in particolare, sono poco influenzate dalla natura delle lingue scelte per la traduzione. La questione da affrontare riguarda quindi principalmente il reperimento di originali e traduzioni, il relativo allineamento, l'applicabilità degli algoritmi statistici originali alle specificità della lingua araba (Badr, Zbib, & Glass, 2008; Ghoniem, Tokal, & Tawfik, 2011; Zughoul & Abu-Alshaar, 2005).

Il tipo localizzativo è solitamente caratterizzato dal ricorso esclusivo o ibrido a modelli basati su regole, che tentano di rappresentare la struttura sottostante alle realizzazioni linguistiche dell'arabo e delle lingue di traduzione (Hebresha & Ab Aziz, 2013; Shaalan, 2010; Salem & Nolan, 2009).

1.4.2 Allineamento e memorie di traduzione

L'allineamento costituisce il processo chiave delle tecniche statistiche di TA. Esso consiste nell'analisi di database di testi multilingua al fine di associare correttamente attraverso algoritmi specifici gli elementi corrispondenti nelle diverse lingue; siano essi paragrafi, frasi, sintagmi, parole.

Nel caso dell'arabo, le tecniche di allineamento a livello di frase solitamente prevedono modelli interpretativi basati sulla lunghezza delle stringhe o sulla loro componente lessicale (Salameh, Zantout, & Mansour, 2011).

Uno dei metodi più utilizzati prevede l'identificazione di alcune parole di 'ancoraggio' nel testo d'origine (TO) e nel testo tradotto(TT) attraverso metodi statici o vettoriali bidimensionali (Simard & Foster, 1992).

L'allineamento tra parole è interessato maggiormente dalle differenze strutturali, lessicali e grammaticali tra le lingue. Le tecniche prevedono o un approccio di tipo associativo che usa modelli euristici (Osh & Ney, 2003) oppure uno di tipo estimativo che impiega regole di tipo statistico (Melamed, 1996). Entrambe le strategie mostrano la ridondanza del pretrattamento morfologico del testo (Habash & Sadat, 2006).

Le memorie di traduzione (*translation memories*) sono una strategia diversa di applicazione delle tecniche di allineamento. Se la traduzione automatica le usa come elemento di partenza per ottenere la traduzione di una singola parola e valutarne automaticamente la pertinenza a seconda del contesto, le memorie di traduzione limitano l'allineamento a frasi e in qualche caso parole, ma invece di produrre autonomamente la traduzione ritenuta migliore, offrono come output una selezione di contesti in originale e in traduzione nei quali la parola stessa è impiegata. La valutazione della scelta migliore di traduzione è quindi lasciata all'intervento umano. Le memorie di traduzione sono quindi uno strumento computazionale di aiuto alla traduzione, e non sostituiscono il lavoro del traduttore. Inoltre, non dovendo operare scelte di traduzione, riducono o meglio neutralizzano l'ambiguità dell'output presentato (Lagoudaki, 2006).

Per la lingua araba è presentato un esempio di codice tratto dalla *Meedan Translation Memory* (Meedan, 2009), una memoria di traduzione arabo-inglese che accoppia circa 40.000 frasi ed è rilasciata con licenza Open Source, quindi liberamente consultabile nei suoi file di programma:

```
<tu tuid="1252551073875">
<tuv xml:lang="EN">
<seg>Frederick Kanoute, Sevilla's player, reading
Al-Fatihah before going to the stadium</seg>
</tuv>
<tuv xml:lang="AR">
<seg>فريدرك كانوتية لاعب اشبيلية يقرأ الفاتحة قبل نزول الملعب</seg>
</tuv>
</tu>
```

Le versioni araba e inglese della stessa frase sono identificate per un mero e lingua di appartenenza, e organizzate con la sintassi tipica dei linguaggi descrittivi come l'XML[12].

1.5 Rappresentazione della conoscenza e ontologie per l'arabo

All'interno della più generale area di studi della semantica computazionale, l'identificazione di reti semantiche che connettono concetti e realizzazioni linguistiche avviene analizzando la mappa di relazioni tra significati e la loro posizione reciproca (Vossen, 1998). Queste analisi hanno condotto alla costruzione di ontologie, vale a dire sistemi formali ordinati di relazioni tra i significati che tentano di modellare in modo complementare la conoscenza della realtà esterna (Gómez-Pérez & Corcho, 2002). Ad esempio il significato del lemma 'automobile' non è inteso tanto come definizione a sé stante, ma come uno specifico posizionamento in una rete che comprenderà relazioni sinonimiche con 'autoveicolo' e 'macchina', relazioni iponimiche con 'veicolo' e relazioni iperonimiche con 'berlina' o 'utilitaria'. Le ontologie sono oggi un elemento fondamentale per l'analisi semantica computazionale, come dimostrano la diffusione e l'utilizzo NLP di risorse come *SUMO* (Niles & Pease, 2001), *OpenCyc* (Matuszek, Cabral, Witbrock, & DeOliveira, 2005) e *DOLCE* (Masolo, Borgo, Gangemi, Guarino, Oltramari, & Schneide, 2003).

Per quanto riguarda le ontologie specifiche per la lingua araba e i concetti della cultura arabo-islamica, esse sono di tre tipi. Il primo comprende l'estensione e l'adattamento per l'arabo di ontologie esistenti. Nel caso di *SUMO*, la distribuzione corrente è corredata di una ridotta ontologia di 70 concetti che mostra però una scarsa accuratezza di annotazione delle informazioni culturali[13].

[12] Per magiori dettagli sull'XML cfr. titolo 3.2.2.

[13] Ad esempio il termine *suḥūr*, il pasto consumato dai musulmani durante il mese di digiuno di Ramaḍān è codificato come sotto-classe di 'mangiare' senza alcuna referenza alle implicazioni culturali correlate al concetto.

Il secondo tipo di ontologie per l'arabo prevede versioni parallele di ontologie preesistenti, come il caso di *Arabic WordNet* (Black, et al., 2006) che è la versione localizzata per l'arabo di *Princeton WordNet* (Miller, 1995; Fellbaum, 1998) *Arabic Wordnet* ha una copertura molto più limitata del *WordNet* originale (11.000 *synsets*, o set di sinonimi, contro circa 118.000), è stata annotata manualmente, e si limita alle risorse testuali contemporanee.

Il terzo tipo identifica ontologie specificamente costruite per la cultura arabo-islamica. Tra queste, un esempio rilevante è *l'Arabic Ontologic Project* (Jarrar, 2011), che però è tuttora in fase di sviluppo. Il progetto mira a costruire un'ontologia specifica solo dopo aver mappato le ontologie esistenti, aggiungendo a queste un database ontologico arabo di circa 10.000 termini estratti in modo semiautomatico da dizionari esistenti.

1.6 L'approccio digitale all'arabo classico

Lo sviluppo delle tecniche di elaborazione del linguaggio naturale per l'arabo, soprattutto per quanto riguarda gli approcci empirici che richiedono una grande quantità di dati analizzabili, ha decisamente privilegiato l'approccio alla lingua scritta contemporanea rispetto che a quella del patrimonio letterario arabo classico e moderno. Nella letteratura computazionale di settore infatti, i principali modelli di analisi sono concepiti e verificati utilizzando basi di dati e risorse lessicali soprattutto basate su testi molto recenti appartenenti al genere giornalistico e documentale.

Le potenzialità e i vantaggi computazionali di riferirsi al corpus di letteratura classica sono interessanti e numerose. Innanzitutto la disponibilità di ingenti quantità di opere appartenenti allo stesso genere letterario, una forte tendenza alla standardizzazione interna dei generi, la possibilità di fruire di testi a vocalizzazione parziale o totale e infine, in alcuni generi, una certa formalizzazione dell'organizzazione interna del testo, per cui testi analoghi presentano le medesime caratteristiche strutturali. In campo computazionale, la presenza di una struttura omogenea e ricorrente alle stesse condizioni in contesti diversi è sovente una condizione ideale per analisi efficaci e consistenti.

Seppure il patrimonio letterario arabo è stato raramente interessato da lavori o implementazioni specifiche, la variante di arabo classico-letterario costituisce l'oggetto di alcuni studi specifici, tra cui due modelli per la traduzione automatica dall'arabo classico all'inglese basato su regole linguistiche (Hebresha & Ab Aziz, 2013; Amna, 2011), un modello semi-automatico di valutazione della pertinenza e correttezza delle traduzioni inglese del Corano (Shenassa, 2008) , e un sistema di *information retrieval* per l'identificazione dei siti internet che contengono il testo coranico (Fauzan & Othman, 2006).

1.6.1 Un corpus interattivo di arabo coranico

I lavori descritti al paragrafo precedente sono principalmente concentrati sul corpus coranico ma si occupano più di tecniche di estrazione di informazione piuttosto che di analisi linguistica vera e propria . In un'altra prospettiva, il progetto *Quranic Arabic Corpus*[14] sfrutta parzialmente un paradigma di ricerca che è comune a questa dissertazione. Il testo arabo, in questo caso il Corano, è qui trattato non solo come oggetto di analisi ma fornisce le basi e l'ispirazione per la modellazione del processo di elaborazione stesso. Il corpus annotato e l'interfaccia grafica sviluppata consentono la navigazione da parte dell'utente nel testo coranico, passando per i piani morfologico, sintattico, semantico e traduttivo (Dukes, Atwell, & Sharaf, 2010).

Sul piano morfologico, ogni parola coranica è annotata con le relative informazioni morfologiche e presentata graficamente nella sua divisione in morfemi.

[14] http://corpus.quran.com [accesso il 10 marzo 2013]

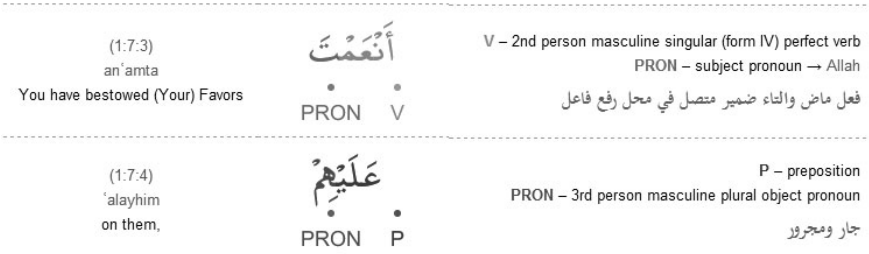


Figura 1.2: esempio di annotazione morfologica.
Fonte: http://corpus.quran.com/wordbyword.jsp [accesso il 10 marzo 2013]

Ciascuna parola è inoltre riconducibile alle proprie consonanti radicali, e una voce di un apposito dizionario coranico ne spiega in dettaglio il significato, ne elenca dinamicamente tutte le forme presenti nel Corano e ne consente il raggruppamento per radici. È possibile effettuare ricerche e generazioni di concordanze verbali, studi di frequenza sui lemmi e ricerche morfologiche avanzate.

Dal punto di vista sintattico il testo è annotato in modo da poter generare gli alberi sintattici di tutte le frasi coraniche:

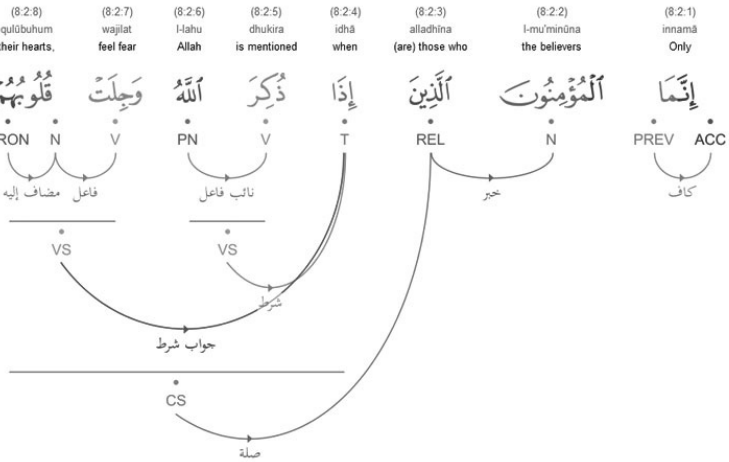


Figura 1.3: esempio di albero sintattico. Fonte: http://corpus.quran.com /treebank.jsp

Dal punto di vista semantico, il sistema usa la teoria di rappresentazione della conoscenza per definire i concetti chiave del Corano e le relative interrelazioni, usando la logica dei predicati. Ogni concetto è posto quindi in relazione grafica con gli altri, secondo i paradigmi di costruzione delle ontologie.

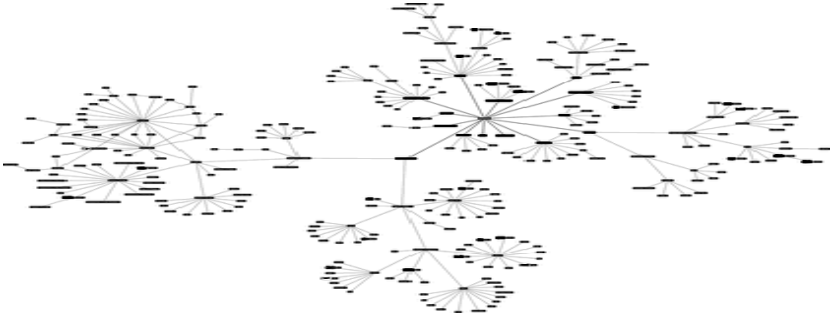


Figura 1.4: rappresentazione dell'ontologia coranica, in cui ogni nodo del grafo rappresenta un concetto e ogni segmento un collegamento di tipo iponimico-iperonimico. Fonte: http://corpus.quran.com/ontology.jsp

Infine, l'allineamento del testo arabo a sei traduzioni in inglese è consultabile per ogni versetto coranico.

2. Il testo : le collezioni di *ḥadīṯ*

2.1 *Ḥadīṯ* e tradizione

2.1.1 Definizione

La parola araba *ḥadīṯ* significa generalmente 'narrazione, racconto, informazione, notizia, leggenda, tradizione' (Traini, 1993, p. 196). La radice lessicale della parola è composta dalle tre consonanti *ḥ d ṯ*, associate al significato di 'accadere, succedere', da cui derivano i sensi secondari di 'raccontare un fatto, parlare di, relazionare su'.

Dal punto di vista degli studi arabo-islamici identifica un testo narrativo, di lunghezza variabile, che registra e trasmette una tradizione connessa con la vita di Muḥammad (571-632), l'ultimo profeta dell'Islam. Tale narrazione è un breve resoconto di qualcosa che Muḥammad ha detto, fatto, a cui ha assistito o che ha approvato tacitamente (Robson J. , 1978, p. 23)[15].

La stessa parola, talvolta scritta con l'iniziale maiuscola (in caratteri latini) o al plurale (ad es. ar. *'aḥadīṯ*, ingl. *ḥadīṯs*), può anche riferirsi a una raccolta di queste narrazioni. Sin dal primo secolo dell'era musulmana infatti, esse venivano solitamente radunate dai compilatori in collezioni.

Una terza estensione del significato di *ḥadīṯ*[16] è quella di indicare l'insieme della letteratura della tradizione islamica (composta appunto da

[15] Al di fuori di questo contesto, il ḥadīṯ identifica anche una tipologia narrativa specifica, affine alla forma letteraria dello *ḫabar* (Leder & & Kilpatrick, 1992) e indica qualsiasi tipo di informazione trasmessa cronologicamente su un qualche soggetto in cui esista un'attenzione particolare per la veridicità e l'integrità della trasmissione stessa. Molti generi della letteratura arabo-islamica hanno utilizzato questa forma, ad esempio biografie, resoconti geografici, opere lessicografiche (Burton, 1994, p. 29; Speight, 2000).

[16] Burton assegna al termine *'Ḥadīṯ'* con la prima lettera maiuscola proprio questo significato, vale a dire "the whole science or literature of the Islamic tradition", laddove inve-

tutte le narrazioni relative a Muḥammad) e talvolta anche le relative scienze di trasmissione, interpretazione e validazione (Burton, 1994, p. 198).

Infine con *Ḥadīṯ* si può anche intendere la *sunna,* letteralmente 'modo di vita', che costituisce il sistema norme, basate sulla tradizione, al quale il fedele deve conformarsi[17].

Riassumendo, *ḥadīṯ* può identificare la singola narrazione, un gruppo di esse, l'intero corpus che ne comprende la totalità, la scienza che li ha elaborati e studiati, fino a includere l'insieme della Tradizione islamica[18].

2.1.2 Natura e importanza

Burton afferma che il Corano è per i musulmani ciò che Cristo è per i cristiani. Se nel cristianesimo i dettagli sulla vita del fondatore della nuova religione sono contenuti in una parte dei testi sacri stessi (i Vangeli), nel Corano ci sono pochissimi accenni diretti a Muhammad, e la descrizione del modello di vita del profeta al quale conformarsi è contenuta in altri testi paralleli (Burton, 1994, p. 17). I *ḥadīṯ* rappresentano quindi una sorta di biografia del profeta Muḥammad perpetuata attraverso la memoria della sua comunità, in modo da servire come modello esemplificativo e comportamentale. I *ḥadīṯ* sono per i musulmani una delle principali fonti, seconda solo al Corano, per l'elaborazione della *sharīʿa,* la legge islamica. Integrando i precetti del Libro Sacro e cercando di illuminare le zone d'ombra coraniche, i *ḥadīṯ* riferiscono il comportamento e il pensiero di Muḥammad, aiutando a definire ciò che per un credente è obbligatorio, meritevole, tolle-

---

ce la forma in caratteri minuscoli identifica "a report of religious or legal significance" (Burton, 1994, pp. ix-x).

[17] Burton definisce più ampiamente la *sunna* come "hence, use and custom of any group of Arabs" (Burton, 1994, p. 201). EI specifica invece "normative custom of the Phrophet or of the early community" (Robson J. , 1978, p. 23).

[18] Salvo indicazione contraria in questo testo con i termini '*ḥadīṯ*' e 'tradizione' si indicheranno le narrazioni, in quanto singoli brani o raggruppamenti di essi. Per evitare appesantimenti, laddove il termine sia richiesto al plurale, verrà conservata la forma '*ḥadīṯ*' al posto del plurale arabo '*aḥadīṯ*' oppure Con 'Tradizione' si intenderà il corpus dei ḥadīṯ nella sua totalità, sia nella sua fase di elaborazione sia in quella più matura di fonte principale di elaborazione del diritto e della *sunna.*

rato, sconsigliato o vietato, e fornendo dettagli per la costruzione di norme regolanti tutti gli aspetti della vita terrena in preparazione della vita ultraterrena (Robson J. , 1978).

2.1.3 Origini

Il testo coranico, in quanto parola divina rivelata, e in analogia con le sacre scritture di altri sistemi religiosi, tratta molti argomenti ma non esaurisce tutti gli aspetti relativi al comportamento umano, non essendo per sua natura una raccolta di disposizioni giuridiche. L'Islam sin dalle origini ha privilegiato, accanto al rapporto diretto e privato tra l'essere umano e Dio, una dimensione comunitaria della religione, nella quale il pio credente è colui che il più possibile uniforma il proprio comportamento al volere divino. Nell'Islam il diritto, fatto sociale benché divino nelle origini, trova nella convivenza umana la sua ragion d'essere (Vacca, Noja, & Vallaro, 2009). Muḥammad non fu solo il messaggero esclusivo della rivelazione divina conclusiva, ma anche il capo e l'organizzatore della nascente comunità musulmana. Con lui in vita, i fedeli non dovevano preoccuparsi di definire gli elementi caratteristici del buon musulmano, era sufficiente guardare a lui come esempio e modello da seguire in ogni suo comportamento e azione.

La morte di Muḥammad costituì quindi un evento sensazionale nel nascente Islam, fu la fine della possibilità di ricevere ulteriori rivelazioni divine e insieme la scomparsa di colui il cui comportamento guidava e uniformava quello personale e comunitario di tutti i fedeli. Il Corano era indubbiamente la fonte prioritaria di guida e solo gradualmente, al sorgere di nuovi problemi, si sentì il bisogno di un'autorità sussidiaria che integrasse i precetti coranici. La rapida espansione politica e geografica dell'Islam richiese infatti l'elaborazione di un complesso sistema di norme che regolassero tutti gli aspetti della vita privata e pubblica di individui e istituzioni. Questo sistema aveva necessariamente e urgentemente bisogno di ricorrere a fonti per l'elaborazione del diritto, identificate nel Corano e nell'esempio della vita di Muḥammad. Ma alla morte del profeta tali né una né l'altra fonte esistevano ancora in una forma scritta definitiva e formalizzata tale da costituire un canone.

Il processo di fissazione in canone del Corano e della Tradizione del profeta avvengono con modalità e tempi differenti. Il Corano, in quanto parola rivelata era per definizione unico e non ambiguo nella sua formulazione divina e nella sua trasmissione profetica. Da un punto di vista musulmano si trattava in sostanza di ovviare alla difficoltà umana di fissare una volta per tutte un messaggio divino, per sua natura assolutamente univoco e perfetto nella sua emanazione.

Per quanto riguarda la codificazione di una fonte attendibile sulla vita di Muḥammad, essa non poteva che riferirsi alle testimonianze di coloro che gli erano stati vicini e lo conoscevano, i cosiddetti Compagni. I *ḥadīṯ* nascono proprio come testimonianze orali e poi scritte dei comportamenti e delle azioni di Muḥammad direttamente osservati da coloro che gli erano prossimi e familiari. Il processo di canonizzazione di tali fonti fu però piuttosto lungo e solo col tempo acquisì sofisticatezza di indagine e alto grado di formalizzazione[19]. Se infatti per i primi musulmani la formulazione scritta del Corano trovava la sua ovvia giustificazione nell'esigenza di fissare una rivelazione già di per sé univoca, la riluttanza a registrare parole e atti del profeta, in qualche modo sacralizzandoli e accostandoli alla Scrittura, era più sensibile. Il fatto che Muḥammad disapprovasse tale pratica è comunque una prova della sua diffusione[20]. Dopo la morte di Muḥammad, parecchi studiosi svilupparono un crescente interesse per la Tradizione, e molti di questi tradizionisti viaggiarono in tutto il territorio musulmano alla ricerca di testimonianze e resoconti. Il pellegrinaggio rituale alla Mecca era un'occasione importante per lo scambio e la circolazione di queste tradizioni. Tra i primi studiosi che procedettero alla registrazione per iscritto spiccano a Medina 'Urwa Ibn al-Zubayr (m. 712-717 d.C.) che raccolse le tradizioni trasmesse da 'Ā'isha, una delle mogli di Muḥammad, e in Siria Muḥammad Ibn Muslim Ibn Šihāb al-Zuhri (m. 741).

[19] Inoltre, al di fuori della comunità musulmana, che da sempre riconosce l'autenticità delle raccolte canoniche seppur in gradazioni differenti, il dibattito sulla veridicità e l'effettivo riferimento storico di questa tradizione è lungi dall'essere concluso.
[20] Abū Dā'ūd riporta due *ḥadīṯ* che contengono l'uno il consenso l'altro il diniego profetico a registrare le tradizioni per iscritto (Robson J. , 1978, p. 24).

2.1.4 Formazione del canone

Durante primo secolo dalla morte di Muḥammad, il riferimento alla Tradizione cominciò lentamente a configurarsi come fattore centrale per l'elaborazione della legge e nella modellazione della società islamica. Ma occorsero almeno altri cent'anni affinché le decine di migliaia di *ḥadīṯ* circolanti fossero sottoposte a un processo di selezione e verifica dell'attendibilità e della veridicità. Nacquero alcune discipline attraverso le quali gli esperti tentavano di classificare i vari *ḥadīṯ* in base alla loro attendibilità e di fornire per quelli considerati autentici un'attestazione di veridicità. Questo avveniva principalmente attraverso l'asseverazione del processo di trasmissione orale o scritta di un fatto della vita di Muḥammad, dai testimoni oculari via via nel tempo fino al trasmettitore più recente per cronologia storica. Tra le prime opere che mostrano uno sviluppo embrionale delle scienze di validazione spiccano la biografia di Muḥammad a opera di Ibn Isḥāq (m. 767-768 d.C.), nella quale sono riportate alcune tradizioni ma con parziali catene di trasmissione, e *Al-muwaṭṭa'* di Mālik ibn Anas, un testo giuridico le cui tradizioni mostrano o meno, in modo discontinuo, la presenza di indicazioni sui relativi trasmettitori (m. 795 d.C.).

Solo nel secolo successivo il perfezionamento delle scienze teoriche di studio e il contributo di grandi giuristi come al-Shafi'ī (m. 820) resero i *ḥadīṯ* una fonte indispensabile e riconosciuta per la formulazione del diritto islamico (Powers D. S., 1986). Alla fine del III secolo dall'Egira, la reputazione e l'autorità delle collezioni riconosciute come canoniche si stabilizzò definitivamente, consolidando un corpus di tradizioni oramai difficilmente modificabile con l'aggiunta di nuovo materiale, e dal quale eccentricità e stravaganze erano state definitivamente espunte. Le tradizioni erano infine diventate a tutti gli effetti un elemento permanente e disciplinato della struttura autoritativa islamica, il secondo in ordine di importanza dopo il Corano, e precedente gli strumenti giuridici del ragionamento analogico (*qiyās*) e del consenso (*iǧmā'*), fonti ulteriori di legislazione.

2.1.5 Riunire più *ḥadīṯ*: le principali collezioni

Lo sviluppo delle scienze di verificazione della tradizione consolidò la pratica di riunire i *ḥadīṯ* in collezioni che avevano come caratteristica quella di essere state redatte da un unico tradizionista in osservanza ai criteri di validazione che dette scienze avevano stabilito (Brown, 2007).

Le prime raccolte vennero chiamate *musnad* ('sostenute') in quanto i *ḥadīṯ* al loro interno erano raggruppati insieme sotto il nome dei rispettivi testimoni oculari i Compagni del profeta. In seguito, affinché per i giuristi fosse più agevole trovare i *ḥadīṯ* relativi alla questione legale che li interessava, si affermò la consuetudine di utilizzare una classificazione che organizzasse le varie tradizioni per argomento ('*ala al-abwāb*', 'secondo capitoli'), per cui le raccolte acquisirono il nome di *musannaf* ('classificato'). Le raccolte di al-Ṭayālisī (m. 818 d.C.) e Aḥmad Ibn Ḥanbal e il *Muwaṭṭa'* di Mālik sono le prime a mostrare tale classificazione.

Nel IX secolo alcune di queste raccolte, grazie al consenso che godevano i rispettivi autori circa il rigore scientifico e la serietà di verifica delle fonti di trasmissione, emersero sulle altre e acquisirono la dignità di autorità riconosciuta e canone[21], seppur con alcune differenze nelle comunità musulmane che non si riconoscevano pienamente nella *sunna*, ad esempio gli sciiti. Tali raccolte comprendevano solo *ḥadīṯ* ritenuti autentici oppure, in caso contrario, classificavano comunque ogni tradizione in base alla sua autenticità (Melchert, 2001).

La Tradizione considera come raccolte canoniche sei principali *musannafāt*, sebbene l'accordo pressoché unanime sulla piena autenticità dei *ḥadīṯ* contenuti ci sia solo sulle prime due, i *Ṣaḥīḥ* di al-Buḫārī e Muslim.

2.1.5.1 *Al-ǧāmiʿ al-ṣaḥīḥ* di al-Buḫārī

Abū ʿAbd Allāh Muḥammad ibn Ismāʿīl al-Buḫārī (810-870 d.C.) è considerato quasi unanimemente dai sunniti come il più riverito e stimato compilatore di *ḥadīṯ* e studioso di scienze della tradizione. Nato a Bukhara in Asia Centra-

[21] In assenza di un'istituzione che commissionasse o supervisionasse il canone, questo divenne tale per il graduale addensarsi del consenso della comunità intorno ad alcune opere piuttosto che altre (Robson J. , 1978, p. 24).

le passò la propria vita a studiare scrupolosamente le tradizioni del profeta, senza rinunciare a lunghi viaggi nell'impero musulmano per reperire informazioni sui vari trasmettitori e sulla loro affidabilità. Raccolse e sottopose a verifica circa 600.000 *ḥadīṯ*, ne selezionò 7397 ritenuti autentici[22], li dotò di *isnād* dettagliati e li compilò infine nel *Al-ǧāmiʿ al-ṣaḥīḥ* ('la collezione autentica'), la sua opera principale[23], che richiese 16 anni di redazione. L'opera è un tipico *musannaf* in cui le varie sezioni si susseguono ordinate per argomenti. Questi non si limitano alla sfera giuridica e normativa, ma comprendono anche materiale biografico e commentari coranici.

2.1.5.2 Il *Ṣaḥīḥ* di Muslim

Considerato di statura analoga ad al-Buḫārī, Abū al-Ḥusayn Muslim Ibn al-Ḥaǧǧāǧ al-Qushayrī (817-875 d.C.), persiano di origini, è l'autore del *Ṣaḥīḥ*, una raccolta di *ḥadīṯ* preceduta da un'introduzione sui criteri metodologici di selezione. Il materiale raccolto è in larga misura analogo a quello dei principali compilatori sui contemporanei, ed è caratterizzato da un uso sensibile del consenso comune e locale come strumento di asseverazione delle catene di trasmissione.

2.1.5.3 Il *Kitāb al-sunan* di Abū Dāʾūd

Abū Dāʾūd al-Siǧistānī (m. 888 d.C.) raccolse nell'opera *Kitāb al-sunan* ('libro delle tradizioni) 4800 *ḥadīṯ* riguardanti soggetti prevalentemente giuridici (come suggerisce l'uso nel titolo del termine *sunan* ('tradizioni' come fondamento di un codice di comportamento) piuttosto che *al-ǧāmiʿ* utilizzato ad esempio da al-Buḫārī per indicare la varietà degli argomenti trattati).

2.1.5.4 *Al-ǧāmiʿ al-ṣaḥīḥ* di al-Tirmiḏī

Abū ʿIsā Muḥammad al-Tirmiḏī (m. 892 d.C.), analogamente ad al-Buḫārī intitolò la sua raccolta *Al-Ǧāmiʿ al-ṣaḥīḥ*, corredandola di note sulle differenze

[22] 2762 se si unificano nel conteggio i ḥadīṯ che riportano il medesimo contenuto.
[23] A guisa d'introduzione della sua collezione scrisse anche *Al-Taʾrīkh al-kabīr* ('la grande storia'), una raccolta di biografie e notizie relative ai personaggi coinvolti nella trasmissione dei ḥadīṯ.

interpretative delle quattro scuole giuridiche sunnite (*madhāhib*) e di un capitolo finale sulle tecniche di validazione.

2.1.5.5 Il *Kitāb al-sunan* di al-Nasāʾī

Abūʿ Abd al-Raḥmān al-Nasāʾī (830–915 d.C.) usò per la sua collezione lo stesso titolo di Abū Dāʾūd, *Kitāb al-sunan,* e le tradizioni ivi raccolte concernono in modo particolare il diritto religioso specifico per gli atti di rito e devozione.

2.1.5.6 Il *Kitāb al-sunan* di Ibn Māğā

Abū ʿAbdallāh ibn Māğā (824-886 d.C.), discepolo di al-Siğistānī, compilò un *Kitāb al-sunan* in cui la i criteri di selezione dei *ḥadīṯ* tendono a essere in una certa misura meno rigorosi di quelli degli altri cinque compilatori. Per tale ragione essa ebbe difficoltà a diventare parte del canone e ancora nel XIV secolo alcuni studiosi come Ibn Ḫaldūn (m. 1406) la espungevano dal novero e altri la sostituivano con il *Muwaṭṭa'* di Mālik.

2.1.5.7 Altre collezioni

Accanto alle sei precedenti collezioni, considerate in qualche modo un corpus canonico omogeneo e consultabile in tutte le sue parti, esistono altre raccolte, variamente considerate sul piano giuridico ma spesso celebri a livello popolare e pietistico. Tra queste spicca la collezione di *ḥadīṯ* di Mālik ibn Anas, precedente per redazione a tutte le altre e molto conosciuta, anche se, per via del minore livello di formalizzazione e applicazione delle scienze di verifica non sempre considerata fonte autoritativa e legale.

Raccolte commentate e digesti di *ḥadīṯ* basati sulle collezioni canoniche, ebbero fortuna nei secoli successivi soprattutto come ausili didattici per l'educazione religiosa e popolare[24]. Tra le più celebri vi è l'opera *Maṣābīḥ al-Sunna* ('le lampade della Sunna') di Abū Muḥammad al-Baghawī (m. 1122 d.C.), in seguito espansa e arricchita da Walī al-Dīn nell'ancora più popolare *Mishkāt al- Maṣābīḥ*.

[24] Particolare fortuna ebbero le selezioni di 40 *ḥadīṯ* su di un unico argomento , conosciute con il nome di *'Arba'ūn* ('quaranta').

2.1.6 *Ši ʻa* e tradizione

In ambito sciita, la più significativa tra le confessioni minoritarie non sunnite dell'Islam, le tradizioni relative al profeta esistono ma si discostano da quelle sunnite principalmente per il ruolo in esse rivestito da ʻAlī, cugino di Muḥammad e figura centrale dello sciismo. Gli sciiti non riconoscono quindi le collezioni canoniche dell'Islam sunnita e hanno codificato in modo autonomo e indipendente un proprio corpus di *ḥadīṯ*. Qui l'importanza dell'*isnād* e del consenso locale nel confermare l'autenticità di una tradizione (le basi della tradizione sunnita), vengono posti in secondo piano rispetto all'autorità dell'Imam, che rappresenta uno degli elementi cardinali dello sciismo stesso. I concetti di tradizione e trasmissione dell'informazione vengono pertanto ridimensionati e reinterpretati alla luce del primato dell'opinione imamita. Le principali compilazioni di tradizioni sciite sono relativamente tarde (IV e V sec. E.) e consentono solo quei *ḥadīṯ* provenienti dall'imamato e dalla famiglia alide. Tra le principali organizzate come *musannafāt*, ci sono il *Kāfī fī ʿilm al-dīn* ('tutto ciò che serve sapere sulle scienza della pratica religiosa') di Abū Ǧaʿfar Muḥammad Ibn Yaʻqūb al-Qulīnī (m. 938 d.C.), il *Kitāb man lā yaḥduruhu al-faqīh*, di Abū Ǧaʿfar Muḥammad Ibn ʻAlī detto al-Bābuya al-Kummī (m. 991 d.C.), il Taḥḏīb al-Aḥkām di Abū Ǧaʿfar Muḥammad Ibn al-Ḥasan al-Ṭūsī (m. 1067-68 d.C.)

2.2 *Ḥadīṯ*: struttura e composizione

2.2.1 Aspetto e struttura

Dal punto di vista dell'organizzazione strutturale interna, un *ḥadīṯ* è tipicamente (ma non necessariamente) formato da due parti, l'*isnād* e il *matn.*
Le due parti confluiscono naturalmente una nell'altra in una successione ininterrotta. Il cambio di sezione non è chiaramente espresso con marcatori grafici specifici, e si serve piuttosto di elementi lessicali più o meno riservati.

2.2.1 L' *isnād*

L'*isnād*, letteralmente 'supporto' (sul quale l'informazione poggia), costituisce la prima parte di un *ḥadīṯ* e ha come finalità quella di attestare la provenienza e il percorso dell'informazione stessa attraverso l'elenco dei testimoni oculari e dei successivi trasmettitori. È quindi tipicamente formato da una lista di nomi dei responsabili della registrazione e della trasmissione del *matn*, in ordine dal più recente al più remoto. L'esempio seguente mostra un *isnād* tratto dal primo ḥadīṯ del *Ṣaḥīḥ di* al-Buḫārī *:*

حَدَّثَنَا الْحُمَيْدِيُّ عَبْدُ اللَّهِ بْنُ الزُّبَيْرِ قَالَ حَدَّثَنَا سُفْيَانُ قَالَ حَدَّثَنَا يَحْيَى بْنُ سَعِيدٍ الْأَنْصَارِيُّ قَالَ أَخْبَرَنِي مُحَمَّدُ بْنُ إِبْرَاهِيمَ التَّيْمِيُّ أَنَّهُ سَمِعَ عَلْقَمَةَ بْنَ وَقَّاصٍ اللَّيْثِيَّ يَقُولُ سَمِعْتُ عُمَرَ بْنَ الْخَطَّابِ رَضِيَ اللَّهُ عَنْهُ عَلَى الْمِنْبَرِ قَالَ[...]

Al-Ḥumaydī Abdullah bin al-zubayr ci ha narrato che Sufyān disse che Yaḥya bin Sa'īd al-'anṣārī narrò loro di essere stato informato da Muḥammad bin 'Ibrāhīm al-Taymī sul fatto che lui sentì 'Alqama bin Waqqāṣ al-Layṯī dire "Ho sentito dal pulpito 'Umar bin al-Ḫattāb, che Dio si compiaccia di lui, dire [...]"

2.2.1.1 I marcatori di tipologia di trasmissione

I nomi dei trasmettitori sono preceduti o seguiti da verbi o preposizioni che esprimono in modo parzialmente formalizzato come la trasmissione stessa sia avvenuta (oralmente, per iscritto, in forma privata o pubblica). Le espressioni più comuni comprendono:

- *ḥaddaṯanī* 'mi ha raccontato',
- *ḥaddaṯanā* 'ci ha raccontato',
- *aḫbaranī* 'mi ha informato',
- *aḫbaranā* 'ci ha informato',
- *sami'tu* 'ho ascoltato',
- *anba'anī* 'mi ha annunciato',
- *anba'anā* 'ci ha annunciato'',
- *'an*, 'in base all'autorità di'

Sembra accertato che almeno fino alla fine del IX secolo d.C. non ci fosse consenso tra gli esperti sulla corrispondenza biunivoca tra ciascun termine e un particolare metodo di trasmissione tra quelli classificati dalle scienze della Tradizione (Robson J. , 1978, p. 27), per cui non è possibile utilizzare

questa lista come un chiaro discrimine circa la tipologia utilizzata nella trasmissione.

Più rigorosamente, per al-Ḥakīm *ḥaddaṯanī* è da usarsi quando al momento della trasmissione orale non è presente nessuno al di fuori del maestro e del discepolo, in tal caso è da preferirgli l'espressione *ḥaddaṯanā.* Analogamente, *aḫbaranī* indicherebbe esclusivamente la recitazione privata di un discepolo al maestro, mentre *aḫbaranā* sottolineerebbe il carattere pubblico della recitazione, con almeno un altro testimone oltre al ricevente. A*nba'anī* sarebbe invece riservato alla generica sottomissione di tradizioni allo *šayḫ*, previa concessione della *iǧāza*.

Il significato più generico è quello espresso dalla preposizione *'an*, che riferisce semplicemente la trasmissione all'autorità del trasmettitore precedente. *'an* è solitamente usato o quando l'affidabilità del trasmettitore precedente è certa (al-Buḫārī ne fa un vasto impiego), oppure per ragioni di concisione testuale nel caso di *isnād* particolarmente estesi, come rilevato da al-Ḫaṭīb al-Baġdādī (m.1071).

2.2.1.2. L'organizzazione logica dell'*isnād*

Traducendo l'*isnād* precedente non con un discorso indiretto ma privilegiando l'uso del discorso diretto e dei relativi segni di punteggiatura,

> Al-Ḥumaydī Abdullah bin Al-Zubayr ci ha narrato: "Sufyān disse: "Ci ha narrato Yaḥya bin Sa'īd Al-'anṣārī: "Mi ha informato Muḥammad bin 'Ibrāhīm Al-Taymī che lui sentì 'Alqama bin Waqqāṣ Al-Laytī dire: "Ho udito 'Umar bin Al-Ḫattāb, che Dio si compiaccia di lui, dire dal pulpito [...]"

Utilizzando delle parentesi quadre indicizzate numericamente per delimitare i vari livelli di trasmissione, è possibile esplicitare ulteriormente una struttura di tipo parentetico ricorsivo (a 'buccia di cipolla' o a 'matrioska'), dove, partendo dallo strato più esterno (e più recente), si entra man mano nei livelli interni, fino ad arrivare al *matn* stesso:

> $_{7}$[Al-Ḥumaydī Abdullah bin al-Zubayr ci ha narrato: $_{6}$[Sufyān disse: $_{5}$[Ci ha narrato Yaḥya bin Sa'īd al-'anṣārī: $_{4}$[mi ha informato Muḥammad bin 'Ibrāhīm al-Taymī che $_{3}$[lui sentì 'Alqama bin Waqqāṣ al-Laytī dire: $_{2}$[Ho udito 'Umar bin al-Ḫattāb, che Dio si compiaccia di lui, dire dal pulpito che $_{1}$[*matn*]$_{1}$]$_{2}$]$_{3}$]$_{4}$]$_{5}$]$_{6}$]$_{7}$"

Un'altra rappresentazione possibile è quella lineare monodimensionale, nella quale il contenuto giunge al compilatore attraverso un'ideale linea di trasmissione contenente i trasmettitori:

COMPILATORE ←T7 ←T6←T5←T4←T3←T2← CONTENUTO

Queste rappresentazioni dell'organizzazione logica interna del *ḥadīṯ* saranno riprese in dettaglio e ampliate al titolo 4.2.2.

2.2.2 Il *matn*

Rappresenta il contenuto stesso dell'informazione veicolata dal *ḥadīṯ*. I temi delle collezioni canoniche di *ḥadīṯ* coprono un ampio ventaglio di argomenti, quali ritualità, diritto e procedura civile e penale e teologia, tecnica ed esegesi, nel tentativo di includere escatologicamente ogni aspetto della vita dell'essere umano nella sua dimensione spirituale, morale, comportamentale, privata, pubblica, lasciando spazio all'eterogeneità e all'enciclopedismo dei soggetti trattati (Burton, 1994; Fück J. , 1938; Juynboll, 2001; Lucas, 2002).

La valutazione delle caratteristiche letterarie del contenuto dei *matn* è sempre stata minoritaria rispetto alle considerazioni di tipo giuridico-normativo, religioso o di critica delle fonti, anche se già Ibn Qutaiba nel suo commentario *Kitāb ṭa'wil muḫtalif al-ḥadit* e poi al-Raḍī nel *al-Maǧazāt al-nabawiyya* ne avevano tentato un approccio più strutturato. Tra gli studi recenti, si ricordano l'analisi delle tipologie di discorso narrativo contenuto nel matn (Speight, 2000) e alcune considerazioni sui *ḥadīṯ* come genere letterario (Leder & Kilpatrick, Classical arabic prose literature: a researchers' sketch map, 1992).

2.3 Le studio e la trasmissione della Tradizione

2.3.1 Le scienze di registrazione e trasmissione

Gli studi della Tradizione nati e sviluppatisi all'interno della comunità musulmana sono conosciuti sotto il nome originario di *'ulūm al-ḥadīṯ*

('scienze della Tradizione'). Tali opere riunivano in formato parzialmente enciclopedico, le classificazioni delle tradizioni, le notizie biografiche sulle varie categorie di trasmettitori, prime fra tutto quelle comprendenti i Compagni del profeta, nozioni e tecniche per il recepimento e la trasmissione delle tradizioni, dettagli sul reperimento delle fonti, norme ortografiche e strutturali per una chiara e corretta registrazione manoscritta (Robson J. , 1961).

Tra le prime opere che affrontano questi temi in modo sistematico troviamo *al-Muḥaddiṯ al-fāṣil bayna l-rāwī wa-l-wā'ī* di Abū Muḥammad al-Ramahurmuzī (m. 971 d.C.) e la *Ma 'rifa 'ulūm al-ḥadīṯ* di al-Ḥākim al-Naysābūrī, che presenta una suddivisione in 52 categorie modello per le opere successive. Sessantacinque categorie mostra invece lo *'Ulūm al-ḥadīṯ* di Ibn al-Ṣalāḥ, considerato il lavoro fondamentale di questo genere.

2.3.1.1 Tecniche di ricezione e trasmissione della tradizione

Sebbene gli studiosi concordassero nella classificazione dei diversi metodi con i quali una tradizione veniva lecitamente trasmessa da un trasmettitore a un altro, manca l'unanimità sul grado di importanza di una tecnica rispetto a un'altra.

Per Ibn al-Ṣalāḥ il metodo più prestigioso è quello dell'ascolto diretto (*samā'*), dove un ricevitore (discepolo) ascolta la tradizione direttamente da un discorso orale del trasmettitore precedente (*šayḫ*, 'maestro').

Seconda per importanza è la recitazione davanti al maestro (*al-qirā'a 'ala l-šayḫ*, anche conosciuta come *'arḍ*, 'sottomissione [di qc. a qn.]'), nella quale il ricevitore legge o recita a memoria la tradizione al trasmettitore e questi la conferma grazie al proprio materiale scritto o alla propria capacità mnemonica.

Il terzo metodo è la *iǧāza*, 'licenza', e consiste nella concessione da parte del trasmettitore dell'autorizzazione a trasmettere il matn ad altri, la quale poi può avvenire con modalità differenti.

Il metodo della *munāwala* prevede che una copia scritta delle tradizioni di uno *šayḫ* sia consegnata al ricevente, anche se alcuni studiosi riconoscono questa tecnica solo se accompagnata dalla *iǧāza* dello stesso *šayḫ*. Questo

metodo è in qualche misura analogo a quello della *mukātaba* (‘corrisponden-(‘corrispondenza’), dove la tecnica di trasmissione si concentra sul registro scritto.

Tra i metodi considerati da alcuni validi e da altri no (Ibn al-Ṣalāḥ per esempio), spiccano il ‘lascito’ (*waṣiyya*), dove il discepolo ottiene in eredità dal maestro il testo contenente la tradizione, e il ‘ritrovamento’ (*wiǧāda*) dove le tradizioni sono scoperte per iniziativa del discepolo nei manoscritti di un maestro. Questi metodi trovano poi una corrispondenza, seppure non sistematica, nella descrizione della terminologia effettivamente usata nella redazione dell’*isnād* per collegare linguisticamente un trasmettitore all’altro.

Un importante argomento relativo alle tecniche di trasmissione riguarda la necessità di assicurarsi di avere un *isnād* affidabile non solo per le singole tradizioni, ma anche per le collezioni di queste, questo soprattutto fino a che non si diffuse la stampa e la collazione tipografica. Ogni manoscritto contenente gruppi di tradizioni doveva quindi essere certificato anch’esso allo stesso modo dei singoli *ḥadīṯ*, attraverso la conferma della correttezza della catena di trasmissione.

Altre discussioni di metodo riguardavano i limiti massimi e minimi di età per accettare un individuo come trasmettitore, ma ad esempio Ibn Ṣalāḥ riteneva che non avesse senso stabilire dei limiti precisi (cinque anni come soglia minima e ottanta come massima) validi in tutte le circostanze, poiché la maturità e il decadimento degli esseri umani sono criteri troppo variabili da persona a persona per essere formalizzati.

2.3.1.2 Discordanze e contraddizioni

L’inconsistenza di alcune tradizioni (*muḫtalif al- ḥadīṯ*) riconosciute come autentiche, vale a dire la discrepanza o talvolta la contraddizione evidente fra due *ḥadīṯ*, fu un altro rilevante problema affrontato dalle scienze della tradizione. Alla fine del XIX secolo il *Ta’wīl muḫtalif al- ḥadīṯ* di Ibn Qutayba (m. 1908) tratta di questo tema in modo dettagliato, sostenendo o una possibile riconciliazione dando la prevalenza alla tradizione i cui trasmettitori sono più affidabili o prevedendo che la tradizione più recente

possa abrogare la tradizione più antica su modello della disciplina coranica dell'abrogante e dell'abrogato (*al-nāsiḫ wa-l-mansūḫ*).

Proprio negli anni in cui Ibn Qutayba tentava la strada della conciliazione tra le tradizioni discordanti, Goldziher in Europa affrontava lo stesso problema, ma usando invece le contraddizioni come una delle giustificazioni per sostenere l'artificiosità di gran parte della Tradizione (cfr. titolo 2.4.1).

2.3.2 Le scienze di critica e verifica

In considerazione del carattere autoritativo e vincolante delle tradizioni canoniche nell'elaborazione del diritto islamico è evidente che una delle preoccupazioni principali dei primi giurisperiti musulmani fosse quella di poter discernere con certezza tra i *ḥadīṯ* esistenti, quelli autentici da quelli falsi o fabbricati (Burton, 1994, p. 106). Prima che il canone della Tradizione si affermasse, il corpus dei *ḥadīṯ* era aumentato vertiginosamente, e molti seri studiosi riconoscevano che la gran parte di essi fossero fabbricati. Tali invenzioni potevano servire a corroborare tesi contrarie all'ortodossia o, all'opposto, anche a esortare i credenti a comportarsi in modo ancora più pio (Robson J. , 1978, p. 25). Queste tendenze sono testimonianza di quanto la Tradizione stesse acquisendo un ruolo centrale nell'Islam (Burton, 1994, p. xiii).

All'interno della comunità islamica nacquero e si svilupparono quindi alcune discipline (*'ulūm al-ḥadīṯ*) in grado di fare luce sui 'conflitti d'informazione' che i vari *ḥadīṯ* presentavano al loro interno (dubbi sulla veridicità del contenuto) e nel rapporto reciproco (discordanze nel riportare uno stesso fatto).

L'analisi del contenuto dell'informazione di per se stessa (*matn*) al fine di trovare in essa la conferma o meno dell'autenticità fu però scarsamente perseguita, proprio perché avrebbe comportato l'uso di strumenti insidiosi come la soggettività di giudizio o la presupposizione personale dello

studioso[25]. Le scienze della tradizione si concentrarono piuttosto sull'analisi di dati oggettivi e storicamente valutabili, come la verifica dell'attendibilità sia dei presunti testimoni di un fatto della vita di Muḥammad sia di coloro che ne furono informati in un secondo tempo da terze parti (Philips, 2007).

La necessità di verificare l'attendibilità dei trasmettitori portò alla nascita di scienze biografiche specifiche (*'ulūm al-riğāl*, 'scienze degli uomini') in grado di analizzare i dati genealogici e le vite personali dei narratori di tradizioni. Ad esempio, un importante criterio di autenticità era verificare se i due individui uno all'altro successivi nella catena di trasmissione avessero avuto l'effettiva possibilità di conoscersi in vita. Tale verifica veniva effettuata sul piano temporale (confronto di date di nascita e morte) o su quello geografico (luoghi di residenza dei trasmettitori e loro viaggi o spostamenti). Altri studi di tipo comparativo cercavano di asseverare o meno la credibilità di un trasmettitore attraverso la disamina del suo *entourage*, delle sue opere o del giudizio dato su di lui dai contemporanei.

All'interno delle scienze della Tradizione si svilupparono classi di termini tecnici in grado di interpretare i *ḥadīṯ* secondo vari criteri e quindi classificarli in categorie, ciascuna riportante il livello di autenticità attribuito dagli studiosi. L'accordo generale sul significato e sui limiti della terminologia richiese parecchio tempo e soprattutto nelle fasi iniziali di applicazione, i termini potevano parzialmente sovrapporsi e non essere perfettamente complementari (Robson J. , 1978, p. 25).

2.3.2.1 Classificazione generale[26]

I *ḥadīṯ* sono classificati secondo tre categorie principali: *ṣaḥīḥ*, 'valido', *ḥasan*, 'buono' o *ḍāʿīf*, 'debole'[27]. Ciascuna delle tre categorie comprende

[25] Il caso più celebre di *ḥadīṯ* ritenuto falso unicamente sulla base dell'analisi del *matn* è quello che recita "Io sono il sigillo dei profeti; non ci sarà alcun profeta dopo di me a meno che Dio lo voglia". In questo caso la tradizione, pur corredata con un *isnād* molto solido, fu rifiutata proprio per l'implicazione eretica dell'espressione "a meno che Dio lo voglia" (Robson J. , 1978, p. 24).

[26] Per la classificazione generale e i termini utilizzati in questo titolo cfr. (Brown, 2007; Burton, 1994; Sezgin, 1996).

ulteriori distinzioni sulla base del perfezionamento delle scienze di verifica e, successivamente, in base all'aderenza con le compilazioni di tradizioni che divennero canone.

Le tradizioni di tipo *ṣaḥīḥ* sono ulteriormente articolate in sette gradi: (1) quelle riconosciute tali da al-Buḫārī e Muslim; (2) quelle considerate piena-pienamente valide solo da al-Buḫārī; (3) quelle considerate pienamente valide solo da Muslim; (4) quelle non riconosciute valide da al-Buḫārī e Muslim ma che tuttavia soddisfano le condizioni (*šurūṭ*) di autenticità poste da entrambi; (5) quelle che soddisfano solo le condizioni di al-Buḫārī; (6) quelle che soddisfano solo le condizioni di Muslim; (7) quelle riconosciute come valide da altri studiosi di tradizioni (*muḥaddiṯūna*) (Robson J. , 1978, p. 25).

Le tradizioni classificate come *ḥasan* costituiscono indubbiamente il gruppo numericamente più consistente, e seppur considerate inferiori a quelle *ṣaḥīḥ* furono comunque una base importante per l'elaborazione del diritto, soprattutto in quegli aspetti non illuminati dalle tradizioni *ṣaḥīḥ.* Anche in questa categoria esistono delle suddivisioni interne, ad esempio quelle usate da al-Tirmiḏī, ma esse mancano di elementi che le distinguano esplicitamente una dall'altra.

Le tradizioni *ḍāʿīf*, o deboli, seppur escluse nell'elaborazione delle norme giuridiche, possono essere tuttavia valide come esortazione morale o *exemplum,* oppure per la definizione di norme supererogatorie, ma che non comportino obbligatorietà o divieti.

La classificazione può basarsi anche su una serie di altri fattori legati principalmente all'*isnād* e quindi alle catene di trasmettitori.

2.3.2.2 Classificazione per numero dei trasmettitori

In considerazione del numero e dell'attendibilità dei trasmettitori, un *ḥadīṯ* può essere classificato come:

---

[27] Definizioni alternative prevedono termini quali *ṣāliḥ,* 'sano' e *saqīm* 'infermo'. Abū Da'ūd intende *con ṣāliḥ* quelle tradizioni sulle quali non esistono particolari critiche o osservazioni di non autenticità, altri riconoscono le tradizioni *ṣāliḥ* come intermedie tra quelle *ḥasan* e quelle *ḍāʿīf.*

1. *mutawātir*: con un alto numero di trasmettitori ritenuti affidabili (*ṣaḥīḥ*);
2. *mašhūr*: con più di due trasmettitori, di cui alcuni sicuramente affidabili;
3. *mustafīd*: un livello intermedio tra *mutawātir e mašhūr* o da alcuni associato a uno o all'altro;
4. '*azīz*, con un trasmettitore di tale autorità da condividere con almeno tre altri individui la trasmissione della medesima tradizione;
5. *ġarīb*: tra i trasmettitori originari è annoverato un solo *sāḥib* (Compagno del profeta) o trasmettitore[28];
6. *fard*: con un solo trasmettitore per ogni anello della catena di trasmissione, oppure la cui trasmissione è avvenuta unicamente in un' area geografica;
7. *šaḏḏ*: proveniente da una sola autorità e differente da ciò che altri riportano, e quindi passibile di essere accettato o rifiutato a seconda dell'affidabilità dell'autorità stessa nel trasmettere altre tradizioni;
8. (8) *āḥād*: con un basso numero di trasmettitori.

2.3.2.3 Classificazione per tipologia di *isnād*

Se è possibile tracciare il percorso di trasmissione senza interruzioni dal collezionatore fino all'origine, il *ḥadīṯ* è detto *muttaṣil, marfū'* se risale fino a Muḥammad o di *mawqūf* se è riconducibile a un Compagno o Successore di Muḥammad. Se una tradizione è sia *muttaṣil* sia *marfū'* è definita *musnad,* anche se esistono dei *musnad* che sono invece *mawqūf*. Le tradizioni che risalgono solo fino ai Successori vengono invece classificate *maqṭū'*, e questo termine e generalmente applicato a tutte le tradizioni in cui un anello della catena di trasmissione è interrotto, mancante o particolarmente debole. Un *ḥadīṯ* le cui interruzioni nell'*isnād* sono multiple può essere definito *munfaṣil* ('separato, diviso'). Un *ḥadīṯ* che riporta un *isnād* parziale o ne è sprovvisto del tutto è classificato come *mu'allaq*, ('sospeso'). Una tradizione in cui un

[28] Questa definizione può essere applicata all'intero *ḥadīṯ* ma anche solo all'*isnād* o al *matn*. è altra cosa dall'attributo *ġarīb al-ḥadīṯ*, che concerne invece la presenza nel *matn* di *hapax* o termini rari.

Successore cita direttamente il profeta, saltando quindi il livello dei Compa-Compagni, è detta *mursal*, mentre una che mostra debolezze nell'*isnād* o nel *matn* viene detta *ma'lūl* o *mu'allal*, e nel merito al-Ḥākim vi comprende *ḥadīṯ* misti o comprendenti informazioni errate sui trasmettitori.

2.3.2.4 Caratteristiche speciali di *isnād* e *matn*

Ziyādāt al-ṯiqāt sono le aggiunte di autorità che non sono presenti in altre trasmissioni. Un *isnād* è definito *mu 'an 'an* se utilizza genericamente la preposizione *'an* (nel senso di "sull'autorità di') per collegare un trasmettitore all'altro, senza indicazioni specifiche sulla tipologia di trasmissione. Se un *isnād* contiene un difetto non esplicito, la tradizione è detta *mudallas*. Questo è il caso ad esempio in cui i trasmettitori millantano di aver ricevuto la tradizione da un contemporaneo (*tadlīs al-isnād*), identificano un'autorità con un nome o un appellativo insolito (*tadlīs al-šuyūḫ*), oppure omettono il nome di un traasmettitore debole compreso tra due autorità valide (*tadlīs al-taswuiya*). *Mubham* ('oscuro') indica quando un trasmettitore è indicato vagamente e il nome è quindi omesso, *maqlūb* ('trasposto') quando avvengono trasposizioni di *isnād* o di attribuzione di autorità tra una tradizione e l'altra. Anche le differenze nell'ordinamento della parole tra due versioni di una stessa tradizione sono da alcuni segnalate come *munqalib*. La variazione anche solo di una lettera di parola identifica la tradizione come *muḥarraf* ('alterata in una lettera'). Errori veniali, solitamente di vocalizzazione, rendono il *ḥadīṯ muṣaḥḥaf* ('travisato, errato'). Il termine *mudraǧ* ('inserito') indica generalmente se vi sono glosse o inserimenti nell'*isnād* o nel *matn* che fanno apparire una tradizione parte di un'altra. Se due o più trasmettitori sono in disaccordo tra loro nel proporre la stessa versione di una tradizione, questa viene detta *muḍṭarib* ('incongrua'). Applicato a un trasmettitore (*muḍṭarib al-ḥadīṯ*) lo identifica come non affidabile in quanto confuso nel riportare le tradizioni. Un *isnād* con pochi ma affidabili trasmettitori è detto *'ālī*, ('alto') ed è tenuto in elevata considerazione per il fatto che poche trasmissioni limitano naturalmente la possibilità di errori. All'opposto, un *isnād nāzil* ('basso') presenta lunghe catene di trasmettitori. Quando due trasmettitori

contemporanei si comunicano a vicenda tradizioni diverse, è usato il termitermine *mudabbağ* ('screziato, abbellito'). L'*iʿtibār* è il procedimento con il quale si verifica se un trasmettitore è il solo a riportare una tradizione, oppure se la tradizione ha solo un'autorità e non ne esistono delle altre.

2.3.2.5 Riferimento alla validità delle tradizioni

Una tradizione debole ma confermata da un'altra debole è detta *ma 'rūf* ('riconosciuta,'). Il termine si applica anche ai tradizionisti che trasmettono la tradizione ad almeno due trasmettitori successivi. Se di un tradizionista si ignora la biografia o il livello di affidabilità, egli è detto *mağhūl* ('sconosciuto'). Una tradizione valida (*ṣaḥīḥ* o *ḥasan*) e che quindi soddisfa tutti i criteri di verifica è detta *maqbūl* ('accettata'), mentre una tradizione che, comparata a un'altra più debole (*šaḏḏ*), è più importante, è detta maḥfūẓ ('mandata memoria'). Una tradizione ascrivibile a un unico trasmettitore è detta *munkar* ('ignorata'), ma se un trasmettitore è *yarwi l-manākīr* ('che trasmette tradizioni ignorate') questo non comporta automaticamente la cassazione di tutte le sue tradizioni. All'opposto della tradizione pienamente accettata (*maqbūl*), quella rifiutata *in toto* è detta *mardūd*, che però più specificamente identifica quei *ḥadīṯ* dotati di un trasmettitore debole che contraddice altre autorità riconosciute. Se un trasmettitore e sospettato di falsità parziali o scarsa accuratezza nella trasmissione, le sue tradizioni vengono abbandonate (*matrūk*), mentre le tradizioni palesemente fabbricate guadagnano l'attributo *mawḍu*' ('rigettato').

2.4 Studio dei *ḥadīṯ* e critica occidentale

Sebbene le scienze della tradizione islamica avessero operato in un modo dettagliato e metodologicamente fondato tale da non mettere in dubbio l'onestà scientifica di voler distinguere le tradizioni autentiche vere da quelle false, i primi studiosi occidentali esterni alla comunità islamica che si approcciarono allo studio della tradizione, esordirono con un atteggiamento

di relativo scetticismo circa l'effettiva autenticità dei *ḥadīṯ*, pure quelli considerati dalla comunità musulmana come indubitabilmente validi.

Indipendentemente però dal valore delle conclusioni raggiunte, il nascente interesse degli studiosi europei per i *ḥadīṯ* inaugura una sorta di dualismo interpretativo nel dibattito generale sulla Tradizione Islamica, che vede contrapporsi, ma in tempi recenti anche integrarsi, due approcci critici diversi per natura: uno prevalentemente interno alla comunità islamica, dove per il ricercatore la dimensione critica è sovente accompagnata da quella di fede, di credenze religiose non tanto personali quanto comunitarie, e uno esterno a essa (sovente definito come 'occidentale'), dove l'oggetto della ricerca è, o almeno dovrebbe essere per intenzione, separato e lontano dalle credenze personali e religiose.

2.4.1 Analisi e scetticismo

Il primo studioso occidentale che affrontò sistematicamente e con un metodo rigoroso il corpus della Tradizione fu Ignác Goldziher[29], un accademico ungherese della fine del XIX secolo. Nei suoi scritti egli, attraverso un'analisi dettagliata delle raccolte canoniche, confuta la sostanziale connessione tra i *ḥadīṯ* e la vita reale del profeta, avanzando l'ipotesi della fabbricazione postuma *ex novo* della maggior parte della Tradizione stessa (Goldziher, 1889–1890).

L'analisi particolareggiata delle discrepanze, delle confusioni e delle contraddizioni interne al corpus delle tradizioni lo spinse a concludere che i *ḥadīṯ* si svilupparono principalmente come strumento di lotta politica e religiosa tra fazioni diverse. Sotto tale punto di vista, anche lo strumento dell'*isnād* gli parve servire principalmente come leva apologetica ed espediente per la legittimazione di visioni e norme sicuramente posteriori e avulse dal contesto storico della vita di Muḥammad. Goldziher inoltre sottolineava la sostanziale mancanza di interesse per lo sviluppo della tradizione durante il califfato omayyade, posponendo l'intera creazione

[29] Altri studiosi europei come Weil (1848).e Sprenger (1869) avevano in precedenza messo in dubbio l'autenticità di gran parte del canone della Tradizione, ma senza l'organicità e la sistematicità dell'apparato critico di Goldziher.

della Tradizione alle epoche successive come operazione puramente intel-intellettuale senza un contesto culturale e popolare di riferimento. I *ḥadīṯ* sarebbero quindi tarde, deliberate invenzioni allo scopo di convalidare le tesi giuridiche delle varie scuole o di giustificare gli esistenti costumi locali. Perso quindi il valore documenti originali delle primissime fasi dell'Islam, i *ḥadīṯ* conservano però agli occhi di Goldziher il valore di testimonianza delle tendenze più tarde in materia di elaborazione della giurisprudenza.

Tra le varie contraddizioni identificate nel contenuto delle tradizioni, Goldziher evidenzia la presenza di formule ed espressioni bibliche nei discorsi di Muḥammad, la citazione di città e luoghi molto distanti dall'originario territorio arabico oppure non ancora esistenti alla nascita dell'Islam, il riferimento implicito ma abbastanza evidente a nomi e fatti del Califfato Omayyade e Abbaside, oppure il fatto che i Compagni più giovani di Muhammad narrassero più *ḥadīṯ* dei Compagni più anziani, che per la maggiore età avrebbero dovuto ricordare più cose.

Al di là del giudizio negativo sull'origine e la natura delle tradizioni, debitore in una certa misura del contesto culturale orientalista del tempo nella concezione della storia e del pensiero islamico, il suo enorme lavoro di disamina puntuale dei testi e di pedante rilevazione delle incongruenze resta una pietra miliare per le scienze critiche della Tradizione.

Nel solco del tentativo di confermare o rigettare l'intera Tradizione come autenticamente basata su testimonianze coeve al profeta o costruita a posteriori, l'interesse di Joseph Schacht si concentra sulla natura e le origini della Tradizione stessa, arrivando a supporre che lo stretto accostamento tra la *sunna* e la vita del profeta (e quindi il fatto che la biografia profetica fosse essenziale nella definizione del comportamento legale e sociale dei credenti) sia avvenuto relativamente tardi, con al-Šafiʿī (m. 819 d.C.). Schacht si concentrò sulla sfera legislativa e sulle tradizioni normative, privilegiando l'analisi di quei *ḥadīṯ* a carattere legale fondamentali nel dibattito tra i giuristi islamici e tralasciando invece quella parte di Tradizione connessa agli aspetti religiosi, rituali e sociali (Maghen, 2003). Delle tradizioni venne quindi in modo particolare considerato il ruolo e la funzione nella definizione della teoria e della pratica legislativa islamica. In

questo senso anch'egli come Goldziher, dopo aver proceduto con metodi analitici alla datazione degli *isnād* e dei relativi *matn*, sostiene che lo strumento dell'*isnād* fosse un invenzione tarda rispetto alla stratificazione del contenuto stesso della Tradizione (Schacht, 1950).Schacht rilevò inoltre che per molti *ḥadīṯ* l'*isnād*, ma anche alcuni *matn*, subirono evoluzioni successive, contraendo o espandendo il numero di parole da cui erano composti. Una delle debolezze del suo approccio risiede proprio nel fatto di approfondire l'analisi solo sui *ḥadīṯ* che trattano un particolare soggetto (la sfera legale) e di non estendere l'analisi alle tradizioni di natura religiosa o di sminuire il ruolo del Corano nella definizione delle tradizioni stesse (Burton, 1994, p. xxv).

2.4.2 L'autenticità della tradizione scritta

In alternativa se non in contrasto con l'approccio scettico-analitico circa l'autenticità della Tradizione inaugurato da Goldziher e corroborato in una certa misura da Schacht, una corrente di studi rappresentata da figure come Abbott, Sezgin, Azami, ha cercato con metodi e impianti teorici diversi di recuperare la sostanziale validità e autenticità della Tradizione, attraverso analisi e considerazioni che comunque non potevano più prescindere dall'analisi dell'isnād e delle relazioni tra i trasmettitori (Berg, 2000).

Nabia Abbott sostiene la tesi della presenza di pratiche originarie e continuative circa la registrazione scritta dei *ḥadīṯ*, sostenendo che la pratica della trasmissione scritta cominciò tra i Compagni del profeta e proseguì ininterrotta fino ai compilatori del canone affiancando i metodi di trasmissione orale. La persistenza e la trasmissione del documento scritto sarebbe la garanzia della sostanziale autenticità (Abbot, 1972).

Fuat Sezgin condivide l'opinione della Abbott sull'esistenza certa sin dalle origini di materiale scritto per la tradizione, ma la integra con un tentativo scientificamente fondato di invalidare lo scetticismo di Goldziher affermando che oltre a fraintendere alcuni termini chiave delle scienze della tradizione, non abbia avuto le fonti e le risorse adeguate per una ricerca sistematica. Attraverso un'analisi approfondita della terminologia di trasmissione, Sezgin sostiene ad esempio che delle otto modalità di

trasmissione tradizionalmente riconosciute solo due (ascolto e recitazione orale) coinvolgevano la memoria, avendo le altre necessità di registrazione su supporto, pertanto confermando l'ipotesi della prevalenza della trasmissione scritta su quella orale (Sezgin, 1996).

Anche Azami difende la sostanziale autenticità dei *ḥadīṯ*, ma concentrandosi sullo studio dell'isnād per cercare di confutare l'ipotesi di Schacht che tutte le catene di trasmissione risalenti fino al profeta o ai suoi Compagni fossero fabbricate posteriormente (Azami, 1977).

2.4.3 Una posizione intermedia

Un altro gruppo di studiosi, propone invece apparati critici che Berg chiama "the search of middle ground" (Berg, 2000) e sostiene siano in qualche modo a metà strada tra lo scetticismo sistematico di Goldziher e Schacht e la forte rivendicazione sull'autenticità dei *ḥadīṯ* portata avanti da Abbott, Sezgin e Azami. Tra questi studiosi 'moderati' ci sono Juinboll e il suo metodo di approcccio agli *isnād* che vivifica e raffina l'analisi di Schacht (Juynboll, 2001), e Motzki, con le ipotesi sulla non plausibilità della fabbricazione postuma di molti *ḥadīṯ* (Motzki, 2007). Infine un cenno va allo stesso Berg, che propone un metodo alternativo di tipo parzialmente statistico, con il quale analizza un corpus di circa 6000 *ḥadīṯ* contenuti nel *Ǧāmi' al-bayān 'an ta'wīl āy al-qur'ān*, il commentario coranico di al-Ṭabarī (838–923 d.c.) per la verifica delle catene di trasmissione di quelle tradizioni che interessano l'esegesi coranica (Berg, 2000).

Parte II:
ANALISI

3. Metodologia e materiali

In questa seconda parte, che costituisce il nucleo del lavoro dottorale, saranno presentate e analizzate alcune strategie di analisi computazionale applicate al testo arabo di due raccolte canoniche di tradizioni. Il testo dei *ḥadīṯ* è stato processato con un ampio numero di strumenti e programmi informatici, la maggior parte dei quali è stata appositamente concepita e realizzata. La descrizione di questi strumenti e della loro giustificazione teorica e procedurale è preceduta da alcune considerazioni di tipo metodologico sulle tecniche e sulle fonti utilizzate per l'elaborazione.

3.1 Parametri fondamentali

Ai fini di rendere comprensibile e valutabile il lavoro di analisi testuale ai diversi livelli (strutturale, morfo-sintattico, semantico) e di interpretare in un unico quadro di riferimento le differenti strategie applicate, si è scelto di introdurre un sistema di parametri attraverso i quali definire e collocare i modelli e le applicazioni sviluppate. Tali parametri sono basati sulla classificazione esplicita di alcune delle tendenze di ricerca della linguistica computazionale trattate in dettaglio nel capitolo 1.

3.1.1 Grado di automazione

Il primo parametro consiste nella valutazione del grado di automazione dello strumento di analisi, vale a dire fino a che punto il trattamento dell'input in questione può avvenire senza l'intervento umano (Vlachos, 2011).

A carattere di elementare esempio, si consideri come input dato un brebreve estratto di testo digitalizzato e come output atteso la creazione della relativa lista di frequenza delle parole[30] che lo compongono (figura 3.1).

Un processo con parametro negativo di automazione e positivo di intervento umano comporta ad esempio che l'operatore umano copi ciascun termine separato da spazi bianchi su di un nuovo paragrafo, controlli a una a una ogni stringa, conti le frequenze di ricorrenza e infine cancelli le eventuali ripetizioni.

Un processo con entrambi i parametri positivi sarà invece caratterizzato da una parziale automazione integrata dall' intervento umano. In questo caso uno *script*[31] potrebbe sostituire automaticamente gli spazi bianchi con delle interruzioni di paragrafo[32] e un altro *script* ordinare alfabeticamente il testo. A questo punto l'operatore umano semplicemente scorre la lista, conta le ricorrenze di parola, facilmente riconoscibili in quanto successive l'una all'altra ed elimina infine le duplicazioni. In un processo interamente automatico invece, anche quest'ultima serie di operazioni è compiuta dalla macchina (cfr. figura 3.1).

Uno strumento a totale automazione esclude quindi l'intervento umano 'durante' l'elaborazione dei dati, riservandolo alla sola creazione dello strumento stesso oppure alla validazione dei dati ottenuti. Oltre al risparmio di tempo, l'automazione garantisce rispetto all'intervento umano due fattori chiave dell'analisi computazionale: la reiterazione (ripetere il processo tutte le volte necessarie) e la trasferibilità (applicare il processo anche ad altri dati di natura omologa).

[30] Per questo esempio si intende 'parola' in senso limitatamente computazionale e non linguistico, come sequenza di caratteri delimitata da spazi.
[31] Lo *script* è una breve serie di istruzioni per una specifica elaborazione dei dati. Un programma informatico può essere costituito da numerosi *script.*
[32] La funzione 'trova e sostituisci' dei comuni word processors è un esempio di *script* già contenuto in un programma.

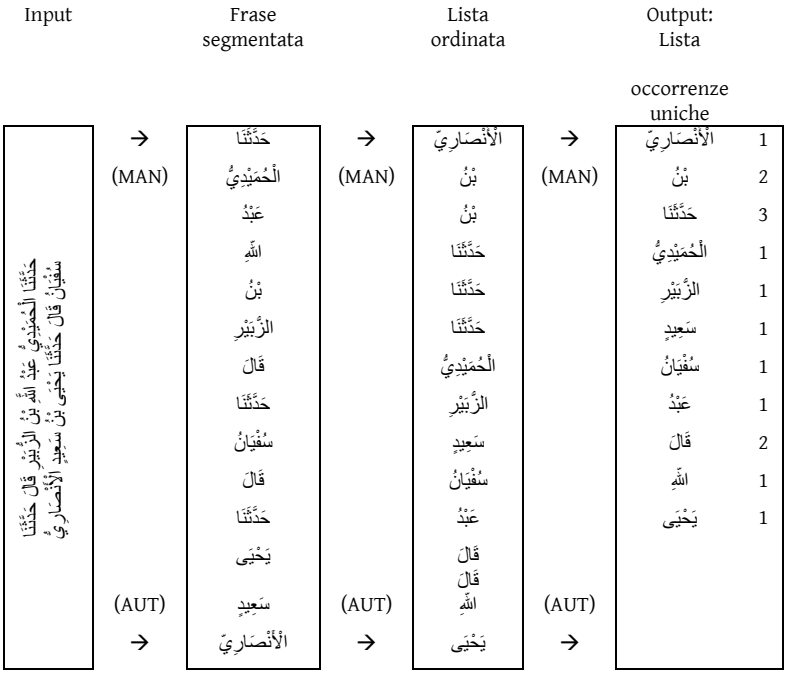


Figura 3.1 – fasi di elaborazione dell'operazione di creazione di una lista di frequenza. Ogni fase può essere compiuta manualmente o in modo automatico

3.1.2 Superficie e profondità del testo

Alcune recenti tendenze nell'analisi testuale computazionale, in particolare lo *Structure Discovery Framework* (Biemann, 2012), propongono un quadro di riferimento teorico che identifica la superficie del testo come oggetto privilegiato di analisi. In questo senso, il testo è interpretato come un insieme di stringhe e caratteri a profondità zero e non dotati di altro significato se non la propria relazione posizionale con gli altri elementi. Proprio questa assenza di 'significato' permette di indagare la superficie senza bisogno di conoscenza linguistica e senza i condizionamenti che essa impone in modo esplicito o implicito. Il fine principale di questo tipo di analisi è identificare, in modo essenzialmente statistico, regolarità,

ricorrenze, eccentricità che consentano di delineare le strutture stesse intorno alle quali il testo e costruito ed eventualmente trovarne di nuove (Perkins, 2010).

Scendendo invece di un grado nell'ideale parametro che valuta la profondità con cui un testo viene considerato, si trovano gli approcci che trattano di una componente linguistica in particolare. Qui la morfologia, la sintassi e la semantica di un testo sono analizzate attraverso strategie variamente basate su regole o su componenti statistiche, ma facendo ricorso a risorse precostituite di tipo linguistico, come basi lessicali, inventari morfologici, liste di relazioni sintattiche (Bolshakov & Gelbukh, 2004).

Il livello ancora più profondo di analisi interessa le implementazioni computazionali dei modelli teorico grammaticali, che tentando di interpretare una lingua nell'interezza dei suoi fenomeni, analizzano un testo fornendone una rappresentazione complessa in cui tutte o gran parte delle componenti linguistiche sono rappresentate (Isac & Reiss, 2008, p. 79-100).

3.1.4 Empirismo e conoscenza

Se il parametro precedente misura il livello di profondità al quale un testo viene analizzato, il dualismo tra empirismo e conoscenza riguarda piuttosto le strategie e gli strumenti usati per affrontare quel testo (Curran, Clark, & Bos, 2007). L'alternanza tra metodi basati su algoritmi empirico-statistici e metodi che prevedono insiemi più o meno completi di regole e condizioni è già stata analizzata nel capitolo 1 in merito alla morfologia computazionale e alla traduzione automatica. Nell'ambito del trattamento computazionale dei *ḥadīṯ* questo parametro assume però un secondo significato, quello di misurare il grado di 'ignoranza' o 'conoscenza pregressa' che uno strumento di analisi dimostra nell'affrontare il testo delle tradizioni. Ad esempio per certi tipi di analisi, come la segmentazione e l'interpretazione della struttura, l'ipotesi che si cercherà di dimostrare è se in tali casi un approccio 'cieco', che non usi cioè informazioni sui *ḥadīṯ* che provengano da un contesto di conoscenza extra-testuale, possa essere più efficace. Il livello di

conoscenza interverrebbe solo in seguito per la verifica dei risultati ottenu-ottenuti.

3.1.5 Dipendenza dal contesto e trasferibilità

Il parametro concerne la progettazione di strumenti che siano impiegabili unicamente sul testo al centro dell'analisi o possano essere estesi ad altri testi analoghi o addirittura trasferiti ad ambiti testuali appartenenti a tipologie diverse (Grefenstette, Semmar, & Elkateb-Gara, 2005). L'efficienza di un sistema computazionale può essere in genere aumentata attraverso specifiche condizioni restrittive ricavate dal contesto, ma questa strategia ha lo svantaggio di localizzare eccessivamente l'approccio e di renderlo quasi esclusivo per quella specifica analisi testuale.

3.1.6 Analisi, rappresentazione e generazione

Uno strumento di elaborazione computazionale al testo può essere valutato secondo il tipo di risultati che può produrre. A un primo livello, l'elaborazione consiste essenzialmente nell'analisi, che può comprendere la segmentazione superficiale, l'estrazione d'informazione, l'annotazione morfologica, sintattica e semantica dei componenti testuali.

A un secondo livello, i dati estratti dall'input possono essere interpretati secondo paradigmi teorici che ne permettono un qualche tipo di rappresentazione. Ne sono esempi, la visualizzazione di alberi sintattici, di reti di relazione semantica, oppure le rappresentazioni a grafi che permettono di visualizzare graficamente le relazioni interne di una serie ordinata di dati (Kaufmann & Wagner, 2001).

L'ultimo livello prevede la capacità del sistema, previa analisi e rappresentazione delle strutture linguistiche e informative di un testo, di generare nuova informazione o nuovo testo. Ciò richiede l'impiego di modelli di grammatica formale in grado di elaborare e rappresentare a livello profondo le componenti morfo-sintattiche e semantiche, al fine di produrre un testo coerente linguisticamente e dal punto di vista dei contenuti espressi.

3.2 Gli strumenti per la programmazione e la gestione digitale delle informazioni

3.2.1 La programmazione in Python

I programmi e gli applicativi originali realizzati a corredo di questo lavoro sono stati elaborati utilizzando l'ambiente di programmazione Python. Python è un linguaggio dotato di alcune caratteristiche che lo rendono un buon candidato per chi ha vuole realizzare strumenti di computazione testuale senza però possedere competenze avanzate e pregresse nel campo delle scienze informatiche e di programmazione (Mertz, 2003).

Il linguaggio è stato creato da Guido Van Rossum all'inizio degli anni '90, e fu chiamato così in omaggio al sodalizio cinematografico inglese Monty Python (van Rossum, 1993). È un linguaggio interpretato ad alto livello, non agisce cioè al livello più basso possibile di calcolo ma necessita di essere interpretato da un altro programma che lo traduce in linguaggio macchina o viceversa. Il paradigma principale di Python è il fatto di essere orientato agli oggetti (*object oriented*). Gli oggetti sono strumenti software capaci di interagire tra loro in modo modulare perché gestiti da classi precostituite che ne definiscono attributi[33] e metodi[34]. La presenza di classi, attributi e metodi facilitano le operazioni di interdipendenza[35] e ricorsività[36] (Lutz, 2007).

Python è dotato di una sintassi di programmazione simile alla logica del discorso umano che la rende particolarmente chiara e intellegibile da parte dell'utente e del programmatore (Beazley, 2000). La comprensibilità e la compattezza delle regole sintattiche di Python (ad esempio l'uso della tabulazione per annidare le istruzioni) ne consentono sia una facile lettura, sia un rapido apprendimento anche a livelli avanzati, rispetto alla complessità di struttura di altri celebri linguaggi di programmazione, come

[33] I dati stessi da elaborare.
[34] Le procedure da applicare a una serie di dati.
[35] L'interazione fra diverse parti del programma.
[36] La proprietà di un'operazione di poter essere riutilizzata ciclicamente.

C. In questo senso, la presenza di classi predefinite e l'essenzialità della scrittura del codice permettono di ridurre notevolmente (nell'ordine di uno a dieci) il numero di righe necessarie per ogni istruzione. A titolo di esempio viene mostrato il codice necessario in Python e in C per tradurre la semplice istruzione "Scrivi sullo schermo la parola 'tradizione'":

Python:

```
print "tradizione"
```

C:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("tradizione\n");
}
```

Python permette inoltre al programmatore di tradurre in modo relativamente trasparente gli algoritmi e i processi logici alla base dell'elaborazione computazionale, in quanto il codice procedurale presenta un alto grado di naturalezza espressiva (Ziadé, 2008). Ad esempio ecco le righe di codice necessarie per impartire l'istruzione "Scrivi tutti i numeri naturali compresi tra 1 e 10":

```
a = 0
while a < 10:
     a = a + 1
     print a
```

Altre caratteristiche peculiari di Python sono la sua struttura modulare, che permette la strutturazione dell'ambiente di programmazione in moduli autonomi e riutilizzabili da altri ambienti, e il suo essere sostanzialmente indipendente dal tipo di piattaforma o sistema operativo su cui è installato. Questo favorisce anche l'integrazione bidirezionale con altri linguaggi di programmazione, ad esempio è possibile in un ambiente Java utilizzare codice Python e viceversa. Inoltre Python può utilizzare il sistema Unicode per codifica dei caratteri, che permette alla grafia araba e a quella basata su caratteri latini di essere processata e visualizzata in seguito[37]. La licenza

[37] Per approfondimenti su Unicode e sistemi di codifica cfr. titolo 3.4.1.1.

Open source[38] con la quale il pacchetto di sviluppo è distribuito, ne rende possibile il riutilizzo, l'espansione e la libera distribuzione a terzi (Hetland, 2005).

3.2.2 Annotare e organizzare: Il linguaggio XML

Python è stato scelto nel presente lavoro come linguaggio per la costruzione e l'implementazione dei programmi di elaborazione dei dati, ma la grande mole e la diversificazione dei risultati prodotti in output ha richiesto un linguaggio più specifico per la strutturazione e l'organizzazione dei dati. Per questo scopo è stato utilizzato il linguaggio XML, acronimo per eXtensible Markup Language. L'XML non è un linguaggio concepito per la programmazione e la compilazione di istruzioni, bensì è un linguaggio di tipo descrittivo, basato su regole sintattiche aperte per la definizione e l'organizzazione di qualsiasi tipo di contenuto in forma digitale (Skonnard & Gudgin, 2001). Il concetto alla base dei linguaggi di descrizione è quello dell'etichettatura o marcatura (ingl. *tagging*). Il codice sarà quindi composto da due tipologie di informazioni, il contenuto e la marcatura esplicita dello stesso. Una volta che ogni unità di contenuto è designata univocamente e definita nei suoi confini, le relative operazioni di estrazione, elaborazione, filtraggio o riorganizzazione dei dati risultano elementari e immediate.

```
<tag>contenuto</tag>
```

L'esempio di cui sopra mostra una marcatura tipica dell'XML, dove il contenuto e racchiuso a sinistra da un *tag* di apertura e a destra da uno di chiusura (che è identico a quello di apertura se non per il simbolo di chiusura '/').

Diversamente da altri linguaggi descrittivi come l'HTML[39] che contengono insiemi predefiniti e chiusi di *tag* ciascuno con la propria

[38] Con *opensource* si intende un tipo di licenza *software* che prevede generalmente l'utilizzo gratuito del prodotto ma soprattutto l'accesso ai file sorgente che contengono il programma stesso, al fine di permettere alla comunità scientifica lo studio e il miglioramento del prodotto stesso (OSI, 1998).

funzione che verrà interpretata dai programmi di navigazione (*browser*), in ambiente XML è virtualmente possibile creare un numero illimitato di *tag* a seconda delle esigenze del contesto. Per limitare però la frammentazione delle informazioni marcate e rendere in qualche modo uniformi i set di marcatura utilizzati nella programmazione, l'XML può essere arricchito da fogli di stile, protocolli e standard condivisi per favorire l'organizzazione e l'interscambio dei dati[40]. Proprio la possibilità di disporre di sistema di marcatura universale ed esplicito rende l'XML un formato ideale per la condivisione e la trasmissione di dati da e tra sistemi diversi.

Accanto alla descrizione precisa e formale dei dati, l'XML consente anche una complessa e rigorosa organizzazione degli stessi attraverso il principio della nidificazione. Ogni segmento di informazione delimitato destra e a sinistra dallo stesso *tag* può cioè essere contenuto a sua volta in segmenti più vasti oppure contenere al suo interno ulteriori sotto marcature in una struttura a padri e figli secondo l'esempio seguente:

```
<tag1>
      <tag2>contenuto1</tag2>
      <tag2>
            <tag3>contenuto2</tag3>
            <tag3>contenuto3</tag3>
      </tag2>
</tag1>
```

La nidificazione permette quindi la presenza di più dimensioni di ordinamento in una struttura però rigorosamente lineare, e opposta ad esempio alle organizzazioni tabulari dei dati tipiche di altri sistemi di basi di dati.

Infine l'XML è pienamente compatibile con il linguaggio di programmazione Python e integrabile in esso (Jones & Drake, 2002).

---

[39] Acronimo per *HyperText Marking Language*, il linguaggio più diffuso e utilizzato per l'organizzazione statica di contenuti ipertestuali e multimediali in rete.

[40] Con XSL (*Extensible Stylesheet Language*) ci si riferisce a quella famiglia di linguaggi descrittivi che permettono di modificare o visualizzare l'XML. Ad esempio lo XSLT (*Extensible Stylesheet Language Transformations*) è un linguaggio per la descrizione delle trasformazioni necessarie a trasformare un documento XML in un altro documento XML (Jones & Drake, 2002).

3.3 Le fonti per l'input

3.3.1 Criteri di selezione

Il rapporto tra input e processamento dei dati può essere di tipo diverso a seconda del contesto in cui nasce l'idea dell'elaborazione computazionale. Da un lato si può avere la preesistenza di un dato testo come input, sul quale modellare e applicare gli strumenti di elaborazione. Altro è il caso in cui la priorità viene invece garantita alla creazione degli strumenti computazionali, che verranno alimentati con dati in input scelti solo successivamente, in modo da adattarsi ai prerequisiti richiesti dal sistema.

Nel caso del presente lavoro, la finalità esplicita è di dare uguale importanza sia all'input sia alla fase di elaborazione, vale a dire valorizzare il più possibile sia la scelta dei *ḥadīṯ* come testo da analizzare sia la costruzione di strumenti originali di analisi. Quindi anche se il prerequisito alla base dell'ipotesi di ricerca è esplicitamente quello di trattare come input privilegiato il testo dei *ḥadīṯ*, la scelta di quale raccolta e quale edizione selezionare all'interno dell'intero corpus della tradizione ha comunque seguito alcuni criteri specifici.

3.3.1.1 Digitalizzazione e disponibilità

I criteri principali nella scelta delle fonti sono stati quello della digitalizzazione, della libera disponibilità in linea e dell'importanza.

La digitalizzazione comporta la scelta di un'edizione che comprenda almeno un'intera raccolta di *ḥadīṯ* completamente riversati in forma digitale e corredati dei segni diacritici relativi a vocalizzazione e rafforzamento consonantico.

Il criterio di usare una versione reperibile in linea e accessibile pubblicamente garantisce sia l'assenza di questioni relative alla pubblicazione di dati protetti dal diritto d'autore sia la possibilità di preservare il pubblico accesso e la consultazione dei dati anche in sede di elaborazione e diffusione dell'output.

Infine, considerato il carattere sperimentale del presente lavoro e il suo supposto valore come un esempio di applicazione del calcolo

computazionale a testi della tradizione classica araba, si è scelto di adottare un testo importante e celebre nella frequentazione popolare come in quella della letteratura scientifica di settore.

3.3.1.2 Attendibilità

L'attendibilità di una versione di una fonte scritta concerne con la corrispondenza integrale e particolare all'originale da cui la versione è tratta e con la possibilità di tracciare il percorso del testo e dei suoi curatori dall'origine fino alla versione in questione. Nel caso di un testo digitalizzato, l'attendibilità è anche valutabile in funzione del metodo usato per la digitalizzazione, vale a dire interamente manuale, interamente automatico o automatico con verifica manuale (Boone, 2003).

Per i fini del presente lavoro si è consapevolmente scelto di non dare eccessiva priorità al criterio di attendibilità nella selezione dell'input. Una delle finalità dichiarate in ipotesi è infatti quella di mostrare l'efficacia di strumenti di analisi computazionale applicati a una collezione di *ḥadīṯ* piuttosto che offrire una versione critica o filologicamente ponderata della collezione stessa. Il testo in input è quindi utilizzato primariamente come materiale di sperimentazione e in quanto tale la sua attendibilità filologica è relativamente rilevante.

3.3.2 Una versione del *Saḥīḥ* di al-Buḫārī

I criteri scelti per la selezione dell'input hanno quindi privilegiato la disponibilità e la completa digitalizzazione rispetto all'attendibilità e alla correttezza nella lezione testuale. In base a questi criteri è stata quindi scelta la versione elettronica del *Al-ǧāmiʿ al-ṣaḥīḥ* di al-Buḫārī secondo l'edizione di Aḥmad Šākir, recentemente ristampata nell'originale formato litografico dalla casa editrice Dār Ṭawq an-Naǧāh (Al-Buḫārī, Al-Jāmiʿ al-Ṣaḥīḥ , 2001).

In questo caso, se l'edizione a stampa di riferimento è affidabile e tracciabile, così non è per il processo di trasformazione digitale del testo, per il quale non è stato possibile reperire informazioni dettagliate. Questa versione elettronica è comunque una delle più celebri e diffuse on-line, ed è

reperibile in vari formati su molti archivi *open source* in linea, tra cui quello ospitato da Internet Archive all'indirizzo http://archive.org/details/yonenih [data di accesso 28 febbraio 2011].

Prima di subire qualsiasi pre-trattamento testuale, il testo al momento della sua acquisizione per scaricamento dall'archivio si componeva di nove file Microsoft Word ciascuno contenente uno dei nove volumi di *ḥadīṯ* nei quali la raccolta è organizzata, digitalizzati in caratteri arabi e pienamente vocalizzati.

3.4 Traslitterazione e resa della grafia araba

Le questioni generali relative alla trascrizione di una lingua dal proprio sistema grafico a un altro o viceversa verranno solo accennate in questa sede Per quanto riguarda la invece la traslitterazione[41] della lingua araba il problema è piuttosto complesso e stratificato e dipende da vari fattori, tra cui:

- un sistema di scrittura difettivo che come comportamento generale non riporta graficamente alcuni fonemi dell'enunciato, quali le vocali brevi e i rafforzamenti consonantici;
- la mancanza di un'autorità accademica o linguistica unanimemente riconosciuta che possa stabilire norme e criteri univoci per la trascrizione e la traslitterazione dall'arabo verso altre lingue e viceversa. Questo ha permesso la compresenza all'interno e all'esterno del mondo arabofono (o meglio, se esistesse il termine, 'arabografo') di un gran numero di consuetudini e sistemi di traslitterazione, che variano in dipendenza di fattori geografici, scientifici e culturali.

[41] Mentre la trascrizione può essere definita come 'l'operazione di rappresentare gli elementi del linguaggio, sia suoni sia segni, in un altro sistema di suoni o segni' la traslitterazione è un tipo particolare di trascrizione che tende a privilegiare il dato grafico del testo originale rispetto a quello fonetico/fonologico (Wellisch, 1975).

Pur all'interno di questo quadro di riferimento (Reichmuth, 2009), l'attenzione è qui rivolta ai problemi di codifica e traslitterazione derivanti dal trattamento computazionale di testi in caratteri arabi.

3.4.1 Testo arabo, traslitterazione e computazione

3.4.1.1 La codifica dei caratteri in font

Se il carattere può essere definito come un grafema, vale a dire la rappre-rappresentazione astratta del segno, il font, in ambito informatico e tipografico, può essere considerato come un grafo, cioè una specifica realizzazione contestuale di un dato grafema (Coulmas, 1994). In questo senso, ogni set di font che viene costruito con particolari caratteristiche grafiche e di aspetto, si basa su di una tabella di associazione che fa corrispondere il singolo font a un determinato codice che interpreta il carattere-grafema di riferimento.

Sin dalle origini delle scienze della computazione sono state quindi create delle tabelle in cui ciascun grafema è associato a un codice univoco interpretabile dal computer. Il primo sistema a essere creato nel 1961, e ancora oggi diffuso, è l'ASCII (*American Standard Code for Information Interchange*), un sistema di codifica che prevede l'impiego di una limitatissima quantità di memoria computazionale (7 bit) per la codifica stessa e pertanto prevede la possibilità di definire solo 128 segni, che corrispondono circa all'alfabeto dell'inglese minuscolo e maiuscolo, ai numeri e ad alcuni caratteri non alfanumerici. Dai tempi dell'ASCII a oggi sono poi nati sistemi a memoria più estesa (ad esempio l'ISO-8859) che riescono quindi a contenere molte più associazioni di caratteri, dai vari diacritici dell'alfabeto latino, fino a sistemi di scrittura completamente diversi o sistemi di simboli. Tra questi sistemi quello chiamato Unicode, con i suoi 21 bit di memoria, potendo virtualmente contenere milioni di codifiche di carattere riesce attualmente a codificare in modo univoco e omogeneo tutti gli alfabeti e sistemi grafici esistenti, relativi a lingue naturali o artificiali[42].

[42] Riferimenti reperibili sul sito dell'Unicode Consortium:

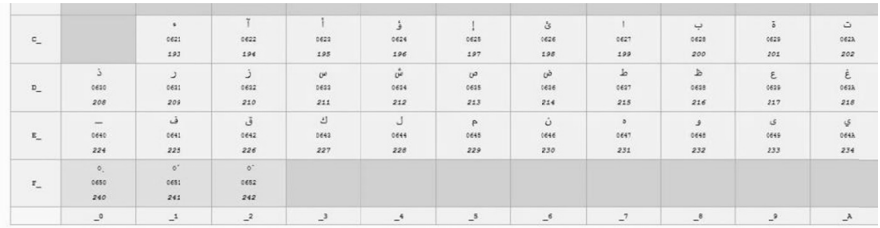


Figura 3.2 La codifica di alcuni caratteri arabi nella tabella del sistema ISO8859-9[43]

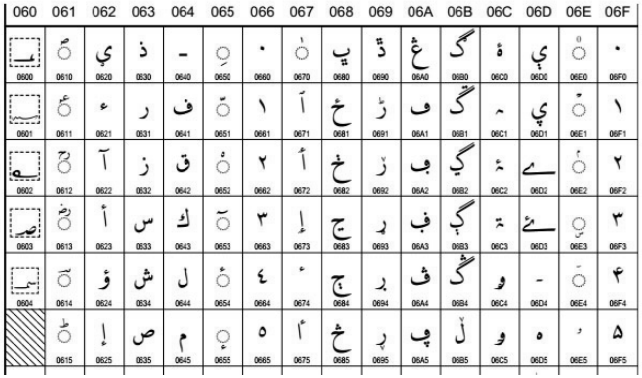


Figura 3.3 La codifica di alcuni caratteri arabi nella tabella del sistema Unicode UTF-8[44]

A oggi però l'esistenza di un protocollo con il quale potenzialmente unificare la codifica di qualsiasi segno digitalizzato non garantisce da sola la diffusione universale dello stesso in tutti i sistemi operativi e i programmi applicativi, per cui sistemi diversi di codifica convivono e rendono talvolta problematica l'interconnessione e lo scambio di informazioni.

http://www.unicode.org/consortium/consort.html [accesso 22 febbraio 2013]

[43] Fonte: http://en.wikipedia.org/wiki/ISO/IEC_8859-6

[44] Fonte: http://www.unicode.org/charts/PDF/U0600.pdf

3.4.1.2 La visualizzazione dei caratteri arabi

Una lingua a caratteri non-latini come l'arabo è stata particolarmente interessata dall'adozione nel tempo di sistemi di codifica diversi e concorrenti tra loro, sia dal punto di vista della mappatura dei caratteri arabi sia da quello dei caratteri latini variamente aggiunti di segni diacritici per la traslitterazione. Un testo digitalizzato in arabo basato sul sistema di codifica Windows-1256 (molto diffuso perché di default sull'omonimo sistema operativo) non sarà leggibile o visualizzabile da programmi o ambienti che utilizzano un altro sistema, ad esempio l'ISO8859, a meno di non usare un programma specifico di conversione.

Un altro problema di compatibilità riguarda invece i set di font per l'arabo o quelli contenenti i caratteri latini per la traslitterazione. In questo caso solo quelli costruiti secondo un sistema di codifica standard (che sia ISO o Unicode o altri) possono essere convertiti correttamente in altri set di font: in caso contrario la conversione è praticamente impossibile a meno di non costruire un sistema *ad hoc* di conversione[45].

3.4.1.3 Caratteri arabi e programmazione

Grazie alla diffusione e alla standardizzazione dei sistemi di codifica la visualizzazione a schermo e a stampa dell'arabo non costituisce più un problema rilevante. Qualche difficoltà sorge invece quando il testo arabo viene sottoposto a elaborazione da parte di programmi computazionali.

Un primo problema consiste nella non completa compatibilità dei diversi sistemi di codifica con i linguaggi di programmazione. Alcuni di questi non consentono l'utilizzo dello standard Unicode o lo gestiscono con procedure complesse. Inoltre nel listato di un programma si possono distinguere due ordini di stringhe, quelle contenenti le istruzioni di elaborazione e quelle

[45] È il caso ad esempio del font specifico per la resa dell'arabo traslitterato *Timlj*, ideato pionieristicamente da Claudio Lo Jacono e ancora oggi utilizzato soprattutto negli ambienti accademici di studi arabi e afroasiatici. Poiché l'associazione tra codice e carattere non rispecchia un sistema di codifica riconosciuto, la conversione in un altro font di un testo scritto in *Timlj* non è automatica (come invece capita quando si cambia ad esempio la formattazione di un testo dal font *Arial* al font *Times New Roman*) e richiede un certo grado di intervento manuale per evitare errori nella visualizzazione.

contenenti i dati testuali. Le stringhe di istruzione sono in genere basate sulla lingua inglese e utilizzano la codifica ANSII, mentre per le stringhe di contenuto testuale è possibile prevedere la codifica in sistemi più complessi come Unicode. Questo significa che ad esempio l'inserimento di caratteri non ANSI all'interno delle stringhe di istruzione potrebbe non essere sem-sempre possibile o consigliato per l'efficienza del sistema, ad esempio nel caso dell'uso della sintassi delle espressioni regolari.

L'ordinamento destra-sinistra della scrittura araba causa un altro inconveniente rilevante durante la stesura di un programma. In Python ad esempio, anche se è possibile inserire del testo in caratteri arabi all'interno delle stringhe di contenuto, non però poi possibile operarvi direttamente con il cursore, in quanto il testo, seppur visualizzato correttamente da destra a sinistra, è ordinato logicamente da sinistra a destra e non permette quindi la corrispondenza fisica tra ciò che si vede e ciò che il linguaggio interpreta. L'uso dell'arabo in sede di programmazione è quindi fattibile solo a condizione che il testo stesso non debba essere modificato o manipolato manualmente (Madhany, 2006).

3.5.2 Il sistema di traslitterazione Buckwalter

Il sistema di traslitterazione per l'arabo fu concepito da Tim Buckwalter agli inizi degli anni '90 con lo scopo dichiarato di avere una trascrizione rigorosa dei segni di scrittura dell'arabo che potesse essere facilmente interpretata dal computer indipendentemente dal sistema di codifica scelto. (Habash, Soudi, & Buckwalter, 2007).

Il principio alla base del sistema è quello di utilizzare per la conversione unicamente i caratteri disponibili nel codice più semplice, l'ASCII. I criteri guida nella scelta delle associazioni sono quelli di privilegiare l'unicità del segno e per quanto possibile, la comprensibilità in lettura. Quindi a unico carattere dell'arabo corrisponde unico carattere della traslitterazione. Laddove possibile sono stati usati i caratteri alfabetici omologhi o simili (ad esempio *r* per ر o *s* per س), mentre per i caratteri solitamente traslitterati con diacritici aggiunti al carattere latino (ad esempio ā oppure) si è scelto di

utilizzare la forma maiuscola dei caratteri latini, poiché priva di valore nella scrittura dell'arabo (ad esempio *S* per ص o H per ح).

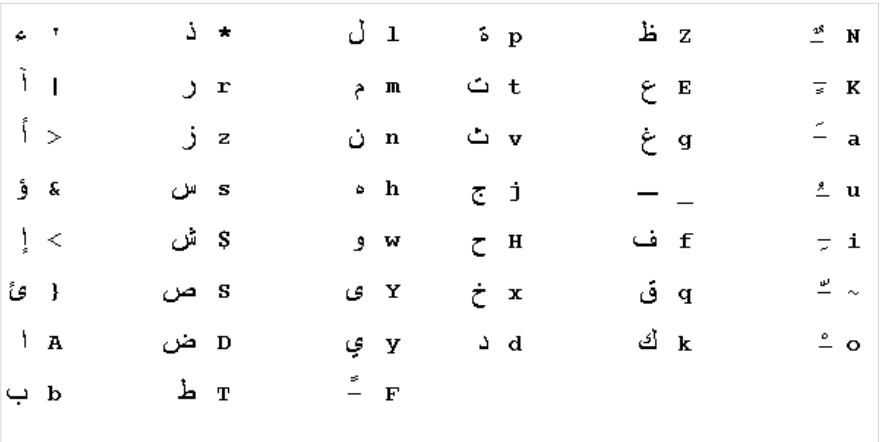


Figura 3.4 Tabella di conversione Buckwalter per la traslitterazione dell'arabo [fonte: http://www.qamus.org/transliteration.htm]

Tale sistema garantisce la completa corrispondenza tra caratteri arabi e traslitterati e permette virtualmente la traslitterazione ricorsiva dello stesso testo in modo illimitato senza perdere informazione. L'associazione biunivoca è rigorosamente mantenuta, in una certa misura a scapito della comprensibilità immediata in lettura, anche tenendo conto che lo scopo principale del sistema è quello di essere adottato soprattutto in sede di elaborazione dati, dove il testo è nascosto al lettore umano.

La capacità di questa traslitterazione di adattarsi, almeno in origine, perfettamente al trattamento computazionale è stata col tempo appannata dalla nascita di linguaggi di programmazione con sintassi che prevedono l'uso funzionale di alcuni dei caratteri utilizzati dal sistema per la resa dell'arabo. Lo stesso Buckwalter ha recentemente notato che per rendere ad esempio il sistema adatto al trattamento con XML, alcune corrispondenze originarie andrebbero cambiate in modo da non interferire con i caratteri

riservati del linguaggio descrittivo, in particolare i segni ‘<’, ‘>’ e ‘&’ (Buckwalter, Buckwalter Arabic transliteration).

3.5.3 Scelte di traslitterazione

La strategia adottata in questo lavoro per la scrittura di programmi che trattano testo arabo consiste nel convertire preventivamente il testo arabo usando un sistema rigorosamente biunivoco di traslitterazione che conservi tutta l’informazione grafematica dell’originale. La fase di elaborazione avviene sul testo traslitterato e solo in fase di produzione dell’output il testo finale è nuovamente traslitterato in arabo.

È stato quindi scelto il sistema Buckwalter di traslitterazione in quanto il più adattabile alle esigenze specifiche di programmazione, ma con una serie di modifiche nell’associazione dei caratteri (Lancioni, 2011), in modo da evitare il più possibile il conflitto con i caratteri riservati di Python, XML e con la sintassi delle espressioni regolari, che costituiscono i tre ambienti di programmazione prevalentemente utilizzati nel trattamento computazionale dei *ḥadīṯ*.

	Carattere arabo	Buckwalter	BW modificato
1	ء	'	-
2	آ	|	O
3	أ	>	E
4	ؤ	&	W
5	إ	<	e
6	ئ	}	J
7	ا	A	
8	ب	b	
9	ة	p	o
10	ت	t	
11	ث	v	C
12	ج	j	
13	ح	H	

	Carattere arabo	Buckwalter	BW modificato
24	ظ	Z	
25	ع	E	c
26	غ	g	G
27	ف	f	
28	ق	q	
29	ك	k	
30	ل	l	
31	م	m	
32	ن	n	
33	ه	h	
34	و	w	
35	ى	Y	
36	ي	y	

14	خ	x	
15	د	d	
16	ذ	*	V
17	ر	r	
18	ز	z	
19	س	s	
20	ش	$	X
21	ص	S	
22	ض	D	
23	ط	T	

37	◌ً	F	N
38	◌ٌ	N	U
39	◌ٍ	K	I
40	◌َ	a	
41	◌ُ	u	
42	◌ِ	i	
43	◌ّ	~	+
44	◌ْ	o	_
45	پ	p	
46	ڤ	v	

Tabella 3.5: Confronto tra il sistema di traslitterazione originale Buckwalter e il sistema modificato (Lancioni, 2011)

Dalla tabella 3.1 si può notare come le modifiche abbiano riguardato tre tipologie:

- eliminazione dal sistema dei caratteri riservati in Python, XML o nelle espressioni regolari (1-6, 16, 20, 43);
- leggibilità (9, 25, 26, 37-39, 44);
- riorganizzazione dell'inventario disponibile (11).

3.5.4 Il modulo di traslitterazione

Il programma che permette il passaggio dalla grafia araba a quella traslitterata e viceversa è stato scritto in Python e concepito come modulo indipendente, invocabile in ogni punto del programma attraverso una funzione specifica. L'algoritmo di funzionamento è piuttosto elementare: il sistema mappa il codice Unicode di ciascun carattere del testo in input e lo sostituisce con il relativo carattere secondo attingendo da una lista interna di tuple ciascuna contenente il carattere traslitterato e la definizione Unicode.

4. Analisi dei *ḥadīṯ*: la superficie del testo[46]

Lo studio moderno e contemporaneo sulla Tradizione islamica, in particolare la cosiddetta 'critica alle fonti' (Günther, 2005) mostra che l'attenzione degli studiosi soprattutto occidentali è concentrata sulla verifica della supposta attendibilità dei *ḥadīṯ* attraverso l'analisi dettagliata degli *isnād* e delle catene di trasmissione ivi contenute, come accennato nel Capitolo 2.

In questa sede acquista invece importanza l'approccio comune di analisi testuale usato dagli studiosi, piuttosto che il contenuto delle loro affermazioni. Le analisi estremamente raffinate sulle catene di trasmissione della tradizione compiute, per citarne alcuni, da Wensinck (1933), Schacht (1950), fino ad arrivare a Juynboll (1983) e Sezgin (1996) rappresentano un'importante innovazione di metodo che ha permesso all'analisi dell'*isnād* di diventare "a sophisticated and most efficient tool of source-criticism of classical Arabic compilations" (Günther, 2005, p. 81).

Accanto all'intrinseco valore scientifico dell'opera di questi studiosi, un tratto che sicuramente li accomuna è di aver compiuto le analisi sul testo con un approccio tipicamente analogico e basato sulla pura elaborazione umana dei dati, salvo alcune eccezioni (Berg, 2000). Si può immaginare la quantità di tempo e di risorse impiegate per leggere manualmente migliaia di tradizioni, identificarne l'*isnād*, analizzarli e costruire sistemi astratti di relazioni di significato. Inoltre è evidente che solo il materiale

[46] Questo capitolo è parzialmente costruito sullo sviluppo e l'ampiamento di alcune considerazioni pubblicate dall'autore in tre articoli durante il percorso di ricerca dottorale (Boella M. , 2011a; Boella M. , 2011b; Boella, Romani, Al-Raies, Solimando, & Lancioni, 2011).

effettivamente e direttamente esaminato può generare analisi e conclusioni, anche con un livello elevato di approfondimento e sofisticazione. Senza in alcun modo sottovalutare l'importanza fondamentale del pensiero dello studioso nell'elaborazione mentale dei dati, è però possibile notare, mutuando arditamente dalla terminologia computazionale e linguistica, come queste analisi sembrino condividere una capacità descrittiva piuttosto elevata e una capacità predittiva e rappresentativa più debole, nel senso che sono molto accurate nel descrivere approfonditamente singoli fenomeni ma raramente riescono (e forse neppure vogliono) a predire i comportamenti generali dei fenomeni stessi e le relative influenze.

Una domanda potrebbe quindi essere posta, in modo lieve e stimolante: che cosa succederebbe se uno studioso invece di leggere ogni singolo *isnād* e costruirsi man mano un sistema di organizzazione della conoscenza potesse avere istantaneamente a disposizione una grande quantità di informazioni sui trasmettitori e sulle tradizioni che provengono direttamente dal testo stesso, già organizzate e facilmente accessibili attraverso un sistema di relazioni interne?

4.1 Il valore 'computazionale' dei *ḥadīṯ*

Si provi ora a guardare un singolo *ḥadīṯ* immaginando di indossare un paio di 'occhiali computazionali' e si legga il testo allo scopo di trovare, o almeno ipotizzare, strutture interne e preesistenti che possano suggerire nuovi procedimenti di organizzazione delle informazioni contenute. A supporto di questo tentativo mutuiamo dalla ricerca storiografica recente alcune suggestioni, in particolare sulla valorizzazione dell'analisi degli anacronismi e delle interpretazioni intenzionalmente anacronistiche come metodo di ricerca. Nel tentativo di superare la comune definizione di anacronismo come l'applicazione cronologicamente errata di categorie appartenenti a un dato contesto temporale su degli eventi che non vi appartengono, alcuni studiosi, Nicole Laroux in particolare, mostrano come l'analisi di ciò che sembra un anacronismo possa essere in certa misura utile per la ridefinizione di alcune categorie interpretative (Laroux, 1993).

Si tenterà quindi in questa sede un modesto approccio anacronistico e interdisciplinare mirante a considerare le collezioni di *ḥadīṯ* come se fossero state originariamente concepite come dei database *ante litteram*, dei sistemi cioè chiaramente strutturati nei quali le informazioni sono codificate in modo formale usando uno specifico 'linguaggio di programmazione' interno.

4.2 Le raccolte di *ḥadīṯ* come basi di dati native

Applicando quindi un'interpretazione anacronistica i *ḥadīṯ* sembrerebbero concepiti basandosi all'origine sugli stessi assunti teorici che regolano la strutturazione dei dati secondo le recenti discipline di teorie dell'informazione. Lo scopo di questo lavoro non è tuttavia indagare se i primi compilatori e studiosi islamici di tradizioni avessero effettivamente concepito il loro lavoro in chiave proto-computazionale, bensì quello di verificare se, date queste premesse, nuovi approcci di analisi sono prospettabili e soprattutto praticabili (Lancioni, 2008)[47].

4.2.1 Database e linguaggi descrittivi

Per cercare di capire se la comparazione tra la struttura nativa delle raccolte di tradizioni e quella di una tipica base di dati abbia valore da un punto di vista scientifico, verranno affrontati alcuni aspetti rilevanti delle discipline informatiche e della teoria dell'informazione.

Nel quadro della teoria computazionale della strutturazione delle informazioni un database è una collezione organizzata di dati digitali che, modellando alcuni aspetti della realtà, fornisce supporto a quei processi che necessitano di informazioni correlate (Kroenke & Auer, 2007). Alcuni recenti

[47] Le speculazioni che cercano consciamente di identificare 'anticipazioni di teorie successive' nel pensiero arabo-islamico sono comunque piuttosto intriganti, anche solo in termini di *divertissement*. A titolo di esempio si pensi alle forti analogie esistenti tra le entità linguistiche definite dai primi grammatici arabi in termini di *mubtada'* e *ḫabar* e il moderni concetti di *topic and comment* approfonditi in particolare dalla Scuola di Praga e dalla grammatica funzionale di Halliday (Bohas & Guillaume, 1984).

modelli di rappresentazione (Brass, 2008) concordano che un database può essere immaginato come una griglia tabulare nella quale ogni riga (entrata o *record*) contiene un insieme discreto di informazioni, che ogni colonna (campo, ingl. *field)* classifica poi in differenti tipologie o categorie. Incro- Incrociando le righe di entrata con le colonne di campo si ottengono la celle, che rappresentano la più piccola unità di informazione della base di dati e sono fornite di due coordinate: la prima identifica la cella come appartenente a un'entrata specifica, la seconda etichetta e tipizza, attraverso il relativo campo, il contenuto stesso.

La griglia tabulare, seppur presente e visibile all'utente in molte basi di dati[48], è però solamente una rappresentazione grafica bidimensionale di qualcosa codificato in linguaggio macchina in modo piuttosto differente. A livelli profondi di elaborazione i database sono organizzati in un unico livello dimensionale usando metalinguaggi descrittivi che alternano in un'unica, continua stringa di testo i dati (*record*) e le strutture (*field*). Tali stringhe tesuali contengono perciò due livelli, uno che identifica il contenuto l'altro che lo organizza e ne stabilisce le relazioni interne ed esterne. Quest'ultimo livello descrittivo, che coincide con il concetto di marcatura (*tagging*) già visto per l'XML al titolo 3.2.2, consente appunto di gestire un contenuto di dati attraverso l'uso regolare di espressioni specifiche appartenenti a una lista chiusa di descrittori:

```
<record1> <field1> contenuto1 </field1> <field2> contenuto2
</field2> </record1> <record2> <field1> contenuto3 </field1>
<field2> contenuto4 </field2> </record2>
```

Nell'esempio l'idea del doppio livello di articolazione è espressa chiaramente: ciascuna unità testuale inclusa tra i simboli '<' e '>' assume una funzione meta-testuale specifica che descrive le restanti unità annidate di informazione. L'intero paragrafo in questione può essere considerato un database a tutti gli effetti: seppure le unità di testo si susseguono l'un l'altra linearmente, esse possono essere convertite in una rappresentazione

[48] Ad esempio i fogli di lavoro come Microsoft Excel o i database come SQL o Microsoft Access.

tabulare usando la piccola 'grammatica di regole' contenuta nel livello meta-testuale e descrittivo del testo stesso:

	Field 1	Field 2
Record 1	contenuto 1	contenuto 2
Record 2	contenuto 3	contenuto 4

4.2.2 Alcune parole dei *ḥadīṯ* come marcatori di contenuto: le espressioni funzionali (EF)

Una collezione di tradizioni può dunque essere interpretata come un database in cui i singoli *ḥadīṯ* sono le entrate (record). Ciascun *ḥadīṯ* è come se organizzasse le proprie unità di informazione (l'*isnād* con la catena di trasmissione e il *matn*) attraverso un set di definito di campi, che segmenta e separa le diverse unità e le connette l'una all'altra.

Tale supposto database non assume la forma della rappresentazione tabulare, bensì è come fosse espresso in modo lineare nel testo stesso attraverso una sorta di linguaggio di marcatura interna la cui funzione è appunto di separare unità di informazione. Questo manca ovviamente del grado di formalizzazione di linguaggi come l'XML, e pertanto non userà, per la marcatura dei confini tra informazione testuale e meta-testuale. dei segni riservati espliciti (ad esempio i caratteri '>' e '<'), bensì alcune classi di parole a cui darà nuovo significato extra linguistico e che verranno in questa sede definite 'espressioni funzionali' (EF).

Le EF sono costituite da quegli elementi testuali che separano e marcano i trasmettitori, come *ḥaddaṯanā*, *ḥaddaṯanī*, *'aḫbaranā*, *aḫbaranī*, *anba'anā*, *sami'a*, *'an* (cfr. titolo 2.2.1.1). Le EF quindi, oltre al loro significato grammaticale e semantico nel testo, mostrano un'ulteriore specifica funzione meta-testuale di marcatura, e si comportano come dei veri e propri *tag*. L'esempio seguente mostra l'interpretazione di un *ḥadīṯ* secondo questo modello, con un commento interlineare che evidenzia l'alternanza tra informazione testuale e meta-testuale:

«ḥaddathanī aḥmadu bnu 'ishkābi ḥaddathanā muḥammadu bnu fuḍaylin
|---FE#1----| |-----INFO#1-----------| |---FE#2----| |----------INFO#2----------|

'an 'umārata bni l-qa'qā' 'an abī zur'ata 'an abī hurayrata
|FE#3||-----INFO#3----------| |FE#4| |-INFO#4--| |FE#5| |----INFO#4---

raḍī (a)llāhu 'anhu qāla qāla l-nabiyyu ṣalla llāhu 'alayhi wa-sallama
(<<<<<EULOGY>>>>>)| |FE#6| |INFO#5 :*MATN*---- *(<<<<<EULOGY>>>>>>>>>>>>>>>>>>)*

kalimatāni habībatāni ila l-raḥmāni khafīfatāni 'ala l-lisāni [...]»[49].
----------INFO#5 [*MATN*]---------- |

La stringa di testo "*muḥammadu bnu fuḍaylin*" corrisponde ovviamente a un nome proprio di persona, in questo caso di un trasmettitore, ma grazie al significato meta-testuale delle adiacenti EF altre informazioni, implicite o non immediatamente evidenti possono essere recuperate, in particolare la posizione del trasmettitore nella catena e come egli ha ricevuto e trasmesso il *matn*:

INFORMAZIONE		TIPO DI INFORMAZIONE
Nome del trasmettitore:	Muḥammad Ibn Fuḍayl	Esplicitata
Posizione nella catena:	2	Implicita
Matn ricevuto da:	transmitter 3 (typology: 'an)	Implicita
Matn trasmesso a:	transmitter 1 (typology: ḥaddathanā)	Implicita

Le EF sembrano quindi poter coprire le seguenti funzioni extralinguistiche:

- segnalare la successiva presenza di un'unità di informazione (il nome del trasmettitore nell'*isnād* oppure l'intera sequenza del *matn*);
- specificare la tipologia dell'unità di informazione, in particolare la modalità di trasmissione, sebbene la presenza di una EF piuttosto non si riferisca necessariamente a un sistema di

[49] (Al-Buḫārī, Saḥīḥ, p. 162 (vol. IX)).

classificazione su cui la critica della Tradizione abbia un consenso unanime (cfr. 2.2.1.1);

- ❖ in certi contesti marcare la direzione del flusso di informazione all'interno della catena di trasmissione: da x a y o da y a x, assumendo che x e y siano due trasmettitori consecutivi nel testo;
- ❖ infine, la posizione dei trasmettitori stessi all'interno della stringa testuale dell'*isnād* fornisce un'informazione implicita ma essenziale, vale a dire l'ordine numerico dei nodi attraverso i quali la trasmissione è avvenuta[50].

In genere i linguaggi descrittivi come l'HTML e l'XML utilizzano un *tag* di apertura e uno di chiusura per delimitare il segmento di informazione. Nel caso dell'ipotizzato 'linguaggio descrittivo' dei *ḥadīṯ* invece, solo il *tag* di apertura è espresso nel testo.

4.2.2.1 Categorie di EF

Le EF possono essere raggruppate in più categorie[51] in dipendenza della loro funzione di segmentazione:

- ❖ separare l'*isnād* dal *matn*;
- ❖ individuare i trasmettitori all'interno dell'*isnād*.

Entrambe le categorie sono interessate da alcuni problemi l'approccio ai quali influenza di molto la successiva scelta delle strategie computazionali di segmentazione.

La EF più ricorrentemente impiegata per segnalare il passaggio dall'*isnād* al *matn* è la voce verbale *qāla* (o *qālat* al femminile), 'ha detto'. Poiché questo termine a causa del suo significato molto comune può occorrere ovunque in un *ḥadīṯ*, anche nel *matn*, senza necessariamente essere una EF, come può la macchina distinguere correttamente quando *qāla* è associato a una funzione meta-testuale e quando no? È cioè possibile, in altre parole, assegnare il

[50] Questa funzione può apparire ovvia al lettore ma i computer non possono procedere per ragionamento ellittico come gli esseri umani ma necessitano della piena esplicitazione dei dati prima di procedere al loro trattamento.

[51] le categorie qui pertinenti sono due, ma una terza categoria è da considerarsi in sede di trattamento computazionale puro (cfr. 4.4.3.3).

giusto valore di EF semplicemente esaminando il testo così com'è senza il supporto di alcun aiuto esterno, precisamente il ragionamento umano? Il problema sarà qui risolto facendo ricorso ad alcune condizioni restrittive di tipo statistico-posizionale, ad esempio dichiarare che *qāla* è una EF solo quando ricorre per la prima volta in un *ḥadīṯ* senza essere seguita a breve distanza da un'altra EF tipica invece della segmentazione interna dell'*isnād* (cfr. 4.4.3.4).

Un'altra questione impegnativa che coinvolge a un certo livello anche l'interpretazione critica delle fonti, è la necessità di identificare correttamente l'inizio del *matn* laddove la struttura testuale si presenta criptica. È il caso ad esempio di quando il profeta Muḥammad o uno dei suoi Compagni possa essere considerato come ultimo[52] trasmettitore piuttosto che parte del *matn* stesso o viceversa.

Alcune considerazioni linguistiche possono invece essere compiute sulle EF tipiche dell'*isnād*. Queste EF sono quasi sempre verbi e talvolta preposizioni: ci si potrebbe domandare se esista un qualche tipo di correlazione tra la natura delle categorie grammaticali a cui appartengono le EF e le informazioni meta-testuali di tipologia e direzione della trasmissione. A un livello di pragmatica linguistica le preposizioni possono essere interpretate come operatori nei quali il valore funzionale è più forte di quello semantico e tende a costruire relazioni (spaziali o temporali) tra due elementi[53]. I verbi invece sono entità più complesse: essi possono generalmente reggere più argomenti rispetto alle preposizioni, hanno un marcato valore lessicale e semantico e costruiscono relazioni sfaccettate tra gli argomenti stessi e i ruoli tematici coinvolti.

Nel nostro modello interpretativo computazionale della struttura dei *ḥadīṯ*, sia verbi sia preposizioni possono agire come EF, ma assumendo per vere le considerazioni di tipo pragmatico appena esposte, sembra evidente che i verbi EF siano potenzialmente più precisi nel definire l'ambiente meta-testuale rispetto alle preposizioni EF. Si prenda ad esempio il caso di due EF

[52] Ultimo ovviamente se si considera l'ordine in cui il nome compare nel ma primo in prospettiva cronologica.
[53] Si veda ad esempio (Zelinsky-Wibbelt, 1993) o (Rice, 1999).

come il verbo *ḥaddatha[nā/nī]* ('egli [mi/ci] ha narrato') e la preposizione *'an*. Mentre la EF preposizionale specifica solamente la direzione di trasmissione da un nodo all'altro, la EF verbale, grazie al suo 'timbro' semantico (COME lo ha trasmesso? Lo ha narrato) e alla disponibilità di esaurire tre argomenti (CHI, A CHI e COSA ha trasmesso?) è potenzialmente in grado di fornire più meta-informazioni a un livello più dettagliato. Questa ipotizzabile 'debolezza' della preposizione 'an rivela alcune attraenti analogie con le scienze di validazione delle tradizioni islamiche, dove appunto *'an* è considerato il modo più ambiguo, generico e indeterminato con il quale alcuni compilatori (tra cui al-Buḫārī) marcano la tipologia di trasmissione (Azami, 1977).

4.2.2.2 Le eulogie come EF?

Le eulogie[54] contenute nei *ḥadīṯ*, sono elementi formulaici appartenenti a un insieme chiuso di espressioni che accompagnano certi elementi testuali (nomi propri) ma essenzialmente non aggiungono dati al flusso di informazione del contenuto. Attraverso però la 'lettura computazionale', questi elementi appaiono sì esterni al contenuto informativo, ma associati in modo regolare ad alcune sue parti e pienamente dipendenti da esse. Mutuando dal lessico della fonologia potremmo dire che le eulogie si comportano quasi come 'varianti contestuali', in quanto sono selezionate da uno specifico contesto (la presenza del nome di Muḥammad o di un suo Compagno) e non sono liberamente utilizzabili dal compilatore come 'portatrici di nuova informazione'. Le regole contestuali alla base della selezione delle eulogie possono invece indicare altre informazioni di tipo meta-testuale, utili alla disambiguazione del testo in sede di trattamento informatico. Poiché esse nel testo sono sempre precedute da nomi propri di persona, diventano degli indicatori potenti di *named entities*: trovare un'eulogia (operazione piuttosto semplice in quanto appartenente a una lista chiusa di elementi) significa trovare sicuramente un nome proprio.

[54] Intendiamo qui per 'eulogie' le *ṣalawāt o* formule di benedizione e protezione che quasi sempre accompagnano il nome del profeta Muhammad o dei suoi Compagni e Successori, ad esempio *ṣallā 'llāhu 'alayhi wa-sallama.*

Inoltre poiché le eulogie sono differenziate e riservate nell'uso a seconda dell'importanza del personaggio, esse si rivelano utili per modellare criteri di classificazione progressiva (*ranking*) e quindi comparazioni automatiche sulla coerenza interna delle catene di trasmissione.

In conclusione le EF, e in una certa misura le eulogie, sembrano in grado di organizzare il contenuto informativo di un *ḥadīṯ* attraverso la propagazione di relazioni multidimensionali. Esse possono essere interpretate come i 'mattoni', o meglio gli operatori principali, di una raffinata ' grammatica dell'informazione' che nasce e si sviluppa dall'interno del testo stesso.

4.2.3 Dalla teoria all'applicazione: l'identificazione semi-automatica della struttura di un *ḥadīṯ*

Una volta verificata la consistenza scientifica dell'accostamento tra struttura esplicita di una base di dati e struttura implicita di una raccolta di *ḥadīṯ*, insistere ulteriormente sul piano teorico se effettivamente i compilatori di *ḥadīṯ* avessero chiara in mente detta organizzazione concettuale rischia di essere uno sterile esercizio speculativo. In mancanza cioè di un'applicazione pratica di questa modellizzazione, gli studiosi di critica delle fonti potrebbero facilmente evidenziarne lo scarso significato per la loro ricerca. Come ricordato nell'introduzione questa preoccupazione si ricollega a uno degli scopi principali che le *digital humanities* dovrebbero avere, e cioè fornire strumenti computazionali per migliorare il lavoro di un esperto nel proprio campo d'indagine umanistica. Le concezioni teoriche devono quindi essere indirizzate a innervare la concezione di strumenti d'analisi quantitativa e qualitativa al servizio di un'area di ricerca tradizionalmente caratterizzata dall'approccio analogico ed esclusivamente umano. Assumere che un *ḥadīṯ* e un record di un database condividono un certo grado di somiglianza nella propria struttura interna deve pertanto significare che il computer sia in grado di elaborarne i dati in modo similare e che quindi sia possibile realizzare dei programmi che trattino il testo scritto, ne riconoscano la struttura e la rendano esplicita attraverso l'estrazione e l'ordinamento delle informazioni contenute.

4.3 Le espressioni regolari per interpretare una struttura testuale

La costruzione di un programma informatico che possa analizzare la struttura dei *ḥadīṯ* trattandoli come fossero in un certo senso semplicemente un database da convertire, è stata preceduta dall'individuazione della sintassi delle espressioni regolari come metodo specificamente computazionale per l'identificazione e il reperimento di dati all'interno di un testo.

4.3.1 Definizione

Il termine 'espressione regolare' (*regular expression*) proviene dalle discipline matematiche e informatiche, dove contraddistingue un tratto tipico delle espressioni matematiche basato sulla nozione di regolarità. La cosiddetta 'sintassi delle espressioni regolari' è quindi un metodo per la ricerca delle regolarità in una serie di dati, e fu concepita negli anni 50 da Kleene come uno strumento della teoria degli automi per la descrizione di linguaggi formali (Bird & Klein, 2006). In origine le espressioni regolari erano applicate solamente negli ambienti teorici legati alle scienze matematiche, ma Thompson e altri presto costruirono implementazioni specifiche per l'utilizzo delle espressioni regolari anche nei processi non-deterministici di automazione a stati finiti, e da qui nei principali ambienti e linguaggi di programmazione, come Perl, .NET(C#), Java, Ruby, Python (Friedl, 2002).

Un'espressione regolare (conosciuta anche con l'abbreviazione *regex* o l'acronimo inglese RE) è una stringa di testo formalmente strutturato per la descrizione di complessi schemi di ricerca (*pattern*), che vengono in seguito applicati a unità testuali più lunghe allo scopo di trovare corrispondenze (*match*). Le RE possono venire efficacemente adoperate per svolgere le seguenti operazioni:

- ricerca (*search*): consiste nello spostarsi lungo una stringa per trovare una sottostringa che corrisponde a un dato schema;
- appaiamento (*match*): consiste nel verificare se una data stringa corrisponde pienamente a un dato schema;

- ❖ sostituzione (*replace*): la stringa di testo che corrisponde a un dato schema viene sostituita con un'altra;
- ❖ segmentazione (*split*): in accordo a uno schema che identifica alcuni caratteri o parole come separatori, un blocco di testo viene suddiviso in una serie di stringhe.

4.3.2 Sintassi e funzionamento

Lo schema (*pattern*) di una RE è costituito da una sequenza di caratteri, alcuni dei quali dotati di funzioni riservate. Se l'intero insieme di caratteri alfanumerici può essere utilizzato nel proprio significato letterale (ad esempio 'a' significa 'a'), due ulteriori sottoclassi definite di caratteri corrispondono a simboli e operatori. La classe dei simboli serve per indicare in modo compatto alcuni insiemi e categorie alfanumeriche, mentre gli operatori applicano funzioni specifiche alla porzione di testo interessato (Kuchling, 2002).

4.3.2.1 I caratteri simbolo

In questa classe sono compresi caratteri o loro combinazioni il cui significato include categorie alfanumeriche più ampie o stringhe, ad esempio nel modulo per le espressioni regolari del linguaggio di programmazione Python:

SIMBOLO	SIGNIFICATO NELLA RE
.	qualunque carattere
/w	qualunque carattere alfanumerico
/s	qualunque spaziatura (spazio, tabulazione, interruzione di riga, interruzione di paragrafo)
/d	qualunque cifra numerica
/D	qualunque carattere non numerico

Leggendo le combinazioni tipiche di caratteri simbolo si può notare come attraverso l'anteposizione del simbolo '/' il carattere successivo cessa di rappresentare il proprio valore letterale e ne assume uno simbolico.

4.3.2.2 I caratteri operatore

I caratteri operatore servono a indicare delle modalità specifiche con le quali effettuare la ricerca sulla sequenza di caratteri immediatamente precedente. Essi possono essere quantificatori o indicatori di posizionamento, secondo la tabella seguente, in cui *x* rappresenta l'elemento a cui si applica l'operazione:

OPERATORE	SIGNIFICATO NELLA RE
`x*`	cerca x zero o più volte (x può esserci o non esserci)
`+`	cerca x una o più volte (x deve comparire almeno una volta)
`?`	cerca zero o una volta sola (x deve comparire solo una volta oppure non comparire affatto)
`x{n}`	cerca x per un numero n di volte
`^x`	x deve trovarsi all'inizio della stringa
`X$`	x deve trovarsi alla fine della stringa
`x\b`	x deve trovarsi all'inizio di parola
`x|y`	cerca x oppure y

4.3.2.2 La sintassi

Alcuni operatori possono essere combinati insieme, ad esempio l'uso di `*?` permette di cercare la sequenza minima di caratteri che soddisfa le condizioni e ignorare le sequenze più lunghe[55]. Ad esempio `(xy)*?` nella stringa `'xyxyx'` restituirà solo `xy` e non `xyxy`.

Combinazioni più complesse di operatori permettono poi di raffinare le ricerche posizionali o di etichettare in un certo modo le informazioni trovate:

OPERATORE	SIGNIFICATO NELLA RE
`x(?=y)`	trova x solo se seguito da y
`x(?!y)`	trova x solo se non seguito da y
`(?<=y)x`	trova x solo se preceduto da y
`x(?P=nome)`	trova x ed etichettalo sotto la categoria 'nome'

[55] Nella sintassi delle RE queste combinazioni sono correntemente chiamate *not greedy*, 'non avide' in quanto cercano la minima combinazione possibile e tendono a economizzare risorse.

I caratteri appartenenti alle tre classi testo, simbolo e operatore sono poi variamente combinati in una stringa lineare secondo semplici regole sintattiche che prevedono la successione degli elementi e l'utilizzo di parentesi tonde per la nidificazione come nelle equazioni matematiche ordinarie. Tali combinazioni permettono di formalizzare ricerche complesse in una forma astratta e simbolica comprensibile all'elaboratore. Ad esempio, l'istruzione discorsiva "trova qualsiasi parola che sia compresa tra la parola 'il', oppure 'lo', oppure 'un' e la parola 'libro' " è tradotta nella seguente, brevissima RE:

```
(il|lo|un)+.+libro+
```

e troverà, se applicata ad esempio alla stringa 'un bellissimo libro' l'occorrenza 'bellissimo'. Ancora, l'istruzione "trova solo una volta uno o più spazi seguiti da un numero" verrà tradotta nella seguente RE:

```
\s+?(?=\d+)
```

che, usando una rappresentazione interlineare significa:

Trova	almeno una volta	uno o più	spazi	seguiti da	una o più	cifre
	?	+	\s	(?=)	+	\d

4.3.2.3 Funzionamento: la ricerca di uno schema RE in una stringa testuale

La sintassi delle RE può essere utilizzata all'interno dei linguaggi di programmazione come Python attraverso un modulo specifico, che prevede il seguente ciclo di elaborazione:

- l'espressione regolare viene scritta rispettando le regole sintattiche relative e memorizzata nel sistema;
- una serie di metodi predefiniti dal sistema cercano lo schema descritto dalla RE in un dato testo in input, e quando trovano una corrispondenza applicano al testo una delle operazioni possibili descritte al titolo 4.3.1 (ricerca, appaiamento, sostituzione, segmentazione).

4.3.3 Le espressioni regolari e l'interpretazione del testo

4.3.3.1 RE e analisi testuale

Grazie alla ricca sintassi e all'alto grado di rappresentazione astratta che provvedono, le RE sono potenti e profondamente espressive, e la loro applicazione si è presto diffusa ben al di là delle discipline matematiche per cui erano nate, soprattutto nei campi di elaborazione del linguaggio naturale (NLP) e analisi del testo (Goyvaens & Levitan, 2009).

Le RE furono originariamente concepite per il trattamento delle lingue artificiali, ivi compresi i linguaggi di programmazione, in cui l'ambiguità è programmaticamente evitata e le strutture di riferimento sono sempre chiare ed esplicitate. Quando le RE trattano linguaggi naturali, che sono naturalmente permeati di ambiguità, ellissi e significati impliciti, è evidente che l'efficacia e i risultati di impiego dipendono fortemente dal livello di organizzazione interna e non equivocità dei testi oggetto d'analisi (Bird, Klein, & Loper, 2009).

Oggi numerose strategie di NLP impiegano proficuamente le RE in vari campi d'applicazione, come la ricerca di concordanze, l'identificazione dei temi di parola, la tokenizzazione, la normalizzazione del testo, la segmentazione e la lemmatizzazione (Jackson P. M., 2002). Il fatto è però che, a oggi, la letteratura e le applicazioni relative alle RE trattano prevalentemente di testi in inglese o in lingue che utilizzano l'alfabeto latino, che favoriscono alcune strategie di RE e ne impediscono altre. In tal modo la lingua dei testi processati condiziona e tipizza fortemente la capacità di analisi e i risultati ottenibili attraverso l'uso delle RE stesse Ad esempio le RE aiutano a identificare i nomi propri solo in quelle lingue che hanno sistemi di scrittura che prevedono la variante maiuscola e la utilizzano prevalentemente per la designazione appunto dei nomi propri, e questo è vero per l'inglese o l'italiano ma non per il tedesco o l'arabo (Bird & Klein, 2006).

Il grado di efficienza delle RE aumenta poi se sono soddisfatte alcune condizioni, come l'esplicitazione (ad esempio i confini di parola sono contraddistinti da segni espliciti, come gli spazi bianchi) o la regolarità (ad

esempio tutti i confini di parola sono indicati sempre e solo con uno spazio bianco). È senz'altro possibile progettare strategie di approccio basate sulle RE per qualsiasi lingua, ma per ogni testo considerato, il valore di questi due parametri ne definisce e talvolta limita l'efficacia di analisi.

4.3.3.2 Le potenzialità delle RE per il trattamento di testo arabo

Per quanto riguarda le tecniche computazionali di trattamento del testo arabo esse, tendono a considerare la riduzione a temi (*stemming*) e l'annotazione morfologica come prerequisiti essenziali anche per quegli approcci tipicamente non morfologici, come l'estrazione di informazione, il *parsing* sintattico, e la conversione dal testo scritto al discorso orale (*text to speech*) (Abu El-Khair, 2007).

L'impiego delle RE come primo approccio al testo arabo può ragionevolmente alterare i paradigmi teorici in cui la morfologia è così preponderante (cfr. Capitolo 1). Poiché le RE semplicemente agiscono sulla 'superficie' del testo, esse sembrano in grado di fornire in modo quasi automatico un gran numero di informazioni utili, a prescindere e prima di qualsiasi pretrattamento o analisi 'pesante' di natura morfologica.

La letteratura sull'uso delle RE per l'analisi del testo arabo è quasi inesistente, ma sono degni di nota alcuni lavori nei quali le RE sono impiegate in modo originale e sofisticato per interrogare corpora testuali in lingua araba. Attraverso le RE il testo è investigato su due livelli, quello delle singole parole (o addirittura morfemi) e quello di gruppi di esse. Sul primo livello le RE permettono di estrarre i temi radicali dalle parole, mostrando in questo modo il potenziale di trattamenti 'leggeri' rispetto agli approcci più 'pesanti ad esempio degli analizzatori morfologici tradizionali (Kouloughli D. , 2008). Per quanto riguarda invece la ricerca di gruppi di parole, le RE permettono di estrarre facilmente collocazioni, sintagmi e associazioni di parole dal testo arabo, usando rapide interrogazioni di tipo semi-automatico (Kouloughli D. , 2009).

In questi lavori le RE sono applicate attraverso l'uso di specifici programmi già presenti sul mercato il cui scopo è permettere all'utente di interrogare un testo e creare *ad hoc* i propri schemi da ricercare attraverso

un'interfaccia grafica. In modo analogo, molte applicazioni comuni di word processing impiegano le RE per offrire servizi di ricerca avanzata all'utente.

4.3.3.3 Le RE come chiave di lettura per testi strutturati

Le RE non sono soltanto un potente e avanzato sistema di ricerca di stringhe: il titolo 4.4 tratterà la possibilità di utilizzarle per estrarre informazione e segmentare automaticamente il testo arabo dei *ḥadīṯ*. Un tale approccio rappresenta un esempio di come far evolvere lo status delle RE da strumento per cercare qualcosa a 'chiave' che permette, in una certa misura, la lettura e la decodificazione automatica dell'organizzazione testuale. Il prerequisito è comunque quello di avere come input un testo dotato di un certo grado di formalizzazione a livello dei propri elementi strutturali.

Uno studio di qualche anno fa mostra le potenzialità del processo di analisi di testi descrittivi di matematica e conseguente traduzione in linguaggio simbolico (Wolska & Kruijff-Korbayová, 2004). Le autrici esaminano la formalizzazione dei simbolismi e delle formule usate nei manuali per l'insegnamento della matematica e ipotizzano un modello che:

- trova le regolarità del testo;
- le utilizza come schemi per estrarre informazione testuale e meta-testuale;
- costruisce una lista di regole basate sugli schemi allo scopo di tradurre automaticamente le espressioni discorsive analitiche in formule matematiche sintetiche, e viceversa.

Questo e altri studi (Bird & Klein, 2006) confermano che la segmentazione può essere usata non solo per identificare stringhe discrete ma anche per cercare di assegnare, attraverso una mappatura delle regolarità e delle ricorrenze, una struttura globale al testo stesso. Questa struttura viene governata da una sorta di "grammatica di regole" contestuale, che controlla anche le connessioni tra il contenuto dell'informazione e la sua organizzazione logica.

4.4 *HadExtractor*: un programma per la segmentazione automatica e l'estrazione di informazione

Le collezioni di *ḥadīṯ* possono essere interpretate come un insieme di parti chiaramente strutturate, nel quale le informazioni sono state in qualche misura 'codificate' attraverso l'uso di una semplice grammatica di regole che definisce una sorta di linguaggio di programmazione interno. La sintassi delle espressioni regolari si adatta perfettamente a tale strategia automatica di conversione del testo originale in una base di dati completa, nella quale le informazioni testuali e meta-testuali sono esplicitamente immagazzinate e gerarchizzate.

4.4.1 Funzioni del programma e diagramma di funzionamento

Per verificare l'applicazione di questo modello è stato concepito un programma originale di elaborazione dati, chiamato *HadExtractor*, che sfrutta la sintassi delle RE per segmentare il testo dei *ḥadīṯ* ed estrarne automaticamente un certo numero di informazioni.

HadExtractor è concepito con tre funzioni principali:

- leggere un'intera collezione di tradizioni e identificare i singoli *ḥadīṯ*;
- segmentare ogni *ḥadīṯ* in *isnād* e *matn*;
- estrarre da ogni *isnād* tutti i nomi dei trasmettitori insieme con le alcune informazioni supplementari (posizione, ordine, direzione e tipologia di trasmissione).

HadExtractor ha le tipiche caratteristiche di un programma *single-run*: una volta che è stato eseguito sul testo di input, produce il relativo output e conclude la sua funzione.

La figura 4.1 mostra il diagramma di funzionamento del processo automatico di elaborazione analizzato attraverso le strategie algoritmiche e le trasformazioni successive subite dal flusso di dati. Ogni modulo di elaborazione sarà quindi analizzato in dettaglio.

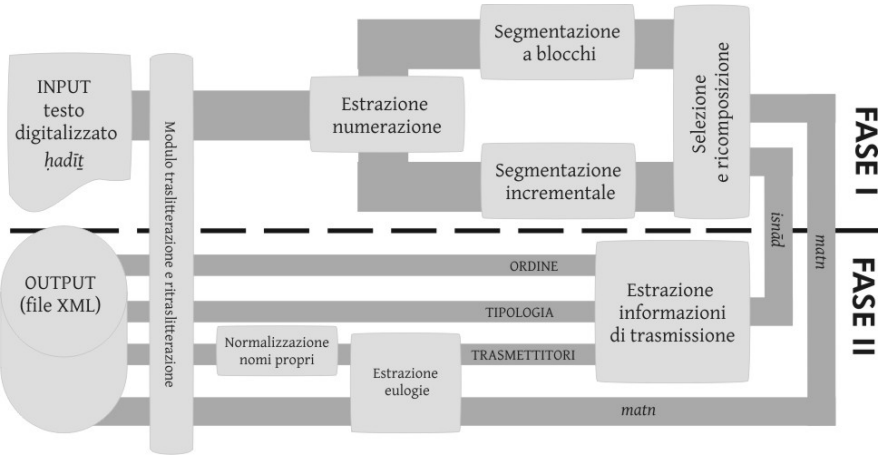


Figura 4.1 Diagramma di funzionamento dei processi di *HadExtractor*

4.4.2 Input: pretrattamento delle fonti

Poiché le RE che costituiscono il nucleo di funzionamento del programma agiscono sulla superficie di una stringa così come essa si presenta, il testo in input non necessita di alcun tipo di pretrattamento manuale, come operazioni di marcatura o normalizzazione.

L'unica operazione preliminare eseguita è la traslitterazione dell'intero testo in input secondo il sistema Buckwalter modificato (Lancioni, 2011) e il modulo di programma appositamente creato (cfr. 3.5.3 e 3.5.4).

Di seguito un esempio di come si presenta una frazione del testo in input, prima e dopo il processo automatico di traslitterazione:

36 - حَدَّثَنَا حَرَمِيُّ بْنُ حَفْصٍ قَالَ حَدَّثَنَا عَبْدُ الْوَاحِدِ قَالَ حَدَّثَنَا عُمَارَةُ قَالَ حَدَّثَنَا أَبُو زُرْعَةَ بْنُ عَمْرِو بْنِ جَرِيرٍ قَالَ سَمِعْتُ أَبَا هُرَيْرَةَ عَنْ النَّبِيِّ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ انْتَدَبَ اللَّهُ لِمَنْ خَرَجَ فِي سَبِيلِهِ لَا يُخْرِجُهُ إِلَّا إِيمَانٌ بِي وَتَصْدِيقٌ بِرُسُلِي أَنْ أُرْجِعَهُ بِمَا نَالَ مِنْ أَجْرٍ أَوْ غَنِيمَةٍ أَوْ أُدْخِلَهُ الْجَنَّةَ وَلَوْلَا أَنْ أَشُقَّ عَلَى أُمَّتِي مَا قَعَدْتُ خَلْفَ سَرِيَّةٍ وَلَوَدِدْتُ أَنِّي أُقْتَلُ فِي سَبِيلِ اللَّهِ ثُمَّ أُحْيَا ثُمَّ أُقْتَلُ ثُمَّ أُحْيَا ثُمَّ أُقْتَلُ

بَاب تَطَوُّعُ قِيَامِ رَمَضَانَ مِنْ الْإِيمَانِ

37 - حَدَّثَنَا إِسْمَاعِيلُ قَالَ حَدَّثَنِي مَالِكٌ عَنْ ابْنِ شِهَابٍ عَنْ حُمَيْدِ بْنِ عَبْدِ الرَّحْمَنِ عَنْ أَبِي هُرَيْرَةَ أَنَّ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ قَالَ مَنْ قَامَ رَمَضَانَ إِيمَانًا وَاحْتِسَابًا غُفِرَ لَهُ مَا تَقَدَّمَ مِنْ ذَنْبِهِ[56]

```
37 - Had+aCanaA eis_maAciylu qaAla Had+aCaniy maAlikU can_ Ab_ni
XihaAbI can_ Humay_di b_ni cab_di Alr+aH_mani can_ Eabiy
huray_raoa Ean+a rasuwla All+ahi Sal+aY All+ahu calay_hi
wasal+ama qaAla man_ qaAma ramaDaAna eiymaAnNA waAH_tisaAbNA
Gufira lahu maA taqad+ama min_ Van_bihi
baAb Saw_mu ramaDaAna AH_tisaAbNA min_ Al_eiymaAni
38 - Had+aCanaA Ab_nu salaAmI qaAla Eax_baranaA muHam+adu b_nu
fuDay_lI qaAla Had+aCanaA yaH_yaY b_nu saciydI can_ Eabiy
salamaoa can_ Eabiy huray_raoa qaAla qaAla rasuwlu All+ahi Sal+aY
All+ahu calay_hi wasal+ama man_ SaAma ramaDaAna eiymaAnNA
waAH_tisaAbNA Gufira lahu maA taqad+ama min_ Van_bihi
```

4.4.3 Fase I: numerazione e segmentazione generale

4.4.3.1 Identificazione dei singoli *ḥadīṯ*

In merito alla separazione tra un *ḥadīṯ* e l'altro, gli elementi strutturali rilevabili nel testo consistono principalmente di un'interruzione di riga seguita dal numero identificativo del *ḥadīṯ*, a sua volta seguito da un trattino. In questo caso il numero costituisce un elemento distintivo unico per marcare l'inizio del *ḥadīṯ* in quanto nell'intero testo non sono presenti cifre numeriche con altra funzione.

La prima RE impiegata da *HadExtractor* è del tipo più semplice, e permette di segmentare l'intera collezione in singoli *ḥadīṯ* basandosi sul riconoscimento automatico del punto in cui comincia il numero identificativo. Poiché quindi nel testo le cifre sono usate unicamente per indicare la numerazione dei *ḥadīṯ*, sarà sufficiente impostare il seguente schema di espressione regolare, che cerca zero o più interruzioni di riga seguite da almeno un carattere numerico:

```
\s*?(?=\d+?)
```

[56] Tratto dal primo file corrispondente al testo digitalizzato del primo volume del Ṣaḥīḥ (Al-Buḫārī, Ṣaḥīḥ al-Bukhārī).

La RE viene poi integrata in un'istruzione di programma che effettua la segmentazione del testo nel punto immediatamente precedente al numero trovato:

```
p = re.compile(r'\s*?(?=\d+?)')
q = p.split(text)
```

La prima linea di codice stabilisce lo schema della RE e la seconda lo applica per segmentare il testo. Questa RE è un esempio base di come sia possibile segmentare un testo in più elementi attraverso l'identificazione di un elemento ricorrente non ambiguo (il numero in questo caso).

4.4.3.2 Numerazione e divisione in volumi, libri e capitoli

Come si è scritto, ogni *ḥadīṯ* è preceduto da un numero che lo identifica, il quale quindi è facilmente estraibile una semplice RE. Per trovare invece le altre informazioni generali di divisione in volumi, libri e capitoli occorrono strategie più definite.

Il numero di volume a cui appartiene un *ḥadīṯ* (il Ṣaḥīḥ è organizzato in nove volumi) è estraibile automaticamente, ma in modo implicito e in qualche misura pragmatico, in quanto esso non è esplicitamente indicato nel testo ma risulta dal fatto che i file di input siano nove e contengano ciascuno un volume. Sarà quindi sufficiente, una volta che il file di input è sottoposto al sistema, far eseguire al programma una *routine*[57] che marchi ogni *ḥadīṯ* trovato con il numero di volume corrispondente, ricavabile dal numero progressivo del file in input.

Il numero e il titolo del libro vengono individuati con un'altra strategia ancora, considerando che i titoli stessi sono presenti in modo continuo nel testo e non sono identificati da elementi grafici numerazioni, bensì dal fatto di cominciare con la parola *kitāb* ed essere preceduti o seguiti dalla formula

[57] Procedimento di elaborazione ricorsiva.

della *basmala*[58]. Un'apposita RE garantirà che vengano trovate solo le stringhe che cominciano con *kitāb* e ma che sono accompagnate dalla *basmala*:

```
\s(?=bis_mi All\+ahi Alr\+aH_mani Alr\+aHiymi\skitaAb\s+?|kitaAb
.*\sbis_mi All\+ahi Alr\+aH_mani Alr\+aHiymi\s+?)+
```

La RE dell'esempio precedente significa "trova un qualsiasi spazio seguito o meno dal testo della *basmala*, seguito da uno spazio, seguito dalla parola `kitaAb`; oppure trova la parola `kitaAb` seguita da una qualsiasi successione di caratteri seguita dal testo della *basmala*". La numerazione del libro è stata poi aggiunta in modo automatico numerando semplicemente in modo progressivo tutte le occorrenze `kitaAb.*` trovate.

Infine il titolo dei vari capitoli all'interno di un libro è recuperato con un'altra semplice RE che cerca `baAb`, la parola araba usata nella collezione per indicare 'capitolo', ma solo se immediatamente preceduta da un'interruzione di paragrafo, in modo da escludere tutte le altre occorrenze non rilevanti.:

```
\s*(?=baAb\s+?)
```

Nell'esempio seguente vien mostrato un estratto del testo in input dove a fini esplicativi sono indicati in grassetto le parti relative a libro e capitolo, in modo da evidenziarne la posizione rispetto agli altri elementi in cui è organizzato il testo:

```
2046 - Had+aCanaA cab_du All+ahi b_nu muHam+adI Had+aCanaA
hiXaAmU Eax_baranaA mac_marU can_ Alz+uh_riy+i can_ cur_waoa can_
caAJiXaoa raDiya All+ahu can_haA Ean+ahaA kaAnat_ turaj+ilu
Aln+abiy+a Sal+aY All+ahu calay_hi wasal+ama wahiya HaAJiDU
wahuwa muc_takifU fiy Al_mas_jidi wahiya fiy Huj_ratihaA
yunaAwiluhaA raE_sahu
bis_mi All+ahi Alr+aH_mani Alr+aHiymi
kitaAb Al_buyuwci waqaw_lu All+ahi caz+a wajal+a {waEaHal+a
All+ahu Al_bay_ca waHar+ama Alr+ibaA} waqaw_luhu {eil+aA Ean_
takuwna tijaAraoN HaADiraoN tudiyruwnahaA bay_nakum_}
baAbu maA jaA-a fiy qaw_li All+ahi tacaAlaY
```

[58] La *basmala* identifica l'espressione *bi-smi llahi l-raḥmāni l-raḥīmi* 'Nel nome di Dio Clemente e Misericordioso', tipica formula ingressiva islamica all'inizio di un testo o di una delle sue suddivisioni.

```
2047 - Had+aCanaA Eabuw Al_yamaAni Had+aCanaA Xucay_bU can_
Alz+uh_riy+i [...]
```

4.4.3.3 Due approcci per l'identificazione delle espressioni funzionali (EF)

Per procedere alla vera e propria segmentazione di un *ḥadīṯ* nelle sue parti costitutive, è necessario definire con precisione la lista delle EF sulla quale basare le operazioni di ricerca tramite le RE (cfr. 2.2.1.1 e 4.2.2).

Per costruire tale lista due approcci sono disponibili:

- approccio basato sulla conoscenza: la lista viene compilata manualmente attingendo dalla letteratura di settore (cfr. 2.2.1.1) e dalla lettura diretta di un certo numero di *ḥadīṯ* per individuarne le EF meno comuni;
- approccio empirico: l'intero testo in input viene sottoposto a una ricerca automatica usando delle RE specifiche che cercano di individuare le EF attraverso l'esplicitazione di alcune loro caratteristiche superficiali, ad esempio il fatto di trovarsi solitamente nella prima parte del testo (riservata appunto all'*isnād*), di essere a breve distanza le une dalle altre (essendo infatti separate tra loro solo dai nomi dei trasmettitori e non da vere e proprie parti testuali), e di ricorrere in modo consistente.

Per sperimentare il secondo approccio è stato pertanto studiato un piccolo programma specifico, esterno a *HadExtractor*, che identifica come EF tutte quelle parole (*token*) che si trovano nella prima metà del testo, si ripetono almeno due volte e sono precedute o seguite, a distanza di non più di cinque parole, da altre parole con le stesse proprietà.

I risultati dell'applicazione del programma concordano sostanzialmente con quelli dell'approccio 'manuale'. Seppure il primo approccio appare più pratico e percorribile, il secondo mostra la potenza dell'analisi di superficie, che permette di trovare degli elementi significanti semplicemente in base ad algoritmi che ne studiano la posizione nel testo e la prossimità con altri elementi.

Le EF identificate con un'integrazione dei due approcci appartengono a tre categorie a seconda della funzione:

- ❖ EF che marcano l'inizio del *matn* e lo sperano dall'*isnād* (EF_MAT);
- ❖ EF che separano i vari trasmettitori nell' *isnād* (EF_ISN);
- ❖ EF di 'accompagnamento': si trovano variamente prima o dopo le altre EF o per ragioni sintattiche e di reggenza o per marcare l'inizio di un discorso diretto. Ai fini dell'estrazione di informazione possono essere considerati elementi ridondanti e non necessari (EF_EXT).

Nella tabella successiva sono riportate tutte le EF individuate con le rispettive funzioni e significati[59]:

CODICE PER LE RE	SIGNIFICATO LINGUISTICO	FUNZIONI POSSIBILI
`qaAl(a|at) | (y|t)aquwlu`	*qāla* 'ha detto' *qālat* 'ha detto' (femm.) *yaqūlu* 'dice' *taqūlu* 'dice' (femm.)	EF_MAT EF_EXT
`Ean+a(.*\s)`	*'anna* [+eventuale pronome suffisso]	EF_MAT EF_EXT
`Had+aCan(iy|aA)`	*ḥaddaṯanī* 'mi ha raccontato' *ḥaddaṯanā* 'ci ha raccontato'	EF_ISN
`Eax_bara (niy|naA|hu)`	*aḫbaranī* 'mi ha informato' *aḫbaranā* 'ci ha informato' *aḫbarahu* 'lo ha informato'	EF_ISN
`samic_(tu|a)`	*sami'tu* 'ho ascoltato' *sami'a* 'ha ascoltato'	EF_ISN
`Ean_baEan(iy|aA)`	*anba'anī* 'mi ha annunciato' *anba'anā* 'ci ha annunciato'	EF_ISN
`can_`	*'an*, 'in base all'autorità di'	EF_ISN

4.4.3.4 La segmentazione a blocchi

La segmentazione a blocchi rappresenta la strategia principale di *HadExtractor*, e consiste nel cercare come prima cosa l'elemento di separazione tra *isnād* e *matn*, e solo in seguito analizzare la composizione interna dell'*isnād*.

[59] Ai fini della segmentazione e dell'estrazione il significato linguistico è non rilevante.

La RE impiegata per questa segmentazione è particolarmente complessa, poiché lo scopo non è trovare una particolare stringa di testo bensì modellare in uno schema simbolico l'intera struttura di un singolo *ḥadīṯ*. L'esempio seguente mostra la RE così come appare all'interno di *HadExtractor*, salvo l'accorgimento di aver qui sostituito il lungo elenco delle EF tipiche dell'*isnād* con la sigla già nota EF_ISN, per favorirne la leggibilità:

```
1 |   ^(?P<num>\d+?)
2 |   [ ]*?-?[ ]*?
3 |   (?P<isn>.*?)
4 |   (Ean\+a+?.{0,50}?qaAla(?:t_|A|[ ])+?)|
5 |   (Ean\+a(?:humaA|hum_|hu|haA|[ ])|
6 |   qaAla(?:t_|A|[ ])|(?:y|t)aquwlu+?)
7 |   (?!.{0,100}(EF_ISN)))
8 |   (?P<mat>.*?)$
```

Analizzando linea per linea:

Riga 1: Il simbolo `^` indica l'inizio della stringa, mentre la RE tra parentesi recupera il numero identificativo di *ḥadīṯ* e lo etichetta con il *tag* `num`.

Riga 2: Riconosce il trattino '-' presente dopo il numero. Poiché nel testo originale il trattino è spesso ma non sempre (a causa dei possibili errori umani di digitalizzazione) circondato da spazi bianchi, questa RE mostra la sua flessibilità tenendo conto di tutte le alternative possibili (numero-spazio-trattino-spazio-parola; numero-trattino-parola; numero-trattino spazio-parola; numero-spazio-trattino-parola).

Riga 3: Questa riga identifica tutto il testo da dopo il trattino fino ai limiti imposti dalla prossima condizione come facente parte dell'*isnād*, e significa "Trova almeno zero o più caratteri e etichettali sotto il *tag* `isn`". La dimensione del testo trovato dipende unicamente dal soddisfacimento della condizione alla riga successiva.

Righe 4-7: Provano a identificare il punto di separazione tra *isnād* e *matn*, ipotizzando la necessaria presenza della EF relativa. Sulla base della lista di cui al titolo precedente le uniche EF

con funzione adatta sono due, *qāla* e *anna*. La RE è lunga e intricata, in quanto deve essere in grado di individuare solo i *qāla* e *anna* con funzione di separazione (EF_MAT) e non tutti quelli presenti con una certa frequenza nel resto del testo. La RE deve risolvere due problemi focali, l'alternanza (la EF appartiene a una lista e il suo valore non è sempre lo stesso) e l'ambiguità (non tutte le corrispondenze nel testo sono effettivamente EF). L'alternanza può essere risolta attraverso l'uso del simbolo '|', che ha il valore dell'operatore booleano 'OR' e permette quindi la presenza di uno qualsiasi degli elementi dichiarati. Il complesso problema dell'ambiguità può essere risolto tentando di far emulare alla RE alcuni aspetti del ragionamento umano in particolare la capacità di cogliere ellissi, analogie e riferimenti deittici. Son pertanto state concepite alcune condizioni posizionali, che assegnano a una parola la funzione di EF_MAT solo se essa è preceduta o seguita a una certa distanza da un elemento dato. Le condizioni presenti nelle righe 4-7 della RE sono due, a titolo esemplificativo ma verrà qui analizzata solo la seconda, più rappresentativa (righe 6-7). Le due righe significano "Trova la stringa *qāla* (o le forme flesse *qālat* (III p. sing. femm.) o *qālā* (III p. duale masch.) oppure la stringa *yaqūlu* (o la forma III p. sing femm. *taqūlu*) solo se esse non sono seguite a distanza di 100 caratteri da una tipica FE_ISN". Presupponendo su basi empiriche che la probabile lunghezza massima di un nome proprio in arabo sia inferiore ai 100 caratteri, la RE cercherà quindi solo gli elementi FE_MAT non seguiti a breve distanza da una tipica FE_ISN. Questo è possibile proprio in quanto i nomi propri nell'*isnād* sono sempre seguiti da una EF che marca la trasmissione successiva, e quindi il primo *qāla* che si incontra nel testo non più seguito dalle EF tipiche dell'*isnād* è il punto di separazione tra *isnād* e *matn*. Regole di questo tipo sono ovviamente concepite a base statistica, tentando cioè con

tentativi successivi di raffinamento della regola di ottenere il maggior numero possibile di risultati positivi. L'uso di queste strategie altamente dipendenti dal contesto e caratterizzate da un approccio a tentativi ed errori non sono certamente pienamente efficaci, ma sono in grado di ridurre fortemente l'ambiguità e di conseguenza gli errori di segmentazione[60].

Riga 8: La porzione di RE qui contenuta è molto semplice, e significa "trova tutto il testo da qui fino alla fine del *ḥadīṯ* e considerarlo *matn*, marcandolo con l'etichetta `mat`.

4.4.3.5 Una strategia alternativa: la segmentazione incrementale

Per migliorare i risultati della segmentazione è stata inoltre studiato un approccio completamente differente, che al posto di trovare come prima cosa la FE_MAT, prevede invece l'identificazione progressiva di una FE_ISN dopo l'altra, fino a he, quando il sistema non ne trova più, stabilisce l'inizio del *matn*. Seppur affascinante dal punto di vista della concezione teorica, alla verifica dei dati l'implementazione di tale strategia non ha mostrato un efficienza superiore alla strategia di segmentazione a blocchi, per cui è stata integrata in *HadExtractor* solo come soluzione alternativa laddove la prima strategia fallisce e non restituisce segmentazioni coerenti.

4.4.4 Fase II: *isnād* e estrazione di informazione

Una volta che l'*isnād* è stato identificato, esso è formato sostanzialmente da due tipologie di testo, il nome del trasmettitore e la EF che ne stabilisce la tipologia di trasmissione. La lista dei nomi dei trasmettitori è ovviamente numericamente consistente, di tipo aperto e quindi molto difficile da determinare *a priori*. Invece le EF_ISN, come si è visto in precedenza, appartengono a un insieme chiuso e quindi descrivibile.

Data quindi la natura essenzialmente 'binaria' della struttura dell'*isnād*, se gli elementi dell'insieme chiuso sono tutti identificabili, quelli dell'insieme aperto ugualmente saranno recuperabili per esclusione, dichia-

[60] Sulla valutazione dettagliata dell'efficacia di queste strategie cfr. 6.2.3.1.

rando cioè che tutto il testo compreso tra due EF_ISN corrisponde a un nome proprio di persona. In questo caso la RE da prevedere è nuovamente molto semplice, e ha la seguente struttura, dove FE_ISN sta per uno qualsiasi degli elementi appartenenti alla lista:

```
(FE)(.*?)(FE)
```

4.4.4.1 Segmentazione automatica dell'*isnād*

L'*isnād* è quindi automaticamente segmentato da *HadExtractor* e corredato dalle informazioni di trasmissione fornite dalla FE_ISN e dall'ordine progressivo di comparizione nell'*isnād* stesso.

4.4.4.2 Identificazione delle eulogie

Come mostrato in 4.2.2.2 le formule eulogiche (*ṣalawāt)* presenti nei *ḥadīṯ* costituiscono un'importante categoria di elementi, che non ha funzioni di strutturazione del testo, ma piuttosto di accompagnamento a determinate classi di nomi propri di persona. Poiché le eulogie costituiscono un insieme chiuso e conoscibile e sono sempre selezionate dal contesto, una semplice RE è stata implementata in *HadExtractor* con lo scopo di identificarle sia nell'*isnād* sia nel *matn* e marcarle come 'parti esterne' alla struttura globale di un *ḥadīṯ*.

4.4.4.3 Tipizzazione dei nomi propri

Poiché nella lingua araba classica i nomi propri prevedono la flessione di caso tipica di altre classi nominali, l'occorrenza letterale di essi nel testo dei *ḥadīṯ* può essere leggermente diversa a seconda del caso di appartenenza. Inoltre, nell'uso di espressioni onomastiche patronimiche e di filiazione come 'padre di' o 'figlio di', l'arabo prevede varie forme allografe sia di *ab* ('padre) sia di *ibn* (figlio), a seconda del caso di appartenenza o dei processi di assimilazione della prima consonante di parola. Lo stesso nome proprio può quindi essere scritto diversamente a seconda del contesto linguistico e testuale di appartenenza.

Se per la segmentazione del testo la questione è scarsamente rilevante, essa è invece fondamentale per l'utilizzo successivo dei dati segmentati per altri scopi, ad esempio la rappresentazione della consistenza e dei rapporti 'genealogici' tra i trasmettitori. È stato quindi previsto un modulo interno a *HadExtractor* che provvede alla normalizzazione e alla tipizzazione dei nomi propri, usando le funzioni di sostituzione previste dalla sintassi delle RE (per i dettagli cfr..4.5.1).

4.4.5 Output: Organizzazione delle informazioni in un file XML

Una volta compiuta l'elaborazione dell'input, *HadExtractor* produce come risultato un unico file in formato XML, dove il testo originale è preservato nel suo ordine naturale ma segmentato e annotato in tutte le sue componenti. Ignorando quindi le parti di testo contenenti i *tag* specifici XML e contraddistinte dall'essere sempre comprese tra i simboli < e >, è possibile rileggere l'intero testo nella sua integrità. Questo comportamento 'conservativo' del testo non impedisce però di estrarre dallo stesso file tutte le informazioni meta-testuali ivi annotate.
L'esempio seguente mostra l'output XML relativo alla segmentazione di un singolo *ḥadīṯ*, così come appare in un programma di visualizzazione. Il contenuto si presenta qui ancora in versione traslitterata:

```
</hadith>
<hadith id_ar="2">
    <source_info>
        <vol>1</vol>
        <book>1</book>
        <num>2</num>
    </source_info>
    <isn>
        <trasm type="Had+aCanaA">cab_du All+ahi b_nu yuwsufa</trasm>
        <trasm type="qaAla Eax_baranaA">maAlikU</trasm>
        <trasm type="can_ ">hiXaAmi b_ni cur_waoa</trasm>
        <trasm type="can_ ">Eabiyhi</trasm>
        <trasm type="can_ ">caAJiXaoa Eum+i Al_muW_miniyna <eul>raDiya All+ahu
        can_haA</eul> </trasm>
    </isn>
    <sep>Ean+a </sep>
    <mat>Al_HaAriCa b_na hiXaAmI raDiya All+ahu can_hu saEala rasuwla All+ahi
    Sal+aY All+ahu calay_hi wasal+ama faqaAla yaA rasuwla All+ahi kay_fa
    yaE_tiyka Al_waH_yu faqaAla rasuwlu All+ahi Sal+aY All+ahu calay_hi wasal+ama
    EaH_yaAnNA yaE_tiyniy miC_la Sal_Salaoi Al_jarasi wahuwa EaXad+uhu calay+a
    fayuf_Samu can+iy waqad_ wacay_tu can_hu maA qaAla waEaH_yaAnNA yatamaC+alu
    liy Al_malaku rajulNA fayukal+imuniy faEaciy maA yaquwlu qaAlat_ caAJiXaou
    raDiya All+ahu can_haA walaqad_ raEay_tuhu yan_zilu calay_hi Al_waH_yu fiy
    Al_yaw_mi AlX+adiydi Al_bar_di fayaf_Simu can_hu waein+a jabiynahu
    layatafaS+adu caraqNA</mat>
```

Lo stesso file XML è anche prodotto in output con la traslitterazione del testo in caratteri arabi. L'esempio seguente mostra, a titolo di esempio, la prima parte dello stesso *ḥadīṯ* mostrato sopra:

```
<hadith id_ar="2">
    <source_info>
        <vol>1</vol>
        <book>1</book>
        <num>2</num>
    </source_info>
    <isn>
        <trasm type="خلاثنا"> عبيدُ الله بنُ يوسُفَ </trasm>
        <trasm type="قال أخبرنا"> مالكٌ </trasm>
        <trasm type="عن "> هشام بن عروة </trasm>
        <trasm type="عن "> أبيهِ </trasm>
        <trasm type="عن"> عائشةَ أمِ المؤمنين<eul> رضيَ الله عنها</eul></trasm>
    </isn>
    <sep>أنْ </sep>
```

Il sistema di marcatura e organizzazione del testo segmentato viene descritto nelle sue parti nei titoli successivi, utilizzando la versione traslitterata per comodità di visualizzazione.

4.4.5.1 Numerazione e collocazione nella collezione

Il singolo *ḥadīṯ* è marcato dai *tag* di apertura `<hadith>` e chiusura `</hadith>`[61]:

```
<hadith id_ar="2">
[...]
</hadith>
```

Questo *tag* è caratterizzato inoltre dall'attributo `id_ar,` usato per assegnare un numero identificativo univoco allo *ḥadīṯ*, indipendente dal sistema di numerazione interno alla raccolta.

Il *tag* `<source_info>` apre la sezione in cui sono contenute le informazioni estratte relativamente a numero di volume `<vol>`, numero di libro `<book>` e numero di *ḥadīṯ* secondo l'edizione della raccolta `<num>`:

```
<source_info>
     <vol>1</vol>
     <book>1</book>
     <num>2</num>
</source_info>
```

4.4.5.2 Trasmettitori, ordine e tipologia di trasmissione e *matn*

Il *tag* `<isn>` contiene l'*isnād* segmentato. Ogni nome di trasmettitore è indicato dal *tag* `<trasm>`, che possiede un attributo `type=` che specifica la EF utilizzata per la trasmissione e ne indica quindi la tipologia. La progressione dei trasmettitori e la loro posizione numerica all'interno della trasmissione non è indicata in modo esplicito per evitare ridondanza di *tagging*, ma è comunque comprensibile al computer e ricavabile attraverso il conteggio della successione dei *tag* `<trasm>`:

```
<isn>
     <trasm type="Had+aCanaA">cab_du All+ahi b_nu yuwsufa</trasm>
     <trasm type="qaAla Eax_baranaA">maAlikU</trasm>
     <trasm type="can_ ">hiXaAmi b_ni cur_waoa</trasm>
```

[61] Da qui in avanti verranno descritti solo i tag di apertura. Si tenga presente che a ciascuno di essi corrisponde necessariamente il relativo tag di chiusura, identico nella forma tranne che per la presenza del simbolo / prima del nome del tag stesso.

```
    <trasm type="can_ ">Eabiyhi</trasm>
<trasm type="can_ ">caAJiXaoa Eum+i Al_muW_miniyna <eul>raDiya
All+ahu can_haA</eul> </trasm>
</isn>
```

La EF di separazione tra *isnād* e *matn* è indicata con il *tag* <sep>:

```
<sep>Ean+a </sep
```

Infine, l'intero testo del *matn* è identificato dal *tag* <mat>:

```
<mat>Al_HaAriCa b_na hiXaAmI <eul> raDiya All+ahu can_hu </eul>
saEala rasuwla All+ahi <eul> Sal+aY All+ahu calay_hi wasal+ama
</eul> faqaAla yaA rasuwla All+ahi kay_fa yaE_tiyka Al_waH_yu
faqaAla rasuwlu All+ahi <eul> Sal+aY All+ahu calay_hi wasal+ama
</eul> EaH_yaAnNA yaE_tiyniy miC_la Sal_Salaoi Al_jarasi wahuwa
EaXad+uhu calay+a fayuf_Samu can+iy waqad_ wacay_tu can_hu maA
qaAla waEaH_yaAnNA yatamaC+alu liy Al_malaku rajulNA
fayukal+imuniy faEaciy maA yaquwlu qaAlat_ caAJiXaou <eul> raDiya
All+ahu can_haA </eul> walaqad_ raEay_tuhu yan_zilu calay_hi
Al_waH_yu fiy Al_yaw_mi AlX+adiydi Al_bar_di fayaf_Simu can_hu
waein+a jabiynahu layatafaS+adu caraqNA</mat>
```

4.4.5.3 Altri *tagging* non strutturali

Le eulogie sono marcate direttamente nel testo attraverso il *tag* <eul>, e possono trovarsi sia nella sezione <isn> che in quella <mat> (cfr. esempi precedenti).

4.5 Rappresentazione dell'informazione

Il trattamento di una collezione di *ḥadīṯ* con il programma *HadExtractor*, oltre a permettere una segmentazione completa del testo rende disponibili una serie d'informazioni per le quali è possibile prevedere alcuni modelli di rappresentazione. In particolare il trattamento sull'*isnād* fornisce le seguenti categorie di informazioni:

- i nomi dei trasmettitori;
- la loro posizione reciproca nella singola catena di trasmissione di un *ḥadīṯ*, che in genere dal compilatore ultimo risale fino al profeta Muḥammad ;

- la tipologia di trasmissione, che comprende informazioni sul modo con cui l'informazione è stata trasmessa e sulla direzione di trasmissione.

Queste categorie di dati possono essere interpretate come facenti parte di un modello nel quale degli 'oggetti' (i nomi dei trasmettitori) intrattengono tra loro 'relazioni' di vario tipo. Queste relazioni non sono solo interne a una singola catena di trasmissione ma, grazie al fatto che molti dei trasmettitori tramandano più di una tradizione, l'insieme delle relazioni tra di essi costituisce una complessa rete informativa, di cui si può tentare una rappresentazione di tipo bi- o multidimensionale.

4.5.2 La Rappresentazione a grafi

Il modello ipotizzabile per le catene di trasmissione, vale a dire una serie di oggetti collegati tra loro da una rete di relazioni è indubbiamente simile a quello della teoria geometrico-combinatoria dei grafi (Berge, 1958), nella quale un grafo è definito come una struttura costituita da:

- oggetti semplici, detti nodi (*nodes*) o vertici (*vertices*);
- collegamenti tra gli oggetti, che possono essere orientati (archi, ingl. *arcs*) o non orientati (spigoli, ingl. *edges*);
- eventuali informazioni supplementari associati ai nodi o agli archi.

I nodi e gli archi possono quindi essere rappresentati su una griglia bidimensionale attraverso l'uso di algoritmi specifici che ne permettono una visualizzazione grafica. In queste rappresentazioni la forma grafica è una variabile non essenziale, poiché solo i nodi e le relazioni tra essi sono costanti non modificabili (Bondy & Murty, 2008). Solitamente i nodi sono rappresentati come dei punti o dei cerchi e gli archi come delle linee. Ai punti e alle linee possono essere o meno associate delle etichette contenenti un certo numero di informazioni.

Un altro concetto essenziale è la 'visita' del grafo, che consiste nell'evidenziazione di un nodo in particolare e nell'eventuale trasformazione dinamica del grafo in modo che il nodo visitato si trovi al centro delle relazioni intrattenute con gli altri nodi.

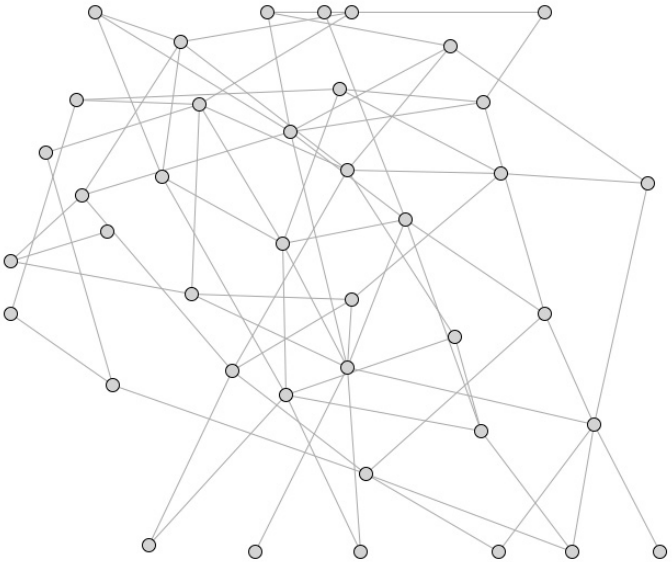


Figura 4.2: Esempio di rappresentazione bidimensionale di un grafo.
Fonte: http://pellacini.di.uniroma1.it/teaching/fondamenti12/lecture17.html

La teoria dei grafi sembra quindi essere uno strumento di rappresentazione ideale delle catene di trasmissione dei *ḥadīṯ*.

4.5.3 Normalizzazione e pretrattamento dei dati

Prima di procedere al modello di interpretazione, i dati da rappresentare necessitano di alcune operazioni di pretrattamento, in particolare la normalizzazione dei nodi e la formalizzazione delle relazioni tra essi.

La normalizzazione prevede che i nodi siano identificati da informazioni univoche e non ambigue. Nel caso dei nomi dei trasmettitori, nel testo originale dell'isnād alcuni di essi appaiono caratterizzati da alcuni tratti di ambiguità:

- uno stesso nome può comparire in forme diverse a seconda della flessione di caso, e questo può interessare fenomeni legati alle

vocali brevi (ad es. `Ab_nu cumara` e `Ab_ni cumara`) o alle vocali lunghe (ad es. `Eabuw Al_yamaAni` e `Eabiy Al_yamaAni`);

- ❖ al posto di un nome per esteso può essere indicata unicamente la relazione di parentela con il trasmettitore precedente (ad es. `Eabiyhi` per `EabuW saAlimi b_ni cab_di All+ahi`).

La normalizzazione di queste forme è automaticamente compiuta da un modulo interno di *HadExtractor*, come descritto al titolo 4.4.4.3.

La formalizzazione dei dati richiede che essi siano organizzati in una forma interpretabile dal programma che poi li trasformerà in nodi e archi. Attraverso uno *script* specifico, i dati relativi *all'isnād* sono quindi preventivamente estratti dal file XML e organizzati secondo una lista di triplette contenenti il nome del trasmettitore, l'ordine in cui esso compare nella catena e la tipologia di trasmissione, secondo il seguente esempio:

```
t_list= [(Eabuw Al_yamaAni, 5,Had+aCanaA),
      (Xucay_bU, 4, Eax_baranaA),
      (Alz+uh_riy+i, 3, can_),
      (caAmiru b_nu sac_di b_ni Eabiy waq+aASI, 2, Eax_baraniy),
        (sac_dI, 1, can_)]
```

4.5.4 *ChainViewer*: un programma per la rappresentazione

Sulla base della letteratura inerente alla produzione di grafi (Di Battista, Eades, Tamassia, & Tollis, 1999; Kaufmann & Wagner, 2001; Zelle, 2003) è stato concepito e sviluppato in Python un programma di nome *ChainViewer* che:

- ❖ carica tutte le informazioni formalizzate di cui al titolo 4.5.2 (nome dei trasmettitori, ordine e tipologia di trasmissione);
- ❖ per ogni tripletta di informazioni relativa a un singolo *ḥadīṯ* assegna ai nomi dei trasmettitori il valore di nodi (*nodes*), alle relazioni (tra il trasmettitore considerato, il precedente e il successivo) quello di archi (*edges*), e trasforma la tipologia di trasmissione in un attributo qualitativo degli archi;

- ❖ attraverso una serie di algoritmi a orientamento variabile genera i grafi per ogni singola catena o unisce insieme un numero variabile di catene nella stessa rappresentazione.

ChainViewer è ancora in una fase di sviluppo. Alcuni problemi devono essere risolti in termini di scelta degli algoritmi, in quanto attualmente essi sono in grado di generare grafi solo per un numero limitato di catene contemporaneamente. L'obiettivo è di ottenere una rappresentazione contemporanea e completa di tutte le relazioni di trasmissione relative alla raccolta di *ḥadīṯ* di al-Buḫārī, attraverso l'uso di movimenti dinamici di espansione e compressione in dipendenze delle visite compiute ai nodi.

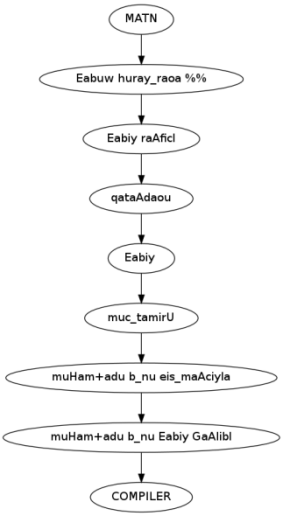


Figura 4.3: esempio di rappresentazione automatica a grafi di un *isnād* orientata sul compilatore e senza l'indicazione della tipologia di trasmissione (il simbolo %% indica la presenza di una formula eulogica).

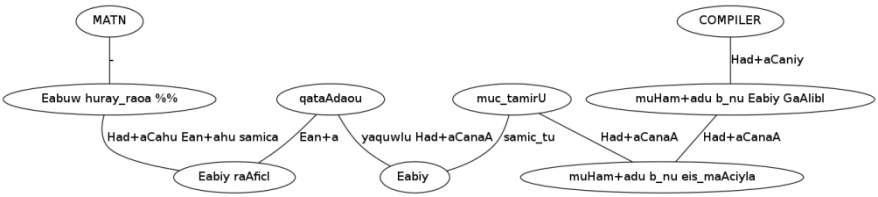


Figura 4.4: esempio di rappresentazione automatica a grafi dello stesso *isnād* di figura 4.3 ma non orientata e con indicazione della tipologia di trasmissione.

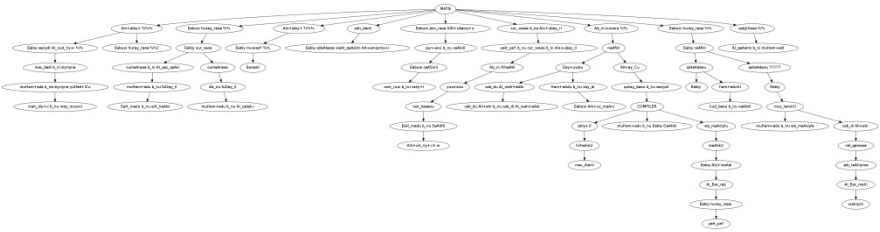


Figura 4.5[62]: esempio di rappresentazione automatica a grafi di sedici *isnād*, orientata sul *matn* e senza tipologia di trasmissione. I nodi a cui arrivano o da cui partono più frecce sono i trasmettitori comuni a più *isnād*.

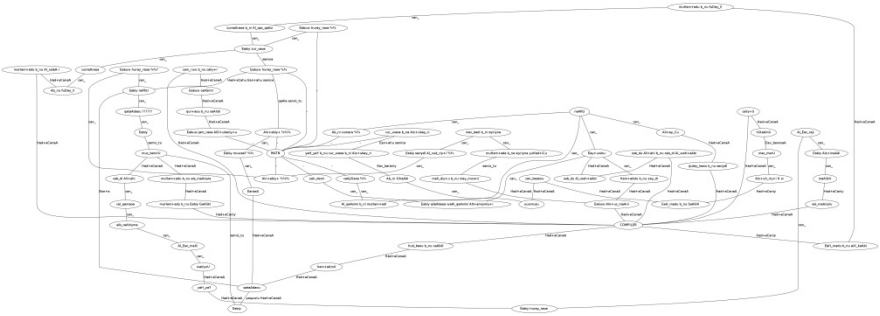


Figura 4.6: esempio di rappresentazione automatica a grafi di undici *isnād* doppiamente orientata su *matn* e compilatore e con tipologie di trasmissione.

[62] Seppur il testo sia difficilmente leggibile, è possibile apprezzare l'impostazione generale di nodi e archi.

5. Analisi dei *ḥadīṯ*: all'interno e all'esterno del testo

Questo capitolo tratta di due possibili livelli di analisi dei *ḥadīṯ* più specificamente linguistici. Il piano morfologico è scelto per mostrare una possibile analisi computazionale di una componente 'interna al testo, mentre il piano 'esterno' delle relazioni del testo con altri testi è esaminato attraverso alcune strategie di ricerca testuale e allineamento bilingue

5.1 Esplorazione del livello morfologico

In questa parte si affronta la descrizione di uno strumento esistente per l'analisi morfologica del testo arabo. Per considerare alcune specificità legate alla lingua dei *ḥadīṯ*, il sistema vien quindi modificato per aumentarne l'efficienza di analisi, e poter fornire così al testo dei *ḥadīṯ* un primo livello non-supervisionato di lemmatizzazione e annotazione morfologica.

5.1.2 *AraMorph*: un esempio di analizzatore morfologico compatto

AraMorph è insieme un analizzatore morfologico e un lemmatizzatore realizzato da Buckwalter nell'ambito tipico dei modelli che impiegano le tecnologie a stati finiti in un contesto di sistemi basati sulla conoscenza (Buckwalter, 2002). Rispetto ad altri sistemi analoghi come ad esempio l'analizzatore Xerox (Xerox, 2013), è *open source*, presenta una struttura algoritmica lineare ed elegante nella sua semplicità, capace di funzionare efficientemente in contesti di procedure automatiche non supervisionate (*un-*

supervised). La sua struttura modulare ne permette inoltre agevolmente l'espansione e l'integrazione con altri strumenti[63].

In contrasto con altri sistemi, dove l'apparente mancanza di linearità di molti morfemi arabi enfatizza la necessità di progettare componenti morfologiche complesse e multilivello, *AraMorph* tratta strategicamente le parole[64] come elementi linearmente scomponibili in tre componenti, un prefisso, un tema (*stem*) e un suffisso. Il tema è l'unico componente a dover essere sempre presente, laddove invece prefissi e suffissi possono avere valenza nulla ma essere comunque indicati, a un certo livello di rappresentazione, come 'morfemi zero' (*zero morphemes*).

AraMorph viene alimentato quindi da due tipi di risorse linguistiche:

- tre liste contenenti ciascuna il più alto numero di prefissi, suffissi e temi postulabili, e che costituiscono i dizionari base del sistema;
- tre tabelle di compatibilità, che marcano le combinazioni possibili nella stessa parola di prefissi e temi, temi e suffissi, prefissi e suffissi.

5.1.2.1 I dizionari

Nel caso dei dizionari, quelli di prefissi e suffissi sono relativamente completi data la finitezza intrinseca e la relativa consistenza del numero morfemi di tipo funzionale e grammaticale che una lingua può mostrare. Il dizionario dei temi è invece di tipo aperto e tendente alla perfettibilità, in quanto è sempre idealmente possibile aggiungere nuovi temi a base lessicale in dipendenza della ricchezza del lessico considerato. Se i prefissi enumerati sono circa quattrocento e i suffissi un migliaio, i temi ammontano a più di diecimila voci.

E' importante sottolineare come la definizione di 'prefisso' o 'suffisso' qui si riferisce non necessariamente a un unico morfema, ma potenzialmente a

[63] *AraMorph* è stato utilizzato in molti progetti, principalmente nella sua implementazione Java, ad esempio è incluso come strumento di analisi morfologica nell'*Arabic Wordnet Project,* una risorsa lessicale a reti semantiche per l'arabo (Black, et al., 2006).

[64] Il termine è qui usato in senso computazionale piuttosto che linguistico, intendendo con 'parola' qualsiasi sequenza di caratteri separata da spazi.

tutte le combinazioni di morfemi che possono precedere/seguire ciò che è definito come 'tema' nella lista relativa. Ad esempio l'espressione *bi-l-qalam* ('con la penna') è intesa dal sistema in questo modo:

bi | *l*	*qalam*	*i*
PREFISSO	TEMA	SUFFISSO

Ad esempio la preposizione *bi* occorrerà più volte nel dizionario dei prefissi in voci separate, a seconda che sia sola (*bi-*) o accompagnata da altri morfemi come l'articolo (*bi-l*). Questo processo di 'riduzione' a un'unica forma delle combinazioni multiple di morfemi consente una grande semplificazione del sistema, permettendo infatti di limitare la complessità del campo prefissale o suffissale. Ecco un esempio di tutte le voci del dizionario dei prefissi contenenti la preposizione *bi*[65]:

```
b       bi      NPref-Bi        by;with                 bi/PREP+

wb      wabi    NPref-Bi        and+ by/with            wa/CONJ+bi/PREP+

fb      fabi    NPref-Bi        and+ by/with            fa/CONJ+bi/PREP+

bAl     biAl    NPref-BiAl      with/by + the           bi/PREP+Al/DET+

wbAl    wabiAl  NPref-BiAl      and+ with/by the        wa/CONJ+bi/PREP+Al/DET+

fbAl    fabiAl  NPref-BiAl      and/so + with/by + the  fa/CONJ+bi/PREP+Al/DET+
```

Ogni voce di lista[66] presenta quattro campi principali, alcuni essenziali per l'elaborazione , altri a carattere di informazione supplementare:

- la forma vocalizzata del morfema (essenziale);

[65] Il testo arabo in questo e nei successivi esempi è reso attraverso il sistema di traslitterazione interno e originario usato da *AraMorph* (Buckwalter Arabic transliteration), cfr. 3.5.2. Il testo del codice è tratto dai file corredati all'analizzatore, reperibili sul sito http://www.nongnu.org/*AraMorph*

[66] E questo vale anche per le liste di temi e suffissi.

- la forma non vocalizzata del morfema (essenziale);
- la categoria grammaticale (informativo);
- una glossa traduttiva in inglese (informativo);
- una marcatura (*tagging*) della scomposizione interna in morfemi e del relativo valore, che verrà poi utilizzata dal sistema nella successiva elaborazione (essenziale).

Il dizionario dei suffissi è costruito in modo analogo. È interessante notare che in caso di morfemi suffissi a significato diverso ma omografi, la lista li elencherà tutte le volte necessarie. Ad esempio il suffisso *-kum*:

km	ukum	NSuff-h	your	+u/CASE_DEF_NOM+kum/POSS_PRON_2MP
km	akum	NSuff-h	your	+a/CASE_DEF_ACC+kum/POSS_PRON_2MP
km	ikum	NSuff-h	your	+i/CASE_DEF_GEN+kum/POSS_PRON_2MP
km	kum	NSuff-h	your	+kum/POSS_PRON_2MP
km	~ukum	PVSuff-~th	I <verb> you (pl.)	+tu/PVSUFF_SUBJ:1S+kum/PVSUFF_DO:2MP
km	~akum	PVSuff-~nh	they <verb> you	+na/PVSUFF_SUBJ:3FP+kum/PVSUFF_DO:2MP
km	akum	PVSuff-ah	he/it <verb> you	+a/PVSUFF_SUBJ:3MS+kum/PVSUFF_DO:2MP
km	kum	FWSuff-k	you [masc.pl.]	+kum/PRON_2MP
km	kum	IVSuff-k	you	+kum/IVSUFF_DO:2MP
km	kum	PVSuff-|h	they (both) <verb> you	+A/PVSUFF_SUBJ:3MD+kum/PVSUFF_DO:2MP

Le prime tre voci dell'esempio mostrano anche come la categoria di caso non venga considerata in un livello di analisi a parte, bensì interpretata co-

me semplice morfema in combinazione con altri a formare un unico suffisso.

Le voci del dizionario dei temi appaiono relativamente più semplici. Una caratteristica importante è che ogni plurale lessicale ('fratto') è associato al relativo singolare in una sorta di super-categoria. Ad esempio per *kitāb* ('libro'):

```
;; kitAb_1
ktAb kitAb          Ndu     book
ktb  kutub          N       books
```

La voce `ktAb`, se scritta in origine non vocalizzata, può anche corrispondere ad altri temi, e quindi occorrerà nel dizionario anche associata ad altre vocalizzazioni e relativi significati, ad esempio:

```
ktAb kut~Ab  N      kuttab (village school);Quran school
```

5.1.2.2 Le tabelle di combinazione

Tre tabelle stabiliscono le compatibilità possibili in una stessa parola della compresenza di prefissi e temi, temi e suffissi, ma anche prefissi e suffissi. Questo scopo è ottenuto in modo molto semplice attraverso il mero elenco degli appaiamenti binari consentiti. Ad esempio la presenza in una parola del prefisso pronominale tipico del tempo imperfetto (*muḍāriʿ*) è incompatibile con quella di suffissi pronominali tipici del perfetto (*māḍī*), oppure la presenza nel prefisso dell'articolo *al-* seleziona automaticamente le compatibilità solo con i suffissi esclusivi della flessione nominale, come è evidente dall'esempio successivo tratto dalla tabella di accoppiamento prefissi-suffissi:

```
NPref-Al Suff-0
NPref-Al NSuff-u
NPref-Al NSuff-a
NPref-Al NSuff-i
NPref-Al NSuff-An
NPref-Al NSuff-ayn
NPref-Al NSuff-|-ni
NPref-Al NSuff-wn
NPref-Al NSuff-iyna
```

```
NPref-Al NSuff-ap
NPref-Al NSuff-apu
NPref-Al NSuff-apa
NPref-Al NSuff-api
NPref-Al NSuff-tayn
NPref-Al NSuff-atAn
NPref-Al NSuff-At
NPref-Al NSuff-Atu
NPref-Al NSuff-Ati
NPref-Al NSuff-|t
NPref-Al NSuff-|tu
NPref-Al NSuff-|ti
```

5.1.2.3 La fase di elaborazione

L'analisi è compiuta da un programma specifico, compilato in Perl nel caso dell'implementazione originale di Buckwalter, e in Java nel caso del successivo progetto *AraMorph*[67]. Sulla base dell'input (da cui sono preventivamente eliminate le vocali brevi e gli altri diacritici in modo da avere la forma non vocalizzata della parola), il sistema procede attraverso una tecnica di ricerca a forza bruta (*brute-force search*)[68], che identifica tutte le possibili decomposizioni delle parole inserite in prefissi, temi e suffissi, cercando prefissi lunghi da 0 a 4 caratteri, temi di almeno un carattere e suffissi da 0 a 6 caratteri. Le decomposizioni prefisso-tema-suffisso ottenute vengono dapprima confrontate con i rispettivi dizionari in modo da essere escluse se qualche elemento è mancante; quindi le categorie grammaticali delle forme rimaste sono confrontate con le tabelle di compatibilità e scartate se nessuna combinazione è possibile. Come risultato, ogni parola dell'input originale viene marcata come (i) 'non riconosciuta' se nessuna analisi supera il con-

[67] Perl e Java sono due diversi linguaggi di programmazione ad alto livello, non agiscono cioè al livello più basso possibile di calcolo ma necessitano di essere interpretati da un altro programma che li traduce in linguaggio macchina o viceversa. Questo permette di avere una sintassi e delle regole di compilazione più comprensibili dal programmatore e orientate all'uso (Sebesta, 2006).

[68] La ricerca a forza bruta è una tecnica di risoluzione dei problemi che consiste nell'enumerazione sistematica di tutti i possibili candidati per la soluzione e nella conseguente verifica caso per caso se il candidato soddisfa le condizioni di verità. Il sistema è tanto semplice nella sua concezione quanto costoso in termini di risorse: più candidati ci sono, più il numero delle verifiche aumenta esponenzialmente (Bernstein, 2005).

trollo di compatibilità; (ii) 'univoca' se è possibile un'unica analisi morfologica; (iii) 'ambigua' se più analisi alternative sono possibili.

5.1.2.4 Osservazioni e criticità

Si è detto che in *AraMorph* il testo in input viene pretrattato riconducendolo alla forma non vocalizzata priva di diacritici, e solo successivamente elaborato. Le ragioni di questa procedura sono probabilmente da ricercarsi nella tipologia dei testi per i quali l'analizzatore è prevalentemente concepito, vale a dire testi giornalistici e altri testi non letterari in AMS. Inoltre il sistema è stato implementato utilizzando come base il corpus *Arabic Treebank LDC*, appunto costituito in prevalenza da estratti di testate giornalistiche online[69]. La specializzazione del sistema a trattare testi contemporanei si dimostra più debole nell'approccio a testi della tradizione letteraria araba, primi fra tutti i testi che presentano una vocalizzazione completa come le raccolte di *ḥadīṯ*, ma anche testi dove la vocalizzazione è parziale o rara ma comunque presente. Per ridurre l'ambiguità di interpretazione del testo arabo anche un solo diacritico può rivelarsi essenziale, soprattutto in testi prevalentemente non vocalizzati, dove il singolo diacritico è presente proprio con la finalità di offrire al lettore una lettura univoca di una parola potenzialmente ambigua. Riconducendo per procedura ogni parola al suo scheletro consonantico, *AraMorph* sceglie deliberatamente di ignorare tale supplemento di informazione, comunque utile per la disambiguazione, restituendo in certi casi un alto numero di analisi errate che possono essere evitate con la conservazione dei diacritici laddove presenti in origine.

La seconda debolezza di *AraMorph* è ancora legata alla tipologia di testi utilizzati per la compilazione dei dizionari interni di prefissi, temi e suffissi. Tali liste sono infatti popolate utilizzando il patrimonio lessicale del corpus LDC. I morfemi presenti sono quindi quelli reperibili esclusivamente in testi contemporanei, e molti elementi tipici dell'arabo letterario sono assenti. L'analisi da parte di *AraMorph* di un testo letterario o della tradizione classica sarà pertanto meno efficace, non potendo riconoscere alcune informazioni morfologiche essenziali per tali tipologie di testi.

[69] Cfr. titolo 1.3.3.

La scelta dell'arabo contemporaneo come lingua oggetto d'analisi è il motivo anche di un'ulteriore debolezza del sistema, vale a dire la mancanza di informazioni stilistiche e cronologiche nelle varie voci dei dizionari e delle tabelle di combinazione. In questo modo, molti morfemi e soprattutto temi che sono virtualmente esclusivi di testi in AMS, ad esempio un numero non trascurabile di nomi propri[70] traslitterati in caratteri arabi che non sono rilevanti non solo per testi classici ma nemmeno per moderni testi letterari. L'inclusione di queste espressioni nelle liste di morfemi rischia di creare un certo numero di falsi positivi nell'analisi morfologica qualora venga applicata appunto a testi non strettamente contemporanei.

5.1.3 RAM: Una versione modificata di *AraMorph*

Per risolvere almeno in parte i problemi di *AraMorph* relativi principalmente alla neutralizzazione della vocalizzazione e alla mancanza di una stratificazione lessicale e morfologica di tipo diacronico, il presente lavoro propone alcune variazioni all'algoritmo originale di *AraMorph* (AM) e la conseguente implementazione di una versione aggiornata e modificata (RAM).

5.1.3.1 Vocalizzazione attiva

La prima modifica apportata riguarda il meccanismo di identificazione dei morfemi: invece di neutralizzare le vocali brevi e i diacritici totalmente o parzialmente presenti nel testo prima di effettuare l'analisi, il nuovo procedimento li conserva e ne tiene conto in modo da poter ridurre il numero dei falsi positivi e aumentare l'efficienza di analisi.

Nella versione originale di AM, quando una parola è inserita in input viene spogliata del corredo diacritico/vocalico secondo la rappresentazione X*X*X* → XXX, dove * rappresenta zero o più occorrenze di vocali brevi, *tanwīn* o rafforzamenti consonantici (*tašdīd*) e X le consonanti o le vocali

[70] 'Nome proprio' qui è usato per l'inglese *named entity*, in assenza di un'espressione equivalente e riconosciuta in italiano (entità denominate?). Nelle teorie di estrazione dell' informazione (*information retrieval*) una *named entity* può riferirsi non solo a nomi di persone o toponimi, ma anche a espressioni temporali, di quantità e misura, di valore monetario, percentuale, etc. (Sag, Baldwin, Bond, Copestake, & Flickinger, 2002).

lunghe indicate nel livello obbligatorio di scrittura. Ad esempio la parola *kitāb* viene pre-processata come `ktAb` e quindi confrontata con le voci non vocalizzate della prima colonna delle informazioni contenute nei tre dizionari (cfr. 5.1.2.1). In un caso come questo, anche l'eventuale parola *kuttāb* viene trasformata nella medesima forma `ktAb`, creando un conflitto di ambiguità.

La modifica dell'algoritmo presenta un certo grado di complessità, risolto prevedendo un processo a tre stadi. Nel primo stadio, ogni carattere della parola in input viene marcato o come consonante o diacritico, dove quest'ultimo è qualsiasi carattere o combinazione di caratteri compreso tra due consonanti. Successivamente, la struttura consonantica viene confrontata con le voci non vocalizzate dei dizionari al fine di ottenere tutte le corrispondenze esatte[71]. Come ultima condizione, le vocali brevi originariamente presenti nella parola non devono contraddire la voce vocalizzata del dizionario, vale a dire che se una vocale manca ciò non è invalidante ai fini dell'analisi, ma se è presente essa deve coincidere con quella del dizionario. L'algoritmo modificato quindi confronta la parola in input sia con la prima colonna di dati del dizionario (forma non vocalizzata) sia con la seconda (forma vocalizzata). Il primo confronto deve essere sempre vero per tutti i caratteri, assumendo la forma di corrispondenza esatta. Il secondo confronto è invece di tipo incrementale, ogni vocale breve presente in input sarà cioè verificata con la relativa voce in modo però autonomo e indipendente dalla presenza/assenza di ulteriori vocali brevi in altre posizioni.

5.1.3.2 Arricchimento dei dizionari

Per arricchire di voci il dizionario dei temi e soprattutto per bilanciare la prevalenza lessicale di termini quasi esclusivamente contemporanei, si è scelto di utilizzare un dizionario preesistente, che fosse disponibile in forma digitalizzata e che fosse formalizzato nella strutturazione interna dei vari

[71] Sono state previste alcune specifiche regole di compatibilità, che tengono conto di alcune idiosincrasie nella grafia, ad esempio l'eventuale scrittura difettiva della *alif hamza,* che viene resa in scrittura talvolta solo con *alif*. AM risolve tali problemi in modo sensibilmente inefficiente, moltiplicando per ogni possibilità le voci di dizionario.

campi di lemma, in modo da permettere l'importazione automatica. È stato quindi scelto il dizionario Arabo-Inglese di Antony Salmonè, un'opera della fine del diciannovesimo secolo (Salmoné, 1889), riedito in forma digitale dal Perseus Digital Library Project e disponibile online[72] in formato XML conforme alle regole di digitalizzazione TEI[73]. La strutturazione formale ed esplicita delle voci del dizionario ha consentito un'efficiente segmentazione automatica e conseguente importazione nel dizionario RAM dei temi. Dei 48670 temi (*stem*) presenti nel Salmoné, 32115 erano già presenti nelle liste originali di AM, per cui l'importazione ha aggiunto 16555 nuovi temi. Un simile processo di espansione lessicale richiede solo il tempo necessario a scrivere i semplici programmi di importazione dati dal formato XML, in quanto la fase di elaborazione computazionale è immediata e automatica ed evita quasi completamente l'inserimento manuale e il controllo dei dati da parte dell'operatore umano.

Per quanto concerne i dizionari dei prefissi e suffissi e le tabelle di combinazione, essi sono stati controllati manualmente e integrati con alcune classi di informazioni morfologiche assenti, come l'intera categoria del modo imperativo del verbo[74]. Le classi sono state identificate attraverso un'analisi manuale dei *token* marcati in output come 'non riconosciuti' dall'analizzatore originale AM.

5.1.3.3 Selezione del lessico

Al fine di ridurre la presenza di lessico esclusivamente contemporaneo e in mancanza di indicazioni supplementari di ordine stilistico o cronologico a

[72] http://www.perseus.tufts.edu/hopper/text?doc=Perseus%3atext%3a2002.02.0005

[73] Il TEI (*Text Encoding Initiative*) è un protocollo creato dal *TEI Consortium* che stabilisce norme potenzialmente universali e univocamente formalizzate per l'armonizzazione della digitalizzazione di testi in formato elettronico, in modo da facilitarne la diffusione, l'archiviazione e l'interscambio tra sistemi non omologhi. Tali norme interessano sia l'organizzazione dei contenuti testuali sia l'annotazione con informazioni extra-testuali aggiuntive (TEI Consortium, 2009).

[74] Probabilmente da ascriversi a una scelta esplicita degli sviluppatori del programma originale, in quanto forma raramente presente in testi giornalistici. L'uso dell'imperativo è invece diffuso nella letteratura dei ḥadīṯ, che comprende abbondanza di contesti prescrittivi e performativi.

supporto delle voci di dizionario in AM, si è proceduto a eliminare dalla lista dei temi di RAM un certo numero di elementi sicuramente corrispondenti a *named entities* in particolare nomi di persona e toponimi da lingue europee arabizzati in trascrizione. La versione originale di AM, costruita avendo come base il linguaggio giornalistico, presenta tra i temi un gran numero di nomi trascritti in arabo di personaggi contemporanei anche relativamente celebri, come i nomi della giocatrice di tennis belga Sabine Appelmans o la squadra di calcio ceca Sigma Olomuc. In certi casi la confusione con altri temi arabi che presentano la stessa grafia è possibile, soprattutto tenendo conto del fatto che la grafia araba non prevede alcuna marcatura dei nomi propri, come l'iniziale di parola in carattere maiuscolo in lingue quali l'inglese o l'italiano. Come esempio di ambiguità di analisi, la trascrizione in arabo del nome 'John' *juwn* è omografa del termine arabo *jūn*, 'baia, insenatura' e indistinguibile in base a criteri puramente morfologici . Si è quindi proceduto a estrarre una lista di potenziali *named entities* selezionando dalle glosse in inglese del dizionario AM dei temi tutte le voci comincianti per lettera maiuscola, sfruttando un'osservazione dello stesso autore di AM che afferma che una glossa con iniziale maiuscola corrisponde a una *named entities* nel 99% dei casi (Buckwalter, 2002). La selezione è stata poi comparata con le voci del Salmoné, in modo da conservare nel nuovo dizionario le voci comunque lì presenti. La lista risultante è stata infine espunta dal dizionario dei temi RAM.

In questo modo un gran numero di nomi propri non arabi appartenenti al lessico contemporaneo sono stati esclusi, anche se in questo modo gli unici nomi propri di persona e di luogo presenti nel nuovo dizionario saranno quelli unicamente contenuti nel Salmonè. Dopo aver eliminato i temi non pertinenti, si è cercato di aggiungerne di nuovi più legati alla tipologia testuale classico-letteraria, sempre però con un procedimento automatico. I nomi di tutti i trasmettitori presenti negli *isnād* della raccolta di Al-Buḫārī, identificati attraverso la segmentazione di *HadExtractor* sono infine stati inseriti nel nuovo dizionario RAM dei temi.

5.2 Connettere testo e testi

5.2.1 Un testo nei testi: ricerca avanzata di stringhe

5.2.1.1 *CrossQuran*: la ricerca del Corano nei *ḥadīṯ*

Le tradizioni contengono un certo numero di citazioni di versetti coranici o parte di essi. A differenza dei *ḥadīṯ*, per il corpus coranico sono disponibili in letteratura l'annotazione morfologica e la rappresentazione sintattica (Dukes, Atwell, & Sharaf, 2010). Un processo d'identificazione delle parti del Corano contenute nei *ḥadīṯ* potrebbe quindi fornire un supplemento integrabile di informazione, in modo da evitare, in eventuali successive fasi di elaborazione morfologica e sintattica del matn, di compiere analisi già fatte per certi frammenti testuali. Uno strumento capace di identificare queste citazioni potrebbe arricchire il file in output di *HadExtractor* (contenente l'intero corpus segmentato) con una nuova categoria di informazioni indicante la presenza di passaggi coranici e il riferimento al testo d'origine.

Per verificare questa ipotesi, è stato progettato un piccolo programma, chiamato *CrossQuran*, il cui scopo è appunto di cercare nel testo dei *ḥadīṯ* (ma virtualmente in qualsiasi testo) tutte le occorrenze dei passaggi coranici. Il programma usa come input sia il *matn*[75] estratto con *HadExtractor* sia una versione digitalizzata del Corano (Dukes K. , 2010) che fornisce il testo lemmatizzato e segmentato in capitoli (sure), versi e parole, insieme con le relative annotazioni morfologiche e sintattiche.

Come unità minime discrete da ricercare nei *ḥadīṯ* sono stati scelti i versetti, ma poiché la fonte digitale è completamente lemmatizzata anche unità più piccole, come digrammi o n-grammi potrebbero essere ugualmente impiegate. Per il paradigma di riferimento con cui elaborare le strategie di ricerca ci si è nuovamente affidati alla sintassi delle espressioni regolari, già trattata nel capitolo 3.

Tutte le RE impiegate in quel contesto per operazioni di segmentazione ed estrazione erano caratterizzate da schemi fissi, scritti una volta e poi

[75] poiché *nell'isnād* per definizione non ci sono citazioni coraniche né altri elementi al di fuori dei nomi dei trasmettitori, delle EF e delle eulogie.

sempre uguali a se stessi al mutare del testo analizzato (cioè lo stesso schema era applicato a tutti i differenti *ḥadīṯ*). Tali schemi costituivano quindi una rappresentazione astratta e simbolica di un modello che si aspirava a trovare realizzato nel testo in varie realizzazioni, diverse ma omologhe per struttura.

Qui invece, il processo richiede di trovare corrispondenze per ogni verso coranico in ogni possibile *ḥadīṯ*, quindi lo schema non è più un modello astratto ma è una variabile che cambia in continuazione assumendo di volta in volta il valore di ciascun versetto. La soluzione al problema ha previsto l'inserimento nello schema della RE di una variabile che si riferisce agli elementi di una lista. Per ciascun elemento il sistema crea automaticamente una nuova RE da essere impiegata come schema da ricercare nel testo di arrivo. L'esempio seguente mostra la scrittura della RE a schema variabile usando il linguaggio di programmazione Python:

```
 pat_var = r'%s'
for pattern in patterns:
p = re.compile(pat_var % pattern)
```

Il significato di questa espressione è "per ciascun elemento della lista trova quell'elemento in..."

CrossQuran , che utilizza questa RE come strumento principale di ricerca, è stato applicato al corpus contenente i matn e ha restituito 132 occorrenze, vale a dire che ha trovato nei *ḥadīṯ* 132 versetti coranici citati un numero variabile di volte.

5.2.2 Un modello per la ricerca e la comparazione di testi in testi

CrossQuran è stato realizzato per rispondere a un'esigenza legata a un testo specifico come quello coranico, cercare cioè unità predefinite di lunghezza ridotta in un altro testo. Le stesse strategie di ricerca possono però essere impiegate per la realizzazione di un programma che compara due testi e valuta quali parti del primo testo sono contenute nel secondo e viceversa.

Supponiamo quindi di usare come schema di ricerca un testo nella sua interezza, e di cercarlo in un altro testo riducendo man mano la lunghezza

delle unità cercate fino a trovare una qualche corrispondenza. Questo richiede che la dimensione minima degli schemi variabili da cercare sia resa variabile. Nel caso di *CrossQuran*, la dimensione della stringa era fissa e corrispondeva esattamente alla lunghezza del versetto, che è tipicamente composto da una frase o un sintagma incluso tra due pause.

Per estendere quindi la ricerca da stringhe fisse a combinazioni variabili di sotto-stringhe, è stato ipotizzato il seguente flusso di elaborazione, nel quale il programma candidato:

- carica come input il testo A (che fornirà gli schemi da cercare) e il testo B (che sarà l'ambiente in cui cercare gli schemi);
- segmenta A in unità discrete delimitate da spazi bianchi[76];
- stabilisce (ad esempio su input dell'utente umano) la lunghezza minima e massima delle combinazioni di parole che possono far parte dello schema, ad esempio cinque come minimo e il numero di parole incluse tra due punti come massimo. Fissando la lunghezza minima a due è possibile ad esempio cercare tutte le collocazioni di un testo in un altro;
- costruisce la lista degli schemi, che includerà tutte le possibili combinazioni di parole consecutive, a partire dalla più lunga. Una routine calcolerà quindi la dimensione di ciascuna stringa tra due punti e popolerà una lista con tutte le possibili combinazioni tra il numero massimo e il numero minimo, in questo caso cinque. Assumendo ad esempio che una stringa contenga 13 parole, le combinazioni possibili saranno 44;
- compila una RE che considera i valori di lista come schemi da cercare;
- cerca corrispondenze in tutto il testo B, usando uno degli algoritmi di ricerca esatta di stringhe disponibili in letteratura (Aho, 1990), che possono essere generalmente di due tipi principali. Il primo prevede una ricerca decrementale dalla stringa più lunga

[76] naturalmente assumendo che ciò per l'arabo può comprendere parole isolate o in combinazione con elementi clitici prefissi o suffissi. a meno di non prevedere un pretrattamento di lemmatizzazione del testo.

alla più breve, ma richiede un certo numero di risorse nell'elaborazione. Il secondo utilizza un meccanismo di automazione a stati finiti basato sull'incremento progressivo della dimensione di stringa (Faro & Lecroq, 2013);

- ❖ restituisce le occorrenze trovate.

Un programma costruito su tale modello permetterebbe ad esempio di cercare quanta parte del testo di un *ḥadīṯ* è contenuta in una collezione di altri testi[77]. Il modello è attualmente in fase di sperimentazione e non è pertanto ancora stato valutato in termini di efficienza, ma è stato qui descritto come esempio di possibile estensione applicativa di uno strumento costruito invece per scopi specifici legati al testo dei *ḥadīṯ*.

5.2.3 Il problema dell'esatta corrispondenza

CrossQuran e il modello di programma per la ricerca di stringhe sono in grado di restituire solo la corrispondenza esatta con lo schema impiegato. Per estenderne ulteriormente l'impiego, si potrebbero definire alcune strategie specifiche che permettano una certa elasticità nella ricerca in testi in lingua araba, neutralizzando e rendendo irrilevanti per la corrispondenza alcuni elementi di testo e soprattutto tutti ii diacritici della vocalizzazione.

La strategia qui proposta mira ad affrontare attraverso trattamenti 'leggeri' di superficie la neutralizzazione dell'articolo (che viene prefisso cliticamente alla parola), senza fare ricorso a strumenti di analisi ' pesante' come i lemmatizzatori e gli analizzatori morfologici. Lo stesso procedimento potrebbe essere impiegato anche ad esempio per neutralizzare le coniugazioni dei verbi regolari.

Per quanto riguarda l'articolo *al-*, l'approccio è molto semplice e si ispira ad alcuni metodi di tokenizzazione impiegati da *AraMorph* e dalle tecniche non linguistiche per l'estrazione di 'informazione. Consiste nell'eliminare temporaneamente e solo per la fase di ricerca di corrispondenza tutte le sequenze '*al*' in inizio di parola, sia nel testo che fornisce gli schemi sia in

[77] Virtualmente un algoritmo di questo tipo potrebbe usare come input l'intero testo dei *ḥadīṯ*, ma in questo caso la ricerca necessiterebbe di algoritmi di diversa impostazione (Faro & Lecroq, 2013).

quello di arrivo. In questo modo, le due stringhe di testo seguenti, simili ma non identiche, sarebbero considerate corrispondenti. Ad esempio si ipotizzi di voler cercare corrispondenze per la stringa *al-'amal al 'ilmiyy* ('il lavoro scientifico'). Innanzitutto nei due testi (qui mostrato in traslitterazione Buckwalter modificata) vengono eliminati le vocali brevi e i segni di rafforzamento consonantico eventualmente presenti:

```
Alcamal Alcil_miy+    →    Alcml Alcl_my+
```

Poi sono effettuate l'operazione di neutralizzazione (indicata nell'esempio con `%`), ricerca e ripristino del testo originario:

	Testo A (schema da cercare)	Testo B (ambiente di ricerca)
forma originaria	`Alcml Alcl_my+`	`cml cl_my+`
forma con neutralizzazione	`%cml %cl_my+`	`cml cl_my+`
ricerca corrispondenza	`cml cl_my+`	
ripristino forma originaria	`Alcml Alcl_my+`	`Alcml Alcl_my+`

Ovviamente in questo modo la sequenza '*al*' è ignorata anche per quelle parole dove essa non ha valore di articolo ma appartiene al tema di parola, ad esempio *'alam* ('dolore'), ma se questo sarebbe ambiguo per una strategia lessicale, non lo è per un approccio di superficie che ha come scopo la ricerca di corrispondenze e non di significati linguistici. In questo caso però, bisognerà inserire una regola di neutralizzazione anche per le sequenze `AlAl` e `AlEl`[78], in modo da trovare comunque corrispondenze anche per queste parole. Si veda l'esempio con *al-'alam al 'ilmiyy* ('la pena scientifica'):

	Testo A (schema da cercare)	Testo B (ambiente di ricerca)
forma originaria (non vocalizzata)	`AlAlm Alcl_my+`	`Alm cl_my+`
forma con neutralizzazione	`%m Alcl_my+`	`%m cl_my+`
ricerca corrispondenza	`%m Alcl_my+`	

[78] Che rendono conto dell'eventuale presenza o assenza dell'*hamza* sulla seconda *alif.*

ripristino forma originaria	`AlAlm Alcl_my+`	`Alm cl_my+`

Una simile strategia sembra non considerare quelle parole dove l'articolo è preceduto da altri clitici, come ad esempio *wa-l-'alam* ('e il dolore'). In questo caso basta prevedere però regole di neutralizzazione per i tipici clitici di prefissazione e le loro combinazioni (ad es. *bi-*, *bi-l-*, *li-l-*, *wa-*).

5.2.4 La costruzione di una memoria di traduzione per i *ḥadīṯ*

5.2.4.1 *Ḥadīṯ* e traduzioni

La traduzione in altre lingue del testo dei *ḥadīṯ*, fondamentale dal punto di vista giuridico e religioso islamico è un tema complesso che coinvolge piani linguistici e stilistici e ha più a che fare con le scienze della traduzione che non con gli approcci computazionali. Saranno pertanto tralasciate le considerazioni circa la qualità del contenuto e i processi coinvolti dalla traduzione, e si rimanda per questo ad alcune opere scientificamente significative che si sono occupate della traduzione di parti del corpus della Tradizione (Juynboll, 2007; Vacca, Noja, & Vallaro, 2009).

In questa sede sarà invece affrontato il problema dell'abbinamento automatico delle traduzioni già disponibili con il testo originale. In chiave di linguistica computazionale ciò si ricollega al quadro teorico-pratico della traduzione automatica e delle tecniche di allineamento applicato al testo arabo (Amna, 2011).

5.2.4.2 Memorie di traduzione e allineamento

Delle memorie di traduzione (TM) si è già accennato al titolo 1.3.1. Esse sono dei corpora paralleli di testi bilingue o multilingue in cui è possibile consultare alcune sotto-unità testuali (generalmente periodi e frasi) all'interno del loro contesto e paragonarne le versioni di traduzione in lingue diverse (Chuang, Jian, Chang, & Chang, 2005). La natura dell'organizzazione dei dati tipica delle TM non prevede una direzione specifica originale-traduzione, per cui le due versioni sono trattate strutturalmente come ugualmente valide e paritarie. Il motivo principale per cui, disponendo di un testo e della sua traduzione ci sia bisogno di strategie computazionali per l'allineamento,

è evidentemente che le due versioni per qualche ragione non siano già appaiate tramite un indice comune di numerazione interno ai testi stessi (Ma, He, Way, & van Genabith, 2011).

5.2.4.3 Un modello per la costruzione di una memoria di traduzione

Le TM disponibili in letteratura per l'arabo sono in genere estensioni di TM esistenti o collezioni specifiche la cui variante linguistica utilizzata è prevalentemente l'AMS (Meedan, 2009). Il modello qui proposto si propone di integrare il testo arabo segmentato dei *ḥadīṯ* con una traduzione inglese esistente e di allinearlo a livello di *ḥadīṯ*, in modo che per ogni tradizione siano disponibili affiancate entrambe le versioni.

5.2.4.4 Segmentazione ed estrazione di una traduzione inglese

Si è quindi cercata una traduzione digitalizzata del *Al-ǧāmiʿ al-ṣaḥīḥ* di al-Buḫārī, che soddisfacesse gli stessi criteri usati per la selezione dell'edizione in arabo (cfr. 3.3.1), vale a dire completa digitalizzazione e disponibilità pubblica on-line. Anche in questo caso il valore qualitativo della traduzione è relativamente importante, poiché lo scopo principale è verificare l'agibilità del modello e non l'analisi filologica del testo. Il testo selezionato è la versione digitale della traduzione inglese di Khan (Sahih Al Bukhari, 1984).

In questa traduzione, *l'isnād* non compare nel testo, e il matn è preceduto solo dal primo trasmettitore riconosciuto. Ogni singolo *ḥadīṯ* si presenta quindi nella forma dell'esempio seguente:

> Volume 1, Book 1, Number 1:
>
> Narrated 'Umar bin Al-Khattab:
>
> I heard Allah's Apostle saying, "The reward of deeds depends upon the intentions and every person will get the reward according to what he has intended. So whoever emigrated for worldly benefits or for a woman to marry, his emigration was for what he emigrated for."

In questo caso, la struttura testuale è ancora più semplice dell'originale arabo, in quanto l'inizio del *matn* è chiaramente indicato. Per la segmentazione e l'estrazione delle informazioni relative alla numerazione è stato

quindi applicato al testo il programma *HadExtractor*, modificato per poter leggere la nuova struttura del testo. Poiché *HadExtractor* è uno strumento di analisi strutturale e non linguistica, il fatto che il testo sia in inglese e in caratteri latini e non in arabo richiede minime modifiche in termini di RE impiegate e codifica dei caratteri ma non influisce quasi per nulla sugli algoritmi di elaborazione.

L'output del programma produce un file XML in parte analogo a quello dell' originale arabo, secondo il seguente esempio:

```
<hadith_en id_en="1" id_cor="">
    <source_info>
    <vol>1</vol>
    <book>1</book>
    <num>1</num>
    </source_info>
    <last_trasm>'Umar bin Al-Khattab</last_trasm>
    <mat_en>I heard Allah's Apostle saying, "The reward of deeds
depends upon the intentions and every person will get the reward
according to what he has intended. So whoever emigrated for
worldly benefits or for a woman to marry, his emigration was for
what he emigrated for."</mat_en>
```

5.2.4.5 Il problema dell'allineamento

Poiché i due file XML contenenti i *ḥadīṯ* in arabo e i matn in inglese sono strutturati in modo omogeneo e dispongono di informazioni sulla numerazione dei *ḥadīṯ*, la questione dell'allineamento appare semplice nella soluzione. In realtà i due testi mostrano diversità importanti nella segmentazione e numerazione interna dei singoli *ḥadīṯ*, Il testo arabo comprende 7306 *ḥadīṯ*, mentre la versione inglese ne raccoglie 6916. Entrambe le versioni hanno un sistema numerico di indicizzazione interna, ma mentre esso corrisponde per quanto riguarda le indicazioni di volume (su un totale di 9 volumi) e libro (su un totale di 93 libri), discorda invece in modo piuttosto consistente per la numerazione delle singole tradizioni. Questa differenza è probabilmente dovuta all'utilizzo, in sede di traduzione e digitalizzazione, di due edizioni diverse dell'opera di al-Būḫārī i cui curatori avevano utilizzato criteri differenti per la numerazione. Ad esempio due o più *ḥadīṯ* che hanno numerazioni diverse in un'edizione, nell'altra riportano il medesimo nume-

ro. Inoltre, poiché l'ordine generale in cui i *ḥadīṯ* appaiono non era fissato all'origine da criteri rigorosi ma da indicizzazioni a soggetti, molti *ḥadīṯ* compaiono in posizioni e ordinamenti diversi da un'edizione all'altra.

Per un corretto allineamento, la strategia non computazionale più evidente è quella di associare manualmente ciascun elemento testuale con il corrispettivo nell'altra lingua, anche se essa mostra due principali debolezze, il consumo di risorse, soprattutto nel caso di corpora paralleli molto estesi come questo, la possibilità di errori umani, che sono difficilmente predicibili.

5.2.4.6 Una strategia statistico-lessicale

E' stata identificata una strategia di allineamento in diverse fasi, che cerca di sfruttare tutte le informazioni esplicite e sopperire alle ambiguità attraverso un metodo puramente di calcolo statistico e uno di tipo lessicale:

- pretrattamento: i due file XML arabo e XML inglese vengono utilizzati come input per il programma di allineamento.
- fase I: sfruttamento delle informazioni numeriche. Le informazioni numeriche presenti nei *tag* XML `<vol></vol>` e `<book></book>` servono a identificare 93 insiemi di *ḥadīṯ* (tanti quanto i libri), all'interno dei quali, attraverso una semplice inferenza numerica, è possibile far corrispondere con esattezza il primo e l'ultimo numero in ciascuna delle due versioni. Ad esempio i due insiemi arabo e inglese corrispondenti al libro 9, avranno sicuramente corrispondenza reciproca almeno tra il primo e l'ultimo *ḥadīṯ* della serie. Questo permette quindi di limitare le tecniche di allineamento a serie più piccole di *ḥadīṯ* (circa 80 in media per ogni libro) e non a tutta la collezione (circa 7000 occorrenze).
- fase II: comparazione statistica. Le stringhe di testo nelle due versioni (quindi in media 80 stringhe per l'arabo e 80 per l'inglese) vengono misurate in lunghezza. I valori di misurazione sono poi posti tra loro in relazione proporzionale, e i relativi *ḥadīṯ* sono allineati insieme secondo il modello più prossimo alla

riproduzione esatta del valore dei rapporti proporzionali. Questo secondo il principio che due stringhe in lingue diverse possono differire tra loro per lunghezza, ma se due stringhe differiscono nella stessa lingua il rapporto tra le loro lunghezze è costante anche per le relative traduzioni in altre lingue (Och, Tillmann, & Ney, 1999).

- ❖ fase III: il metodo statistico è verificato attraverso una parziale indicizzazione lessicale (Řehůřek & Sojka, 2010) del testo. Il *matn* è segmentato in parole (qui sequenza di caratteri separata da due spazi) che vengono sottoposte a conteggio di frequenza. Gli elementi lunghi almeno 5 caratteri e che ricorrono un'unica volta nel testo (*hapax*) sono poi sottoposti ad analisi lessicale attraverso l'analizzatore morfologico RAM, al solo fine di estrarne il significato in inglese. A questo punto la lista degli *hapax* in arabo (con relativa traduzione inglese) viene confrontata con le rispettive liste di *hapax* della versione inglese e le possibili corrispondenze sono usate per la selezione progressiva dei candidati all'allineamento.

Al termine del processo di allineamento, il modello produce un nuovo file XML in cui i *matn* dell'originale arabo sono allineati alla corretta versione in inglese, e che costituisce nella sua interezza la memoria di traduzione vera e propria. L'esempio successivo mostra la prima voce della TM, relativa al primo *ḥadīṯ* della collezione:

```
<hadith_ar_en id_cor="1">
<source_info>
<vol>1</vol>
<book>1</book>
<num>1</num>
</source_info>
<last_trasm_ar>عُمَرَ بْنَ الْخَطَّابِ</last_trasm_ar>
<last_trasm_en>'Umar bin Al-Khattab</last_trasm_en>
<mat_ar> سَمِعْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ إِنَّمَا الْأَعْمَالُ بِالنِّيَّاتِ وَإِنَّمَا لِكُلِّ امْرِئٍ مَا نَوَى فَمَنْ كَانَتْ
هِجْرَتُهُ إِلَى دُنْيَا يُصِيبُهَا أَوْ إِلَى امْرَأَةٍ يَنْكِحُهَا فَهِجْرَتُهُ إِلَى مَا هَاجَرَ إِلَيْهِ . </mat_ar>
<mat_en>I heard Allah's Apostle saying, "The reward of deeds depends upon the intentions and every person will get the reward according to what he has intended. So whoever emigrated for worldly benefits or for a woman to marry, his emigration was for what he emigrated for."</mat_en> </hadith_ar_en>
```

Parte III:
CONCLUSIONI

6. Risultati e conclusioni

6.1 Risultati

In sintesi, questa tesi è un tentativo di avvicinarsi al testo dei *ḥadīṯ* in chiave linguistica e computazionale, utilizzando una serie di tecniche di analisi e strumenti applicativi che affrontano i seguenti compiti:

- segmentazione: un programma interpreta la struttura testuale dei *ḥadīṯ* e li decompone automaticamente nelle loro unità fondamentali di informazione, l'*isnād* (contenente le modalità di trasmissione e recepimento dello stesso *ḥadīṯ* dal primo testimone al compilatore della collezione) e il *matn* (il contenuto stesso della tradizione).
- estrazione di informazione: il programma, sulla base dell'interpretazione strutturale e delle indicazioni già presenti in una parte del testo (*isnād*), estrae automaticamente i nomi di tutti i trasmettitori corredandoli con informazioni meta-testuali circa la tipologia e la composizione delle catene di trasmissione;
- rappresentazione di informazione: le informazioni estratte, in particolare i nomi dei trasmettitori, le tipologie e la direzione di trasmissione vengono rappresentate secondo alcuni parametri statistici; alcuni grafi generati automaticamente mostrano poi un modello di rappresentazione bidimensionale dinamica delle catene di trasmissione;
- annotazione morfologica: il testo del *matn* è sottoposto ad analisi morfologica automatica attraverso una versione appositamente perfezionata di un lemmatizzatore esistente; viene quindi segmentato e automaticamente annotato in tutti i suoi costituenti morfologici;

- ❖ comparazione con altri testi: un programma ricerca all'interno del *matn* tutti gli elementi corrispondenti o simili a quelli di un altro testo, in questo caso il Corano;
- ❖ allineamento bilingue: il testo arabo viene automaticamente allineato nelle sue parti a una traduzione inglese preesistente in modo da costruire il nucleo di una memoria di traduzione.

Queste operazioni possono essere interpretate come singoli elementi di un processo di elaborazione modulare che, pur conservando integra la lezione del testo originale dei *ḥadīṯ*, vi applica trasformazioni progressive, fino a ottenere una base di dati relazionale in cui il testo originario è accompagnato da un corredo aggiuntivo di analisi, annotazioni, informazioni e loro rappresentazioni (figura 6.1).

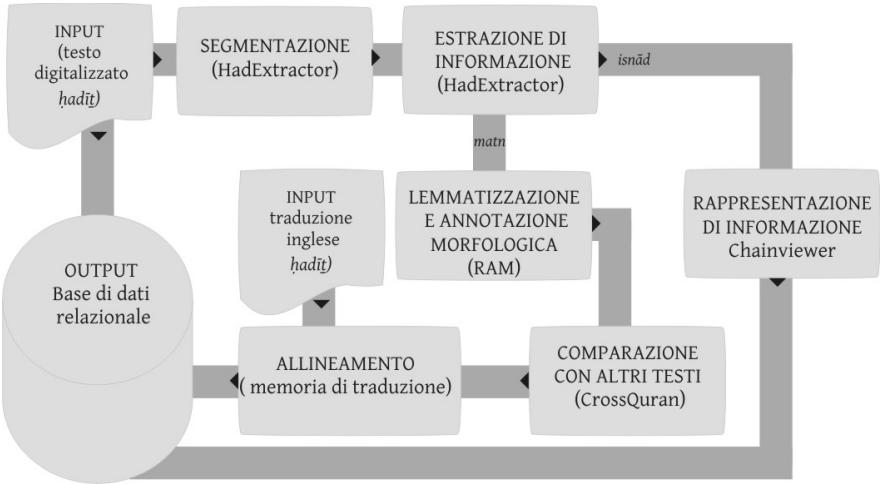


Figura 6.1 Il modello generale per l'analisi dei *ḥadīṯ* in prospettiva modulare

6.2 Valutazione quantitativa

6.2.2 Criteri e parametri per la valutazione

La costruzione di strumenti per l'analisi del testo molto diversi tra loro per tipologia e quadro di riferimento, richiede approcci diversi per una valutazione oggettiva dei risultati.

6.2.2.1 La valutazione caso per caso

Per valutare quanto un programma effettui realmente e correttamente l'elaborazione di un corpus di testi, la soluzione più ovvia e immediata consiste nel verificare manualmente uno dopo l'altro i risultati ottenuti. Questo risulta possibile quando i dati in input e output sono limitati quantitativamente, ad esempio dopo aver inserito nel sistema una frase da analizzare morfologicamente, l'output può essere direttamente verificato nella sua correttezza. Questo procedimento però, oltre a essere dipendente e limitato dalla quantità di output da verificare, è fortemente non predittivo, in quanto certifica la correttezza dei risultati strettamente in funzione del particolare contesto di input, ma non predice la possibile efficienza del sistema a prescindere dal tipo di testo inserito.

La valutazione caso per caso può invece essere efficace nella verifica di modelli basati sulla conoscenza linguistica in grado di analizzare o generare nuovo testo i quali, almeno negli stadi inziali di elaborazione, hanno strutture di regole e comportamenti molto semplici e soprattutto ben conosciuti dal programmatore che li ha compilati, e necessitano quindi di verifiche basate su un numero relativamente basso di input possibili. Questo è il caso ad esempio delle grammatiche di prova (*toy-grammars*), dove la base lessicale e il sistema di regole sono per definizione assai limitati, e quindi permettono l'agevole verifica manuale di ogni risultato (van Rijsbergen, 1979).

7.2.2.2 Procedimenti automatici a supervisione parziale

Nel caso invece di grandi quantità di output dell'ordine di migliaia di risultati e basate su un'elaborazione di input e basi lessicali complessi, è necessario prevedere un sistema di valutazione articolato e in parte basato

sull'automazione e la predizione (Guessoum & Zantout, 2001). Con riferimento ai correnti metodi di valutazione quantitativa di analisi computazionali di tipo empirico-statistico, la procedura standard prevede di suddividere l'insieme dei risultati ottenuti in due corpora, uno di sperimentazione (*testing*), l'altro di apprendimento (*training*). La verifica manuale sarà compiuta esclusivamente sul corpus di sperimentazione, la cui dimensione è generalmente contenuta e può variare tra il 5% e il 50% dell'intero insieme di risultati. Questo metodo consente l'applicazione di procedure di valutazione ricorsiva dell'efficienza del sistema: il programma infatti viene man mano perfezionato nelle sue parti in modo progressivo, ogni volta cercando di migliorare l'efficienza generale sulla base del miglioramento della qualità dei risultati ottenuti nel corpus di sperimentazione. Il corpus di apprendimento è infatti denominato così proprio perché in modo metaforico 'apprende' gradualmente dal corpus di sperimentazione a fornire risultati più efficaci. (Olson & Delen, 2008).

6.2.2.3 Classificazione degli errori

Nel contesto delle teorie di estrazione dell'informazione (*information retrieval*), i risultati in output vengono classificati in tre categorie, che esplicitano il rapporto tra un dato vero o falso di per sé, o considerato tale dal processo di elaborazione (Baeza-Yates & Ribeiro-Neto, 1999).

In questo quadro, un 'vero positivo' (*true positive*) è un risultato correttamente identificato come appartenente alla classe positiva, quindi vero di per sé e vero per la macchina.

Un falso positivo (*false positive*) è un risultato erroneamente identificato come appartenente alla classe positiva, quindi è falso di per sé ma vero per la macchina.

Un falso negativo (*false negative*) è infine un risultato erroneamente identificato come non appartenente alla classe positiva, quindi vero di per sé ma falso per la macchina.

Il quarto possibile caso, vale a dire il 'vero negativo', falso di per sé e falso per la macchina è usato raramente nei sistemi di valutazione ordinari .

Immaginiamo a titolo di esempio che si debba verificare l'efficienza di un programma che identifichi tra una lista di parole i nomi propri. Il programma processerà dapprima una lista di parole in input e produrrà dei risultati:

Lista di input: (1) muḥammad; (2) kitāb; (3) yaḥya; (4) zaynab; (5) madrasa.

Output: (2) kitāb; (3) yaḥya; (4) zaynab

Dall'analisi dei risultati, (3) e (4) sono veri positivi in quanto effettivamente nomi propri e riconosciuti come tali dal sistema; (2) è un falso positivo, in quanto riconosciuto nome proprio dal sistema ma senza esserlo; (1) è invece un falso negativo, in quanto nome proprio ma non riconosciuto tale dal sistema; (5) infine, è un vero negativo.

6.2.2.4 Efficienza, precisione e sensibilità

L'efficienza di un sistema di estrazione dati consiste nel numero di analisi realmente effettuate dal programma, indipendentemente dai risultati ottenuti. Le analisi non compiute rispecchiano quindi errori di tipo non interpretativo ma strutturale, come nel caso di un'analisi rifiutata a priori dal programma perché il testo in input non soddisfa nemmeno i prerequisiti di struttura necessari al suo trattamento, ad esempio un input esclusivamente numerico per un programma che richiede espressamente caratteri alfabetici (Makhoul, Kubala, Schwartz, & Weischedel, 1999).

Nel campo dell'estrazione di informazione, il parametro della precisione (*precision*) serve a misurare il numero di risultati ottenuti che sono rilevanti per la ricerca, ed è espresso dalla seguente equazione:

$$Precision = \frac{tp}{tp + fp}$$

dove *tp* indica i veri positivi e *fp* i falsi positivi.

La sensibilità (*recall*) riguarda invece la misurazione della capacità del sistema di avvicinarsi alla condizione ideale di trovare proprio 'tutti' i risultati che sono rilevanti. La sua equazione è la seguente:

$$Recall = \frac{tp}{tp + fn}$$

dove *tp* indica i veri positivi e *fn* i falsi negativi.

In termini probabilistici e non matematici, la precisione può anche essere considerata come la probabilità che un risultato trovato sia effettivamente rilevante, mentre la sensibilità è la probabilità che un risultato rilevante sia stato effettivamente trovato.

La precisione e la sensibilità di un sistema vengono poi misurate congiuntamente e integrate in un ulteriore parametro, chiamato *F-measure*, che ne rappresenta la media armonica (Powers D. M., 2007):

$$F = 2 * \frac{precision * recall}{precision + recall}$$

6.2.3 La valutazione dell'efficacia di *HadExtractor* e RAM, e allineamento per laTM

6.2.3.1 Segmentazione dei *ḥadīṯ* (*HadExtractor*)

Il programma *HadExtractor* è stato applicato all'edizione digitalizzata del *Ṣaḥīḥ* di al-Buḫārī, composta di 7305 *ḥadīṯ* distinti in modo esplicito. (Al-Buḫārī, Ṣaḥīḥ al-Bukhārī). Dei 7305 *ḥadīṯ* trattati il processo di segmentazione ha avuto esito positivo per 7135 di essi, mostrando una percentuale di efficienza del 97.7% (cfr. Tabella 7.1). Quindi per 170 *ḥadīṯ* il sistema non è stato in grado di proporre segmentazioni possibili, restituendo un errore. Questo è avvenuto nella maggior parte dei casi perché le espressioni regolari per la segmentazione dell'*isnād* dal *matn* non hanno identificato nessuna espressione funzionale tra quelle incluse nel sistema. Per ovviare alla mancata segmentazione di tali *ḥadīṯ* e con la prospettiva della costruzione di un database completo che li contenga nella loro integrità, si è proceduto con la segmentazione manuale dei testi non processati.

L'output contenente i *ḥadīṯ* segmentati è stato quindi diviso in un corpus di sperimentazione (contenente circa 1000 occorrenze) e uno di training (6135 occorrenze). Il corpus di sperimentazione è stato sottoposto alla verifica manuale. Tenendo conto che la segmentazione perfetta e completa del corpus di sperimentazione avrebbe dovuto produrre 6480 stringhe di testo (comprendenti ciascuna il nome di un singolo trasmettitore oppure l'intero

matn), il programma, dopo la verifica manuale, ha invece restituito 5980 veri positivi, 440 falsi negativi e 60 falsi positivi. L'esigua percentuale di falsi positivi è indizio di un'alta precisione del sistema, sicuramente maggiore rispetto alla sua sensibilità a identificare tutti i valori positivi. I falsi negativi riscontrati sono in massima parte dovuti a errate sottosegmentazioni nelle catene dei trasmettitori (due nomi in un unico campo) oppure a *matn* che comprendono erroneamente il nome degli ultimi trasmettitori.

SEGMENTAZIONE		TEST SUI DATI SEGMENTATI	
effettuata	97.7%	Percentuale di errore (fp+fn)	7.7%
non effettuata	2.3%	Precisione (*precision*)	99.2%
		Sensibilità (*recall*)	93.1%
		F measure	96.1%

Tabella 7.1: Riepilogo dei risultati relativi alla segmentazione con *HadExtractor*

Una percentuale di errore del 7.7% è considerevolmente bassa rispetto alla media dei programmi di segmentazione automatica (Abu El-Khair, 2007, p. 33-35) ed è da considerarsi un risultato molto positivo, soprattutto in merito a una delle finalità dichiarate per questo sistema, vale a dire la possibilità di automatizzare il processo di trattamento di grandi quantità di dati testuali. In relazione invece all'altra finalità dichiarata, quella di ottenere una base di dati completa e pienamente segmentata di tutti i *ḥadīṯ* di una particolare collezione, in questo caso quella di Al-Buḫārī, l'esistenza di anche un solo risultato errato dev'essere corretta manualmente per non pregiudicare l'integrità del sistema. Considerando quindi che il corpus di sperimentazione contiene il 14,01% del numero totale di *ḥadīṯ* considerati (il corpus di apprendimento ne conterrà pertanto l'85,99%), il numero di errori da correggere manualmente è prevedibile nell'ordine di qualche migliaia. La correzione manuale è stata in questo contesto facilitata dall'ideazione di tecniche automatiche *ad hoc* di ordinamento e tipizzazione dei dati. È stata creata una lista che abbina i tutti i campi dei nomi dei trasmettitori (comprendente quindi anche gli errori) con la relativa posizione nella collezione. I campi uguali, che contengono cioè lo stesso nome, sono stati ridotti a uno,

preservando però i riferimenti a tutte le posizioni originali nel testo. Da un punto di vista probabilistico, i falsi negativi (che come si è visto costituiscono la maggioranza degli errori riscontrati) coincideranno con i campi più lunghi per estensione, poiché conterranno due o più nomi di trasmettitori al posto di uno solo. Inoltre è molto probabile che contengano comunque almeno una delle espressioni funzionali conosciute e considerate dal sistema. In questo modo, per trovare la maggior parte dei falsi negativi in modo quasi automatico è stato sufficiente combinare le due condizioni in uno specifico *script* di ricerca basato sulle espressioni regolari. In questo modo sono stati risolti circa 2500 dei 3000 errori predetti. Per i restanti 500 si è provveduto alla correzione umana tramite lettura del testo.

6.2.3.2 Analisi morfologica (RAM)

L'analisi del testo con la versione modificata di *AraMorph* (RAM) è stata compiuta sul corpus contenente solamente le parti dei *ḥadīṯ* segmentate ed etichettate come *matn*. L'ammontare totale delle parole separate da spazi (*token*) è di 382.700, un corpus piuttosto esiguo comparato ad esempio ai diversi milioni di parole di strumenti come lo *LDC Arabic Corpus*, ma interessante dal punto di vista della specializzazione e della tipologia di lingua utilizzata. Il corpus è quindi stato sottoposto all'elaborazione sia da parte della versione originale di *AraMorph* (AM), sia da parte del RAM nei suoi tre principali stadi di revisione (vocalizzazione, arricchimento e selezione lessicale). Ogni stadio successivo include anche i precedenti, quindi la versione RAM3 comprende tutti e tre i moduli aggiuntivi. Come descritto al titolo 5.1.2.3, per quanto riguarda la valutazione dell'efficacia, il sistema classifica i risultati come non analizzabili, univoci o ambigui. I parametri di percentuale di errore, precisione, sensibilità e *F measure* sono stati invece utilizzati per valutare in dettaglio il corpus di sperimentazione, costituito dal 15% dell'intero testo in input (57405 parole).

RICONOSCIMENTO				
	AM originale	RAM1 con modulo vocalizzazione	RAM2 con lessico aggiunto	RAM3 con lessico selezionato
Non analizzato	10.36%	12.55%	7.23%	8.12%
Univoco	29.45%	58.98%	62.52%	67.79%
Ambiguo	60.19%	28.47%	30.25%	24.09%
Valutazione del corpus di sperimentazione				
Percentuale di errore	60.54%	32.77%	27.65%	24.58%
Precisione	64.90%	74.57%	81.47%	83.37%
Sensibilità (*recall*)	74.56%	92.66%	90.88%	92.05%
F measure	69.40%	82.64%	85.92%	87.50%

Tabella 7.2: Riepilogo dei risultati relativi al trattamento dei *matn* con AM e RAM

Dall'analisi dei dati della tabella 7.2 è possibile ricavare alcune considerazioni. La percentuale di risultati (*token*) non analizzati varia tra il 7 e il 12% circa, e ha una variazione minima tra AM e RAM. È interessante notare come AM compia più analisi di RAM1, ma questo avviene al prezzo di un più alto numero di falsi positivi, che invece RAM1 discrimina attraverso la vocalizzazione. La presenza di risultati univoci in RAM1 è quasi il doppio rispetto ad AM, per poi aumentare ancora con i moduli lessicali RAM2 e RAM3. In proporzione, anche la percentuale di risultati ambigui o con più significati viene dimezzata con l'uso di RAM, e questo è principalmente dovuto proprio al valore discriminante e disambiguante che i diacritici hanno in RAM. Queste percentuali di miglioramento piuttosto rilevanti sono però da ascriversi principalmente al fatto che il testo analizzato, i *ḥadīṯ* in questo caso, presentano una vocalizzazione pressoché integrale, che contribuisce in modo importante alla disambiguazione.

Passando infine all'analisi del corpus di sperimentazione, l'aumento quasi ovunque progressivo dei valori di precisione, sensibilità e *F Measure* man mano che si passa da AM a RAM3, è indicativo della reale efficienza delle

modifiche apportate al programma originale. Inoltre il tasso di aumento percentuale pressoché identico tra precisione e sensibilità tra AM e RAM3 (circa 18%) conferma che i moduli lessicali e di vocalizzazione riescono a ridurre in modo uniforme sia i falsi positivi che i falsi negativi.

6.2.3.3 Allineamento arabo-inglese per la memoria di traduzione

I risultati ottenuti dal modello di allineamento statistico-lessicale presentato al titolo 5.2.4 sono stati sottoposti a valutazione quantitativa. I parametri di percentuale di errore, precisione, sensibilità e *F measure* sono stati utilizzati per valutare il corpus di sperimentazione, costituito da 700 *matn*, circa il 10% del numero totale della segmentazione originaria in arabo.

ALLINEAMENTO TM	
Percentuale di errore (fp+fn)	9.8%
Precisione (*precision*)	96.33%
Sensibilità (*recall*)	93.34%
F measure	94.81%

Tabella 7.3: Riepilogo dei risultati relativi all'allineamento perla memoria di traduzione

I risultati mostrano un'efficienza del modello relativamente buona, considerando il numero piuttosto alto di falsi negativi prodotti.

6.3 Una valutazione qualitativa

La tabella seguente mostra una valutazione riassuntiva delle strategie impiegate secondo un'interpretazione il corredo di parametri introdotto al titolo 3.1.

	AUT	PRO	CON	DIP
Segmentazione ed estrazione di informazione (*HadExtractor*)	+	-	-	±
Rappresentazione di informazione (catene di trasmissione)	±	-	-	+
Annotazione morfologica (*RAM*)	±	±	±	-
Ricerca di stringhe (*CrossQuran*)	+	-	-	-
Allineamento bilingue (memoria di traduzione)	±	-	±	-

Tabella 7.4: valutazione qualitativa dei risultati computazionali. LEGENDA: AUT: grado di automaticità/non supervisione; PRO: coinvolgimento delle componenti profonde del testo; CON: coinvolgimento di conoscenza (vs statistica); DIP: grado di dipendenza dal contesto.

Si possono quindi tracciare alcune brevi conclusioni sulla qualità e l'originalità degli strumenti computazionali presentati in questa tesi.

6.3.1 Un modello a ridotta supervisione

Una delle ipotesi principali di questo lavoro era di provvedere, per l'analisi dei *ḥadīṯ*, una serie di metodi di elaborazione computazionale che riducessero al minimo la necessità dell'intervento umano.

Tra gli strumenti prodotti *HadExtractor* è sicuramente il modello più riuscito di approccio quasi totalmente automatico, essendo in grado di compiere in qualche secondo di elaborazione un enorme numero di operazioni (circa 8000 segmentazioni di *ḥadīṯ*, 40.000 estrazioni di trasmettitori, 9.000 normalizzazioni) con un margine di errore piuttosto basso (cfr. tabella 7.1). *HadExtractor* può costituire quindi un esempio importante nell'ambito della definizione dei sistemi di annotazione non supervisionata.

Il modello complessivo di analisi ricorre comunque in modo significativo a processi non supervisionati, particolarmente nel caso della segmentazione, estrazione di informazioni, rappresentazione bidimensionale delle catene di trasmissione, allineamento di testo originale e traduzioni. L'intervento

umano quindi, seppur quasi assente durante la fase di elaborazione, si rivela ancora essenziale per la verifica della correttezza dei dati ottenuti, e il conseguente perfezionamento del modello e dei relativi programmi.

6.3.2 L'importanza dell'analisi di superficie

L'analizzatore morfologico agisce nella profondità del testo coinvolgendo in particolare una componente linguistica, ma è l'unico esempio di questo tipo presente nel modello. È l'analisi della superficie del testo a svolgere un ruolo preponderante in questa tesi. In superficie sono cercati e trovati i paradigmi che permettono la strutturazione esplicita del testo e la conseguente estrazione d'informazione, e proprio lo studio della superficie testuale, a prescindere dalle sue componenti linguistiche sembra mostrare, per l'arabo e il testo dei *ḥadīṯ* in particolare, alcune nuove prospettive di ricerca.

6.3.3 Integrazione tra empirismo e conoscenza

Di là programmi sviluppati, che mostrano un'integrazione a diversi livelli tra approccio statistico e approccio basato su regole, una riflessione conclusiva può essere fatta sull'efficacia mostrata dalla combinazione di tecniche di analisi di tipo 'ignorante', basate cioè su algoritmi statistici e posizionali rispetto a quelle che tengono conto in una certa misura del un patrimonio di conoscenza extralinguistico, in questo caso le discipline di studio e critica della Tradizione islamica. Un caso tipico è stato affrontato nel processo di segmentazione con *HadExtractor* al momento della creazione della lista di EF che il programma avrebbe utilizzato per interpretare e suddividere un *ḥadīṯ*. Due approcci erano possibili: quello empirico di calcolo puro, che tra tutte le parole del testo identificava le EF attraverso indici di ricorrenza posizionale o statistica, o quello basato sulla conoscenza, che si serviva delle fonti critiche sulla letteratura delle tradizioni per recuperare la lista delle FE. In sede operativa sono stati sperimentati entrambi i metodi con risultati in fondo abbastanza analoghi, ma le due alternative hanno mostrato alcune fondamentali caratteristiche.

Il modello basato sulla conoscenza permette di accedere a informazioni esistenti e scientificamente affidabili, ma sviluppa la tendenza a cercare nel

testo solo ciò che già si conosce. Recuperando dalla letteratura la lista delle EF e usandole per interpretare un *ḥadīṯ*, non si potrà mai sapere se quella lista le contiene davvero tutte. È difficile cioè riuscire a scoprire nuova informazione dal punto di vista quantitativo.

Il modello statistico invece ha il proprio punto di forza nella possibilità di trovare ciò che non si conosce, proprio perché agisce su altri piani rispetto a quello della conoscenza. Le sue debolezze invece risiedono nell'impossibilità di garantire la correttezza del dato inferito a meno di una verifica manuale che si serva appunto della conoscenza, e nel rischio di produrre come risultati troppi falsi negativi (di non riuscire cioè a trovare proprio tutti i risultati possibili)

6.3.4 Il testo come ispiratore strategico

L'intero approccio metodologico mette il testo dei *ḥadīṯ* al centro del processo di definizione delle strategie computazionali. Questo ha fatto sì che molti degli strumenti creati mostrino una grande dipendenza dal contesto, che ha sì garantito un'alta efficacia di analisi ma ne limita in modo significativo l'applicazione ad altre tipologie di testo arabo.

È il caso ad esempio di *HadExtractor* o *ChainViewer*, che sono stati costruiti col fine specifico di segmentare, estrarre e rappresentare informazioni contenute nei *ḥadīṯ*. L'analisi ha programmaticamente riguardato solo la raccolta di al-Buḫārī, considerata quindi come un testo 'pilota' per la messa a punto degli strumenti computazionali stessi. Un'estensione dell'indagine ad altre raccolte di *ḥadīṯ* è quindi possibile senza alterare la struttura di programmazione, ma considerando attentamente le specificità di ciascun testo, ad esempio l'uso più o meno rigoroso delle EF all'interno dell'isnād o come indicazione di inizio del matn.

Più difficile invece è prevedere la totale trasferibilità dei programmi sviluppati ad altre tipologie testuali dell'arabo classico. Fermo restando che un testo, per essere segmentato in superficie deve già possedere un certo grado di strutturazione, è probabilmente percorribile la strada di utilizzare i presupposti teorici di questa tesi alla base dell'interpretazione del testo (la sin-

tassi delle espressioni regolari ad esempio) per concepire nuovi strumenti specifici.

L'analizzatore morfologico RAM e il programma di ricerca testuale *CrossQuran* sono invece pienamente trasferibili ad altre tipologie testuali. Le modifiche effettuate ad *AraMorph* infatti hanno lo scopo proprio di aumentare le possibilità di trattamento del testo attraverso l'espansione dei moduli lessicali e di gestione della vocalizzazione, mentre *CrossQuran* può essere applicato a qualsiasi testo in caratteri arabi, ed eventualmente modificato per poter usare come input di ricerca non solo il Corano ma virtualmente qualsiasi testo.

6.3.5 Un'elaborazione bilanciata ma ancora esemplificativa

Come ultima conclusione, il modello proposto mostra un certo bilanciamento tra capacità di analisi, rappresentazione e generazione, e può essere considerato come un embrionale tentativo di approccio globale al testo. La preoccupazione di individuare il più alto numero possibile di direzioni di ricerca ha però in qualche misura inficiato l'approfondimento e l'ulteriore elaborazione dei risultati ottenuti.

In relazione ad esempio allo studio critico dei *ḥadīṯ*, questo lavoro non propone nuove ipotesi di ricerca basate sull'interpretazione sistematica dei dati estratti ma si sofferma sulla definizione dei meccanismi alla base dell'estrazione stessa. In un ideale modello che (1) parte dalla conoscenza dei *ḥadīṯ*, (2) costruisce nuovi strumenti di analisi, (3) ottiene un certo numero di risultati e (4) li utilizza per aumentare il livello di conoscenza stessa, questo lavoro si concentra sulle fasi (2) e (3), riservando la quarta per futuri sviluppi.

6.4 Prospettive future d'indagine

L'intero corredo di strategie di analisi testuale sin qui descritte possono essere considerate come un punto di partenza, in certa misura solido ma piuttosto embrionale. Un tale approccio apre comunque alcune interessanti direzioni di ricerca per futuri studi e sviluppi di applicazioni.

6.4.1 Un sistema orientato alla consultazione e all'analisi

L'identificazione di presupposti teorici e la costruzione di modelli algoritmici è stata generalmente privilegiata rispetto alla produzione di strumenti completi, operativi e orientati all'uso. In questo senso un'esigenza urgente per il sistema è quella di dotarsi di un'interfaccia grafica che permetta la fruizione diretta e agevole dei programmi implementati e dei dati ottenuti come risultato. È prevista la realizzazione di un sistema di consultazione della base di dati prodotta, costruito attraverso un'interfaccia web dinamica che produca di volta in volta l'output richiesto dall'utente, ad esempio grafi particolari di trasmissione o interrogazioni (*queries)* specifiche per la comparazione dei dati, oltre che la possibilità di consultare interattivamente il testo di tutti i *ḥadīṯ*, la sua traduzione e tutte le informazioni relative all'*isnād* e alla catena di trasmissione (nomi e relazioni tra trasmettitori, grafi di trasmissione).

6.4.2 Estensione ad altre raccolte di tradizioni

Verificata l'ipotesi della fattibilità dell'analisi strutturale computazionale dei *ḥadīṯ*, il passo successivo potrebbe consistere nell'estensione del processo di segmentazione ed estrazione di informazioni alle sei raccolte canoniche sunnite e virtualmente ad altre raccolte di tradizioni. La forte affinità tipologica e strutturale di questi testi ridurrebbe al minimo la necessità di adattamento e specializzazione dei programmi.

6.4.3 Studi su catene di trasmissione

La rappresentazione bidimensionale a grafi delle informazioni sulle catene di trasmissione estratte dall'*isnād* potrebbe essere potenziata in due direzioni. La visualizzazione grafica andrebbe portata a un maggiore grado di sofisticazione, attraverso l'uso di indicatori a maggiore impatto visivo in merito alla tipologia di trasmissione. Il programma per la trasformazione in grafi potrebbe prevedere delle variabili vettoriali che permettano la visualizzazione contemporanea e dinamica a scale diverse di tutte le relazioni di trasmissione, che ammontano a diverse decine di migliaia nella sola raccolta

di al-Buḫārī. Infine la rete nodale di relazioni tra i trasmettitori alla base della rappresentazione a grafi, potrebbe essere ulteriormente investigata in chiave genealogica (relazioni tra le diverse generazioni di trasmettitori) e geografica (aree di provenienza dei trasmettitori).

6.4.4 Un corpus parallelo multilingue per i *ḥadīṯ*

Attraverso il miglioramento delle tecniche di allineamento automatico del testo arabo originale con la corrispettiva traduzione in un'altra lingua, si potrebbe prevedere l'estensione della memoria di traduzione per comprendere versioni in altre lingue europee o di paesi islamici.

6.4.5 Perfezionamento della trasferibilità dei programmi

Alcuni degli strumenti di analisi realizzati potrebbero essere estesi e generalizzati per poter trattare testi appartenenti ad altri generi o domini della cultura araba classica. Nel caso di segmentatori e interpreti di struttura come *HadExtractor*, il prerequisito essenziale resta che i testi da trattare presentino un certo grado di strutturazione o formalizzazione nell'organizzazione linguistica dei contenuti, come nel caso di repertori genealogici, raccolte di biografie, dizionari specializzati e lessici di definizioni. Solo il riconoscimento di una struttura testuale, esplicita o implicita, permetterebbe infatti l'adattamento e l'ideazione di algoritmi per la segmentazione attraverso la sintassi delle espressioni regolari.

Se *CrossQuran* garantisce la ricerca delle citazioni coraniche all'interno dei *ḥadīṯ*, due sviluppi possibili consisterebbero nel diminuire la dimensione minima della stringa da cercare (per ora il programma cerca interi versetti) e nel trasformare il programma in un dispositivo per cercare parti di un testo A in un testo B. In questo senso sarebbe per esempio virtualmente possibile cercare le citazioni e i riferimenti a passi dei *ḥadīṯ* in qualsiasi genere di testo arabo.

6.4.6 Grammatica formale per l'interpretazione globale del testo

Il contenuto del *matn* segmentato da *HadExtractor* e lessicalizzato da RAM, potrebbe costituire la base per la realizzazione computazionale di un modello grammaticale formale capace di analizzare il testo e generarne di nuovo sulla base di un set di informazioni estratte semanticamente. Un possibile candidato per il modello è la grammatica categoriale-combinatoria, la cui riduzione della morfosintassi a componente semantico si è dimostrata particolarmente adatta alla conversione in algoritmi e applicazioni computazionali (Baldridge, Chatterjee, Palmer, & Wing, 2007; Baldridge & Kruijff, 2003) e all'applicazione alla lingua araba (Lancioni, Categorial Grammar and Arabic: Syntactic and typological issues, 2010).

Bibliografia

Abbot, N. (1972). *Studies in Arabic Literary Papyri, Vol. III: Language and Literature.* Chicago: The University of Chicago Press.

Abu El-Khair, I. (2007). Arabic Information Retrieval. *Annual review of information science and technology*(41), 505-533.

Aho, A. V. (1990). Algorithms for finding patterns in strings. In J. van Leeuwen, *Handbook of Theoretical Computer Science* (Vol. A, p. 255-300). Boston: The MIT Press.

Al-ʻAsqalānī, A. (1959). *Fatḥ al-bārī bi-sharḥ Ṣaḥīḥ al-Bukhārī.* Bayrūt: Dār al-Maʻrifa.

Al-Buḫārī, M. I. (1862-1908). *Saḥīḥ.* Leiden: Krehl and Juynboll.

Al-Buḫārī, M. I. (1984). *Sahih Al Bukhari.* (M. M. Khan., Trad.) Alexandria: Al-Saadawi Publications.

Al-Buḫārī, M. I. (2001). *Al-Jāmiʿ al-Ṣaḥīḥ* . Dār Ṭawq an-Najāh [Edizione digitalizzata tratta da www.almeskkat.net].

Al-Buḫārī, M. I. (s.d.). *Ṣaḥīḥ al-Bukhārī* (1990 ed.). Riyaḍ: Dār Ṭūq al-Najāh.

Al-Najem, S. R. (2007). Inheritance-based Approach to Arabic Verbal Root-and-Pattern Morphology. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology* (p. 67-88). Dordrecht: Springer.

Al-Sughaiyer, I. A., & Al-Kharashi, I. A. (2004). Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology, 3*(55), 189-213.

Amna, M. H. (2011). *Translation of hadith from arabic to english using rule-based approach.* Bangi: Universiti Kebangsaan Malaysia.

Attia, M. (2006). An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks. *The Challenge of Arabic.* LOndon: The British Computer Society.

Attia, M. (2007). Arabic tokenization system. *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources* (p. 65-72). Prague: Association for Computational Linguistics.

Azami, M. (1977). *Studies in Hadith Methodology and Literature.* Indianapolis: American Trust Publications.
Badr, I., Zbib, R., & Glass, J. (2008). Segmentation for English-to-Arabic Statistical Machine Translation. *Proceedings of ACL-08* (p. 153-156). Columbus: ACL.
Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* New York: ACM Press.
Baldridge, J., & Kruijff, G. (2003). Multi-Modal Combinatory Categorial Grammar. *Proceedings Of 10th Annual Meeting of the EACL*, (p. 211-218). Budapest.
Baldridge, J., & Kruijff, G. (2007). *Course Notes on Combinatory Categorial Grammar.* Tratto da http://groups.inf.ed.ac.uk/ccg
Baldridge, J., Chatterjee, S., Palmer, A., & Wing, B. (2007). DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG. *Proceedings of the Workshop on Grammar Engineering Across Frameworks.* Stanford, CA.
Beazley, D. M. (2000). *Advanced python programming.* O'Reilly Open Source Conference.
Bebah, M., Belahbib, R., Boudlal, A., Lakhouaja, A., Mazroui, A., & Meziane, A. (2011). A Markovian Approach for Arabic Root Extraction. *The International Arab Journal of Information Technology, 8*(1).
Beesley, K. (1996). Arabic finite-State morphological analysis and generation. *Proceedings of COLING'96*, *1*, p. 89-94.
Beesley, K. (1998). Consonant spreading in Arabic stems. *Proceedings of the 17th international conference on Computational linguistics - COLING98. 1*, p. 117-123. Stroudsburg: Association for Computational Linguistics.
Berg, H. (2000). *The Development of Exegesis in Early Islam: The Authenticity of Muslim Literature from the Formative Period.* London: Curzon.
Berge, C. (1958). *Théorie des graphes et ses applications.* Paris: Dunod.
Bernstein, D. J. (2005). *Understanding brute force.* Chicago: University of Illinois.
Biemann, C. (2012). *Structure Discovery in Natural Language.* Springer.
Bird, S., & Klein, E. (2006). *Regular Expressions for Natural Language Processing.* University of Pennsylvania.
Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python.* Sebastopol: O'Reilly.
Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., et al. (2006). Introducing the Arabic WordNet Project. *Proceedings of the*

Third International WordNet Conference. Sojka, Choi, Fellbaum and Vossen eds.

Boella, M. (2011a). Reading a text, finding a database: an anachronistic interpretation of hadiths in light of information science. In L. Capezzone (A cura di), *Uscire dal tempo. percezioni dell'antico, del moderno, del futuro* (p. 439-448). Roma: Rivista di Studi Orientali.

Boella, M. (2011b). Regular expressions for interpreting and cross-referencing Hadith texts. *Langues et Littératures du Monde Arabe, 9*, 25-39.

Boella, M., Romani, F. R., Al-Raies, A., Solimando, C., & Lancioni, G. (2011). The SALAH Project: Segmentation and Linguistic Analysis of ḥadīṯ Arabic Texts. In M. V. Salem, K. Shaalan, F. Oroumchian, A. Shakery, & H. Khelalfa (A cura di), *Information Retrieval Technology* (p. 538-549). Heidelberg: Springer.

Bohas, G., & Guillaume, J. (1984). *Étude des théories des grammairiens arabes, I. Morphologie et phonologie.* Damascus.

Bolshakov, I. A., & Gelbukh, A. (2004). *Computational linguistics: models, resources, applications.* IPN-UNAM-FCE.

Bondy, J. A., & Murty, U. S. (2008). *Graph Theory.* Heidelberg: Springer.

Boone, R. (2003). Reading, Writing, and Publishing Digital Text. *Remedial and Special Education, 24*(3), 132-140.

Brass, P. (2008). *Advanced Data Structures.* Cambridge University Press.

Brown, J. (2007). *The Canonization of al-Bukhārī and Muslim: the Formation and Function of the Sunnī Ḥadīṯ Canon.* Leiden: Brill.

Buckwalter, T. (2002). *Buckwalter Arabic Morphological Analyzer Version 1.0.* Linguistic Data Consortium, University of Pennsylvania.

Buckwalter, T. (s.d.). *Buckwalter Arabic transliteration.* Tratto da http:qamus.org/transliteration.htm

Burton, J. (1994). *An introduction to the Hadith.* Edinburgh: Edinburgh University Press.

Busa, R. (1974, febbraio 2). L'Index Thomisticus - Contenuto, Finalità, Prospettive. *La Civiltà Cattolica*(2967), p. 250-257.

Cahill, L. (1990). Syllable-based morphology. *COLING-90, 3*, p. 48-53. Helsinki.

Cahill, L. (2007). A Syllable-based Account of Arabic Morphology. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (p. 45-66). Dordrecht: Springer.

Cardie, C., & Mooney, R. (1999). Machine learning and natural language. *Machine Learning, 1-3*(11), 1-5.
Cavalli-Sforza, V., & Soudi, A. (2007). Arabic Computational Morphology: A Trade-off Between Multiple. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (p. 89-114). Dordrecht: Springer.
Celentano, A., Cortesi, A., & Mastandrea, P. (2004). Informatica Umanistica: una disciplina di confine. *Mondo Digitale*(4), 44-55.
Chuang, T. C., Jian, J.-y., Chang, Y.-c., & Chang, J. S. (2005). Collocational Translation Memory Extraction Based on Statistical and Linguistic Information. *Computational Linguistics and Chinese Language Processing, 1*(10), 329-346.
Clark, A. (2007). Supervised and Unsupervised Learning of Arabic Morphology. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology* (p. 181-200). Dordrecht: Springer.
Coulmas, F. (1994). Typology of Writing Systems. *Handbücher zur Sprach- und Kommunikations- wissenschaft, 10*(2), 1380-1387.
Crone, P. (1980). *Slaves on Horses: the Evolution of the Islamic Polity., Cambridge.* Cambridge: Cambridge University Press.
Cu, G. i., Lu, Q., Li, W., & Chen, Y. (2008). Corpus Exploitation from Wikipedia for Ontology Construction. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, (p. 2125-2132). Marrakech.
Curran, J. R., Clark, S., & Bos, J. (2007). Linguistically Motivated Large-Scale NLP with C&C and Boxer. *Proceedings of the ACL 2007 Demonstrations Session*, (p. 33-36).
Daya, E., Roth, D., & Wintner, S. (2007). Learning to Identify Semitic Roots. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (p. 143-158). Dordrecht: Springer.
Di Battista, G., Eades, P., Tamassia, R., & Tollis, I. (1999). *Graph Drawing; Algorithms for the visualization of graphs.* Upper Saddle River: Prentice Hall.
Diab, M., Hacioglu, K., & Jurafsky, D. (2007). Automatic Processing of Modern Standard Arabic Text. In A. C.-b. Methods, A. Soudi, A. van den Bosch, & G. Neumann (A cura di). Dordrecht: Springer.

Dichy, J., & Farghali, A. (2003). Roots and Patterns vs. Stems plus Grammar-Lexis Specifications: on what basis should a multilingual lexical database centred on Arabic be. *IXth MT Summit Workshop on Machine Translation for Semitic Languages: Issues and Approaches*, (p. 1-8). New Orleans.

Dichy, J., Braham, A., Ghazali, S., & Hassoun, A. (2002). La base de connaissances linguistiques DIINAR.1 (DIctionnaire INformatisé de l'ARabe, version 1). In A. Braham (A cura di), *Proceedings of the International Symposium on The Processing of Arabic.* Tunis: Université de la Manouba.

Dukes, K. (2010). *Quranic Arabic Corpus (version 0.2) based on Tanzil Quran Text (Uthmani, version 1.0.2)*. Tratto da http://corpus.quran.com/

Dukes, K., Atwell, E., & Sharaf, A. (2010). Online Visualization of Traditional Quranic Grammar using Dependency Graphs. *The Foundations of Arabic Linguistics Conference.* Cambridge.

Eijck, J. v., & Unger, C. (2010). *Computational Semantics with Functional Programming.* Cambridge: Cambridge University Press.

Elgibali, A. (A cura di). (2005). *Investigating arabic. current parameters in analysis and learning.* Leiden: Brill.

Faro, S., & Lecroq, T. (2013). The Exact Online String Matching Problem: a Review of the Most Recent Results. *ACM Computing Surveys, 45*(2).

Fauzan, M., & Othman, N. (2006). *An Information Retrieval System for Quranic Texts: A Proposed System Design Faculty of ICT.* Malaysia: International Islamic University.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge: MIT Press.

Friedl, J. (2002). *Mastering Regular Expressions.* Sebastopol: O'Reilly.

Fück, J. (1938). Beiträge zur Überlieferungsgeschichte von Buḫāris Traditionssammlung. *Zeitschrift der Deutschen Morgenländischen Gesellschaft*, 60-87.

Fück, J. (1939). Die Rolle des Traditionalismus im Islam. *Zeitschrift der Deutschen Morgenliindischen Gesellschaft*(93), 1-32.

Ghoniem, M., Tokal, K., & Tawfik, A. (2011). An Analysis and Evaluation of English Arabic Statistical Machine Translation of Terminology-Rich Text. *Proceedings of 9th International Conference on Terminology and Artificial Intelligence*, (p. 115-118). Paris.

Gibbon, D., Moore, R., & Winski, R. (A cura di). (1997). *Handbook of Standards and Resources for Spoken Language Systems.* Berlin: Mouton de Gruyter.

Gill, A. (1962). *Introduction to the Theory of Finite-state Machines.* McGraw-Hill.
Goldziher, I. (1889–1890). *Muhammedanische Studien.* Halle.
Gómez-Pérez, A., & Corcho, O. (2002). Ontology languages for the semantic web. *IEEE Intelligent Systems*, 54— 60.
Goutte, C., Cancedda, N., Dymetman, M., & Foster, G. (2009). *Learning Machine Translation.* Cambridge: The MIT Press.
Goyvaens, J., & Levitan, S. (2009). *Regular Expressions Cookbook.* Sebastopol: O'Reilly.
Grefenstette, G., Semmar, N., & Elkateb-Gara, F. (2005). Modifying a natural language processing system for european languages to treat arabic in information processing and information retrieval applications. *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages* (p. 31-38). Ann Arbor: Association for Computational Linguistics.
Grune, D., & Jacobs, C. J. (1990). *Parsing Techniques - A Practical Guide.* Amsterdam: Vrije Universiteit Amsterdam.
Guessoum, A., & Zantout, R. (2001). A Methodology for a Semi-Automatic Evaluation of the Lexicons of Machine Translation Systems. *Machine Translation, 16*(2), 127-149.
Guidère, M. (2002). Toward Corpus-Based Machine Translation for Standard Arabic. *TRanslation Journal, 6*(1).
Günther, S. (2005). Assessing the Sources of Classical Arabic Compilations: The Issue of Categories and Methodologies. *British Journal of Middle Eastern Studies, 1*(32), 75–98.
Habash, N., & Rambow, O. (2005). Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell woop. *Proceedings of the 43rd Annual Meeting of the ACL* (p. 573–580). Ann Arbor.
Habash, N., & Sadat, F. (2006). Arabic Preprocessing Schemes for Statistical Machine Translation. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, (p. 49-52).
Habash, N., Soudi, A., & Buckwalter, T. (2007). On Arabic Transliteration. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (p. 15-22). Dordrecht: Springer.
Harris, Z. S. (1941). The Linguistic Structure of Hebrew. *Journal of the American Oriental Society*(62), 143–167.

Hebresha, H., & Ab Aziz, M. (2013). Classical Arabic English Machine Translation Using Rule-based Approach. *Journal of Applied Sciences*, 79-86.
Hetland, M. L. (2005). *Beginning python. from novice to professional.* Berkeley: Apress.
Hirschman, L., & Gaizauskas, R. (2001). Natural Language Question Answering. The View from Here. *Natural Language Engineering, 4*(7), 275-300.
Hunston, S. (2005). Corpus Linguistics. In K. Brown (A cura di), *Encyclopedia of Language and Linguistics* (II ed.). Elsevier.
Isac, D., & Reiss, C. (2008). *I-LANGUAGE: An introduction to linguistics as cognitive science.* Oxford: Oxford University Press.
Jackson, P. M. (2002). *Natural Language Processing for Online Applications: Text Retrieval, Extraction & Categorization.* Amsterdam: John Benjamins.
Jackson, P., & Schilder, F. (2005). Natural LanguageProcessing: Overview. In K. Brown (A cura di), *Encyclopedia of Language and Linguistics 2nd Edition* (II ed., p. 503-518). Elsevier.
Jarrar, M. (2011). Building a Formal Arabic Ontology [in Arabic]. *Proceedings of the Experts Meeting on Arabic Ontologies and Semantic Networks.* Tunis: Alecso, Arab League.
Jones, C., & Drake, F. (2002). *Python & XML.* Sebastopol: O'Reilly.
Juynboll, G. H. (1983). *Muslim Tradition: Studies in Chronology, Provenance and Authorship of Early Hadith.* Cambridge: Cambridge University Press.
Juynboll, G. H. (1989). Some Isnad -Analytical Methods Illustrated on the Basis of Several Women-Demeaning Sayings from Hadith Literature. *Al-Qantara*(10), 343-384.
Juynboll, G. H. (2001). (Re)Appraisal of Some Technical Terms in Ḥadīth Science. *Islamic Law and Society, 3*(8), 303-349.
Juynboll, G. H. (2007). *Encyclopedia of Canonical Hadith.* Leiden: Brill.
Kaufmann, M., & Wagner, D. (2001). *Drawing Graphs: Methods and Models.* Heidelberg: Springer.
Kay, M. (1987). Non-concatenative finite-state morphology. *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, (p. 2-10). Copenhagen.
Kiraz, G. (2000). A Multi-tiered Nonlinear Morphology using Multi-tape Finite State. *Computational Linguistics, 1*(26), 77-105.
Kirschenbaum, M. G. (2010). What Is Digital Humanities and What's It Doing in English Departments? *ADE Bulletin*(150), 1-7.

Koehn, P. (2010). *Statistical Machine Translation.* Cambridge: Cambridge University Press.
Kouloughli, D. (2008). Initiation pratique à la constitution et à l'exploitation de corpus électroniques en langue arabe (IIIe partie). *Langues et Littératures du Monde Arabe*(7), 75-93.
Kouloughli, D. (2009). Initiation pratique à la constitution et à l'exploitation de corpus électroniques en langue arabe (IVe partie). *Langues et Littératures du Monde Arabe*(8), 117-133.
Kroenke, M., & Auer, D. J. (2007). *Database Concepts.* New York: Prentice.
Kuchling, A. M. (2002). *Regular Expressions HOWTO.* Python.org.
Kutz, O., Normann, I., Mossakowski, T., & & Dirk, W. (2010). Chinese whispers and connected alignments. *Proceedings of the 5th International Workshop on Ontology Matching, OM-2010.* Shangai.
Lagoudaki, E. (2006). Translation Memory systems: Enlightening users' perspective. Key finding of the TM Survey 2006 carried out during July and August 2006. In *Translation Memories Survey.* London: Imperial College.
Lancioni, G. (2008). Variants, Links, and Quotations: Classical Arabic Texts as Hypertexts. In D. Bredi, L. Capezzone, W. Dahmash, & L. R. (A cura di), *Scritti in Onore di Biancamaria Scarcia Amoretti* (Edizioni Q ed., Vol. 2). Roma.
Lancioni, G. (2010). *Categorial Grammar and Arabic: Syntactic and typological issues.*
Lancioni, G. (2011). *An Adaptation of Buckwalter Transcription Model to XML and Regular Expression Syntax.* Roma Tre University.
Laroux, N. (1993). Élogie de l'anachronisme en histoire. *Le Genre humain*(27), 23-38.
Leder, S., & & Kilpatrick, H. (1992). Classical arabic prose literature: a researchers' sketch map. *Journal of Arabic Literature, 1*(23), 2-26.
Leder, S., & Kilpatrick, H. (1992). Classical arabic prose literature: a researchers' sketch map. *Journal of Arabic Literature, 1*(23), 2-26.
Lucas, S. (2002). *The Arts of Hadith Compilation and Criticism.* University of Chicago.
Lutz, M. (2007). *Learning python* (3 ed.). Sebastopol: O'Reilly.
Ma, Y., He, Y., Way, A., & van Genabith, J. (2011). Consistent translation using discriminative learning - a translation memory-inspired approach. *Proceedings of the 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies (p. 1239-1248). Portland: Association for Computational Linguistics.
Maamouri, M., Bies, A., Buckwalter, T., & Mekki, W. (2004). The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. *NEMLAR Conference on Arabic Language Resources and Tools.*
Madhany, H. N. (2006). *Multilingual computing with Arabic and Arabic transliteration.* Chicago: The University of Chicago Press.
Magerman, D. M. (1995). Statistical decision-tree models for parsing. *Proceedings of the 43rdAnnual Meeting of the Association for Computational Linguistics ACL-95* (p. 276-283). Ann Arbor: Association for Computational Linguistics.
Maghen, Z. (2003). Dead tradition: joseph schacht and the origins of "popular practice". *Islamic Law and Society, 3*(10), 276— 347.
Makhoul, J., Kubala, F., Schwartz, R., & Weischedel, R. (1999). Performance measures for information extraction. *Proceedings of DARPA Broadcast News Workshop.* Herndon.
Mani, I., & Maybury, M. (1997). *Advances in automatic text summarization.* Cambridge: MIT Press.
Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., & Schneide, L. (2003). *The WonderWeb Library of Foundational Ontologies. Preliminary Report.* WonderWeb Deliverable D17.
Matuszek, C., Cabral, J., Witbrock, M., & DeOliveira, J. (2005). *An Introduction to the Syntax and Content of Cyc.*
McCarthy, J. (1981). Prosodic Theory of Non-Concatenative Morphology. *Linguistic Inquiry*(12), 373-418.
McCarty, W. (2005). *Humanities Computing.* Basingstoke: Palgrave Macmillan.
Meedan. (2009). Meedan v.10. Arabic-English Translation Memory. San Francisco.
Melamed, D. (1996). A Geometric Approach to Mapping Bitext Correspondence. *Conference of Empirical Methods in NLP*, (p. 1-12). Philadelphia.
Melchert, C. (2001). Bukhārī and Early Hadith Criticism. *Journal of the American Oriental Society*(121), 7-19.
Mertz, D. (2003). *Text Processing in Python.* Addison Wesley.
Miller, G. (1995). WordNet: A Lexical Database for English. *Communications of the ACM, 38*(11), 39-41.
Motzki, H. (2007). Hadith: Origins and Developments. *Journal of Islamic Studies, 18*(1), 97-99.

Niles, I., & Pease, A. (2001). Towards a Standard Upper Ontology. In C. Welty, & B. Smith (A cura di), *Proceedings of FOIS-2001.* Ogunquit.

Och, F. J., Tillmann, C., & Ney, H. (1999). Improved alignment models for statistical machine translation. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)* (p. 20-28). University of Maryland.

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques.* Dordrecht: Springer.

Osh, F., & Ney, H. (2003). "A Systematic Comparison of Various Statistical Alignment Models. *Computer Journal of Computational Linguistics, 29*(2), 19-51.

OSI. (1998). *Open Source Initiative*. Tratto il giorno gennaio 16, 2013 da http://opensource.org/

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU: A method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (p. 311-38). Philadelphia: AACL.

Pasternack, J. (2008). *The Wikipedia Corpus.*

Perkins, J. (2010). *Python text processing with nltk 2.0.* Birmingham: Packt Pub.

Philips, A. A. (2007). *Usool al-hadeeth : the methodology of hadith evaluation.* Riyadh: International Islamic Publishing House.

Powers, D. M. (2007). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies, 1*(2), 37-63.

Powers, D. S. (1986). *Studies in qur'an and hadith : the formation of the islamic law of inheritance.* Berkeley: University of California Press.

Rau, L. (1991). Extracting Company Names from Text. *Proceedings of the Seventh Conference on Artificial Intelligence Applications*, (p. 29-32). Miami Beach.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks.*

Reichmuth, P. (2009). Tanscription. In K. Versteegh, M. Eid, A. W. Elgibali, & A. Zaborski (A cura di), *Encyclopedia of Arabic Language and Linguistics* (Vol. 4, p. 515-520). Leiden - Boston: Brill.

Rice, S. (1999). Aspects of prepositions and prepositional aspect. *Proceedings of the International Cognitive Linguistics Conference* (p. 225-248). New York: Mouton de Gruyter.

Roark, B., & Sproat, R. W. (2007). *Computational approaches to morphology and syntax.* Oxford, New York: Oxford University Press.
Roberts, A. A.-S. (2006). aConCorde: Towards an open-source, extendable concordancer for Arabic. *Corpus-based Language Learning, Language Processing and Linguistics, 1*.
Robson, J. (1961). Standard applied by Muslim traditionists. *Bulletin of the John Rylands Library*(63).
Robson, J. (1978). Ḥadīth. In *Enciclopaedia of Islam* (Vol. III, p. 23-28). Leiden: Brill.
Sag, I. A., Baldwin, T., Bond, F., Copestake, A., & Flickinger, D. (2002). Multiword Ex-pressions: A Pain in the Neck for NLP. *CICLing-2002*, (p. 1-15).
Salameh, M., Zantout, R., & Mansour, N. (2011). Improving the Accuracy of English-Arabic Statistical Sentence Alignment. *The International Arab Journal of Information Technology, 8*(2).
Salem, Y., & Nolan, B. (2009). UNIARAB: An universal machine translator system for Arabic Based on Role and Reference Grammar. *Proceedings of the 31st Annual Meeting of the Linguistics Association of Germany.*
Salmoné, H. (1889). *An Advanced Learner's Arabic-English Dictionary.* Beirut : Librairie du Liban.
Schacht, J. (1950). *The Origins of Muhammadan Jurisprudence.* Oxford: Oxford University Press.
Schütze, H. (1997). *Ambiguity resolution in language learning : computational and cognitive models.* Stanford: Center for the Study of Language and Information.
Sebesta, R. W. (2006). *Concepts of programming languages.* Boston: Pearson/Addison-Wesley.
Sezgin, F. (1996). *Geschichte des arabischen Schrifttums, Band I: Qur'ānwissenschaften, ḥadīt, Geschichte, Fiqh, Dogmatik, Mystik. Bis ca. 430 H.* Leiden: Brill.
Shaalan, K. (2010). Rule-based approach in Arabic natural language processing. *International Journal of Information and Communication Technology, 3*, 11-19.
Shenassa, M. a. (2008). *Evaluation of Different English Translations of Holy Koran in Scope of Verb Process Type.* Teheran: Islamic Azad University.

Simard, M. (2003). Translation Spotting for Translation Memories. *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, 3, p. 65-72.
Simard, M., & Foster, G. (1992). Using Cognates to Align Sentences in Bilingual Corpora. *Proceedings of 4th International Conference on Theoretical and Methodological Issues in Machine Translation*, (p. 67-81).
Skonnard, A., & Gudgin, M. (2001). *Essential XML Quick Reference.* Boston: Addison-Wesley.
Soudi, A., van den Bosch, A., & Neumann, G. (2007). Arabic Computational Morphology. Knowledge-based and Empirical Methods. In A. Soudi, A. van den Bosch, & G. Neumann (A cura di), *Arabic Computational Morphology. Knowledge-based and Empirical Methods* (p. 3-14). Dordrecht: Springer.
Speight, M. R. (2000). Narrative Structures in the Hadith. *Journal of Near Eastern Studies, 59*(4), 265-271.
Sprenger, A. (1869). *Das Leben und die Lehre des Mohammad : nach bisher Grosstentheils Unbenutzten Quellen.* Berlin: Nicolaische Verlagsbuchhandlung.
Svensson, P. (2009). Humanities Computing as Digital Humanities. *Digital Humanities Quarterly, 3*(3).
TEI Consortium. (2009). *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* Oxford: The TEI Consortium.
Tottoli, R. (2002). *Hadith in modern islam.* Roma: Istituto per l'Oriente C.A. Nallino.
Traini, R. (1993). *Vocabolario Arabo - italiano.* Roma: Istituto per l'Oriente.
Vacca, V., Noja, S., & Vallaro, M. (A cura di). (2009). *Al-Bukhari, Detti e fatti del Profeta dell'islam.* Torino: UTET.
van Rijsbergen, C. J. (1979). *Information Retrieval.* London: Butterworths.
van Rossum, G. (1993). An Introduction to Python for UNIX/C Programmers. *Proceedings of the NLUUG najaarsconferentie.*
Vlachos, A. (2011). Evaluating unsupervised learning for natural language processing tasks. *Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing* (p. 35-42). Edinburgh: ACL.
Vossen, P. (1998). *EuroWordNet: a multilingual database with lexical semantic networks for European Languages.* Dordrecht: Kluwer.
Weil, G. (1848). *Geschichte der Chalifen nach handschriftlichen grosstentheils noch unbenuzten Quellen* (Bd. 2). Mannheim: F. Bassermann.

Wellisch, H. (1975). *Transcription and transliteration: An annotated bibliography on conversion of scripts.* Silver Spring: Institute of Modern Languages.
Wensinck, A. (1933). *Concordance et indices de la tradition musulmane : Le six Livres, le Musnad d'al-Dārimī, le Muwaṭṭa de Malik, le Musnad de Aḥmad ibn Ḥanbal.* Leiden: Brill.
Wolska, M., & Kruijff-Korbayová, I. (2004). Analysis of mixed natural and symbolic language input in mathematical dialogs. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.* Stroudsburg: Association for Computational Linguistics.
Wright, D. R. (2005). *Finite State Machines. Class Notes.* North Carolina State University.
Xerox. (2013, febbraio 24). *Arabic Morphological Analyzer*. Tratto da Open Xerox: http://open.xerox.com/Services/arabic-morphology/Pages/About%20the%20Arabic%20Morphological%20Analyzer
Zelinsky-Wibbelt, C. (1993). *The Semantics of prepositions: from mental processing to natural language processing.* Berlin: Walter de Gruyter.
Zelle, J. M. (2003). *Graphics module reference (python).*
Ziadé, T. (2008). *Expert python programming : [learn] best practices for designing, coding, and distributing your python software.* Birmingham: Packt publishing.
Zughoul, M., & Abu-Alshaar, A. (2005). English/Arabic/English Machine Translation: A Historical Perspective. *Translators' Journal, 50*(3), 1022-1041.

APPENDICE

A. Codici dei programmi

Salvo diversa ed esplicita indicazione, i seguenti programmi sono stati integralmente concepiti e compilati dall'autore della presente dissertazione. Le righe precedute dal simbolo # vengono ignorate dal computer in fase di compilazione e contengono i commenti fatti dall'autore ma utili forse al lettore per la comprensione dei vari algoritmi. I programmi esistenti e usati nella loro versione originale, i programmi modificati in alcune parti, i testi dell'input e alcuni risultati sono repertoriati in un archivio specifico consultabile all'indirizzo: mbpro.net/PhdThesis.

A.1 *HadExtractor*[79]

```
# -*- coding: utf-8 -*-
'''HadExtractor - Draft di programma per la segmentazione
tramite RE di hadith
marco.boella@alice.it
Versione 1.0 - 27 aprile 2013'''

import codecs, re, operator, unicodedata

#========function for back transliteration transl ==> arabic
==================
def conv(text):
    mapping_source={
       'ARABIC COMMA' : ',',
##         'ARABIC COMMA' : '،',
        'ARABIC SEMICOLON' : ';',
        ##'ARABIC TRIPLE DOT PUNCTUATION MARK' : '...',
        'ARABIC QUESTION MARK' : '?',
        'ARABIC LETTER HAMZA' : '-',
        'ARABIC LETTER ALEF WITH MADDA ABOVE' : 'O',
        'ARABIC LETTER ALEF WITH HAMZA ABOVE' : 'E',
        'ARABIC LETTER WAW WITH HAMZA ABOVE' : 'W',
```

[79] Il sotto modulo di traslitterazione/ritraslitterazione è una modifica di un programma realizzato da Giuliano Lancioni.

```
'ARABIC LETTER ALEF WITH HAMZA BELOW' : 'e',
'ARABIC LETTER YEH WITH HAMZA ABOVE' : 'J',
'ARABIC LETTER ALEF' : 'A',
'ARABIC LETTER BEH' : 'b',
'ARABIC LETTER TEH MARBUTA' : 'o',
'ARABIC LETTER TEH' : 't',
'ARABIC LETTER THEH' : 'C',
'ARABIC LETTER JEEM' : 'j',
'ARABIC LETTER HAH' : 'H',
'ARABIC LETTER KHAH' : 'x',
'ARABIC LETTER DAL' : 'd',
'ARABIC LETTER THAL' : 'V',
'ARABIC LETTER REH' : 'r',
'ARABIC LETTER ZAIN' : 'z',
'ARABIC LETTER SEEN' : 's',
'ARABIC LETTER SHEEN' : 'X',
'ARABIC LETTER SAD' : 'S',
'ARABIC LETTER DAD' : 'D',
'ARABIC LETTER TAH' : 'T',
'ARABIC LETTER ZAH' : 'Z',
'ARABIC LETTER AIN' : 'c',
'ARABIC LETTER GHAIN' : 'G',
'ARABIC LETTER FEH' : 'f',
'ARABIC LETTER QAF' : 'q',
'ARABIC LETTER KAF' : 'k',
'ARABIC LETTER LAM' : 'l',
'ARABIC LETTER MEEM' : 'm',
'ARABIC LETTER NOON' : 'n',
'ARABIC LETTER HEH' : 'h',
'ARABIC LETTER WAW' : 'w',
'ARABIC LETTER ALEF MAKSURA' : 'Y',
'ARABIC LETTER YEH' : 'y',
'ARABIC FATHATAN' : 'N',
'ARABIC DAMMATAN' : 'U',
'ARABIC KASRATAN' : 'I',
'ARABIC FATHA' : 'a',
'ARABIC DAMMA' : 'u',
'ARABIC KASRA' : 'i',
'ARABIC SHADDA' : '+',
'ARABIC SUKUN' : '_',
'ARABIC-INDIC DIGIT ZERO' : '0',
'ARABIC-INDIC DIGIT ONE' : '1',
'ARABIC-INDIC DIGIT TWO' : '2',
'ARABIC-INDIC DIGIT THREE' : '3',
'ARABIC-INDIC DIGIT FOUR' : '4',
```

```
        'ARABIC-INDIC DIGIT FIVE' : '5',
        'ARABIC-INDIC DIGIT SIX' : '6',
        'ARABIC-INDIC DIGIT SEVEN' : '7',
        'ARABIC-INDIC DIGIT EIGHT' : '8',
        'ARABIC-INDIC DIGIT NINE' : '9',
        'ARABIC PERCENT SIGN' : '%',
        'ARABIC DECIMAL SEPARATOR' : '.',
        'ARABIC THOUSANDS SEPARATOR' : ',',
        'ARABIC LETTER VEH' : 'v',
        'ARABIC LETTER PEHEH' : 'p',
        }
    un = (lambda x: ord(unicodedata.lookup(x)))
    na = (lambda x: unicodedata.name(unicode(x)))
##mapping_ar_tr=dict([(un(item[0]),unicode(item[1]))  for  item  in
mapping_source.items()])
    mapping_tr_ar = dict([(un(na(item[1])),un(item[0])) for item in
mapping_source.items()])
    output = text.translate(mapping_tr_ar)
##    output = unicode(text,'utf-8').translate(mapping_tr_ar)
    return output
def volume_number(a):
    a = int(a)
    if a<875:
        v = 1
    elif (a>=876 and a<=1772):
            v = 2
    elif a>=1773 and a<=2737 :
        v = 3
    elif a>=2738 and a<=3648 :
        v = 4
    elif a>=3649 and a<=4414 :
        v = 5
    elif a>=4415 and a<=5062 :
        v = 6
    elif a>=5063 and a<=5969 :
        v = 77
    elif a>=5970 and a<=6860 :
        v = 8
    elif a > 6860 :
        v = 9
    else :
        v = 0
    return v
cont_isn = 0
cont_mat = 0
```

```
# ===input and output files' operations===
##f = codecs.open('b_AT_test.txt',encoding = 'utf-8')
f = codecs.open('b_AT_full.txt',encoding = 'utf-8')
text = f.read()
f.close()
##nuovo per libri=============================
perlibri= text
pl      =      re.compile(r'\s(?=bis_mi      All\+ahi      Alr\+aH_mani
Alr\+aHiymi\skitaAb\s+?|kitaAb   .*\sbis_mi   All\+ahi   Alr\+aH_mani
Alr\+aHiymi\s+?)+') ## trova un \s seguito da: (basmala + kitaAb...)
oppure (kitaAb .* + basmala)
libri = pl.split(perlibri)
ind=1
for libro in libri[:2]:
    p22 = re.compile(r'\s*(?=baAb\s+?)')
    print 'Libro n. ' + str(ind) + ':'
    ind +=1
    capitoli = p22.split(libro)
##    for capitolo in capitoli:
##        print capitolo
##        print '_____________'
    print '\n========================================='
print 'Numero totale libri:' + str(len(libri))
## fine nuovo per libri=================
##f = codecs.open('HadExtractedTEST.xml','w', encoding='utf-8')
fTR = codecs.open('HadExtractedTR.xml','w', encoding='utf-8')
f = codecs.open('HadExtractedAR.xml','w', encoding='utf-8')
f.write('<?xml version="1.0"?>\n')
f.write('<HADITH_EXTRACTED>\n')
fTR.write('<?xml version="1.0"?>\n')
fTR.write('<HADITH_EXTRACTED>\n')
lista_trasm = codecs.open('Trasm_list.txt', 'w', encoding='utf-8')
trasmList=[]
# ===First RE, which segments the full text in single hadiths
(through a lookahead expression in RE and a method that splits the
text or before hadith's number or before non-hadith's text
(es:baAb... .)===

p = re.compile(r'\s*?(?=\d+?)')
##p = re.compile(r'\s*?(?=\d+?)'+r'\n(?=baAb+?)')
q = p.split(text)
# ===Second RE, which separates isnad from matn. ===
p1 = re.compile(r"""
     ^(?P<num>\d+?)      # look for hadith source's number (not al-
ways unique)
```

```
    [ ]*?-+?[ ]*?      # look for dash with/without empty spaces
    (?P<isn>.*?)       # identifies this part of text as isnad
    (?P<sep>            # here it begins the identification of "separation text", between isnad and matn.
    (Ean\+a+?.{0,50}?qaAla(?:t_|A|[ ])+?)        # identify "'anna" as separator only if it is followed by a "qala" by 50 characters (weird but it works...)
    |(Ean\+a(?:humaA|hum_|hu|haA|[                ])|qaAla(?:t_|A|[ ])|(?:y|t)aquwlu+?)

(?!.{0,100}(samic(?:_tu|a)|Had\+aCaniy|Had\+aCanaA|Eax_baranaA|Eax_barahu|Eax_baraniy|Ean\+a(?:hum_|hu|haA|humaA|[  ])))     #identifies "'anna" or "qaala" as separators only if they are not followed by a typical isnad element in the next 100 characters
    )
    (?P<mat>.*?)$      # identifies this part of text as matn
    """,
                re.VERBOSE | re.MULTILINE)

# ===Writes on an output file in xml code   various extracted info, ===
er1 = 0 #errors' counters for prompt information
er2 = 0
id_ar = 0
l_typ = [] #serve solo per la lista delle tipologie di trasmissione
for had in q :
    had = had.replace('\r',' ')
    had = had.replace('\n',' ')
    id_ar +=1
    m = p1.search(had)
    if m:
        nn = m.group('num')
        f.write('\t' + r'<hadith id_ar="'+ str(id_ar) + r'" ' + r'id_cor=""' + '>' + '\n')
        f.write('\t\t' + '<source_info>' + '\n')
        f.write  ('\t\t\t' +'<vol>' + str(volume_number(nn)) + '</vol>' + '\n')
        f.write ('\t\t\t' +'<book>' + '</book>' + '\n')
        f.write ('\t\t\t' +'<num>' + str(nn) + '</num>' + '\n')
        f.write('\t\t' + r'</source_info>' + '\n')
        f.write('\t\t' + r'<isn>' + '\n')

        fTR.write('\t' + r'<hadith id_ar="'+ str(id_ar) + r'" ' + r'id_cor=""' + '>' + '\n') #output traslitterato
```

```
        fTR.write('\t\t' + '<source_info>' + '\n')#output traslitte-
rato
        fTR.write  ('\t\t\t'  +'<vol>'  +  str(volume_number(nn))  +
'</vol>' + '\n')#output traslitterato
        fTR.write  ('\t\t\t'  +'<book>'  +  '</book>'  +  '\n')#output
traslitterato
        fTR.write  ('\t\t\t'  +'<num>'  +  str(nn)  +  '</num>'  +
'\n')#output traslitterato
        fTR.write('\t\t' + r'</source_info>' + '\n')#output traslit-
terato
        fTR.write('\t\t' + r'<isn>' + '\n')#output traslitterato

        isn = m.group('isn')

#da qui parte la segmentazione dell'isnad

        p_isn = re.compile (r"""
        (
         qaAla\sHad\+aCan(?:iy|aA)|
         qaAla\ssamic_tu|
         qaAla\sEax_baran(?:iy|aA)|
         qaAla\sEan_baEan(?:iy|aA)|
         yaquwlu\ssamic_tu|
         Ean\+ahu\ssamica|
         Eax_barahu\sEan\+a|
         (?:y|t)aquwlu|
         Had\+aCan(?:iy|aA)|
         can_\s|
         samic_(?:tu|a)|
         (?:wa)*Eax_bara(?:niy|naA|hu)|
         Ean_baEan(?:iy|aA)|
         qaAla|
         Ean\+a

        )
        (.*?)
        (?=qaAla\sHad\+aCan(?:iy|aA)|
         qaAla\ssamic_tu|
         qaAla\sEax_baran(?:iy|aA)|
         qaAla\sEan_baEan(?:iy|aA)|
         yaquwlu\ssamic_tu|
         Ean\+ahu\ssamica|
         Eax_barahu\sEan\+a|
         (?:y|t)aquwlu|
         Had\+aCan(?:iy|aA)|
```

```
         can_\s|
         samic_(?:tu|a)|
         (?:wa)*Eax_bara(?:niy|naA|hu)|
         Ean_baEan(?:iy|aA)|
         qaAla|
         Ean\+a|
         $)

        """,
                re.VERBOSE | re.MULTILINE)

        isnad_segmented = p_isn.findall(isn)

        for segment in isnad_segmented :
            f.write('\t\t\t' + r'<trasm type="' + conv(segment[0]) +
'">' + conv(segment[1])  +r'</trasm>' + '\n') # output in arabo
            fTR.write('\t\t\t'  +  r'<trasm  type="'  +  segment[0]  +
'">' + segment[1]  +r'</trasm>' + '\n') # output traslitterato
            l_typ.append(segment[0]) #serve solo per la lista delle
tipologie di trasmissione
            cont_isn +=1

        f.write('\t\t' + r'</isn>' + '\n')
        fTR.write('\t\t' + r'</isn>' + '\n')# output traslitterato
        f.write('\t\t' + r'<sep>' + conv(m.group('sep')) + r'</sep>'
+ '\n') # output in arabo
        fTR.write('\t\t' + r'<sep>' + m.group('sep') + r'</sep>' +
'\n') # output traslitterato
        f.write('\t\t' + r'<mat>' + conv(m.group('mat')) + r'</mat>'
+ '\n')# output in arabo
        fTR.write('\t\t' + r'<mat>' + m.group('mat') + r'</mat>' +
'\n')# output traslitterato
        cont_mat +=1
        f.write('\t' + r'</hadith>' + '\n')
        fTR.write('\t' + r'</hadith>' + '\n')# output traslitterato

    else : ##stampa uno per uno gli hadit non segmentati
##              f.write('\t'  +  '<hadith  id_ar="'+  str(id_ar)  +  '"
id_cor="">' + '\n')
##         f.write(had)
##         f.write('\n\t' + r'</hadith>' + '\n')
##         print('HADITH NOT SEGMENTED: \n')
##         print(had + '\n\n')
        er2 += 1
```

```
f.write('</HADITH_EXTRACTED>\n')
fTR.write('</HADITH_EXTRACTED>\n')# output traslitterato
print('\n========Matn errati: ' + str(er2) + '; Isnad errati: ' +
str(er1) + '==========\n')
f.close()
fTR.close()

##y= [it.replace(' qaAla','') for it in trasmList]
##
##x    =    [lista_trasm.write(item    +    '\n')    for    item    in
list(sorted(set(y)))]
##lista_trasm.close()
print ('\nNumero matn segmentati: ' + str(cont_mat) +'\n')
print ('\nNumero trasmettitori segmentati: ' + str(cont_isn)+ '\n')


##==========trova lista tipologie di trasmissione
f = codecs.open('typologies.txt','w', encoding='utf-8')
for el in sorted(set(l_typ)):
    f.write(el + '\n')
f.close()
```

A.2 *HadExtractor* (per la traduzione inglese)

```
'''HadExtractor(ENGLISH) - Draft di programma  per la segmentazione
tramite RE di hadith
marco.boella@alice.it
Versione 0.1  - 26 gennaio 2011'''
import codecs, re
# ===operazioni su files input output===
f = codecs.open('buck_EN.txt',encoding = '1256')
text = f.read()
f.close()
f = codecs.open('HadExtractedENGLISH.xml','w', encoding='utf-8')
f.write('<?xml version="1.0"?>\n')
f.write('<HADITH_ENGLISH>\n')

p2 = re.compile(r'\s*?(?=Volume+ *\d+?)')
had_all = p2.split(text)
```

```
#re per segmentare gli hadith:
p3 = re.compile(r"""
          ^.*Volume\s*(?P<vol_en>\d+?) #numero volume
          ,*\s*?
          Book\s*(?P<book_en>\d+?) #numero libro
          ,*?\s*?
          Number\s*(?P<num_en>\d+[a-z]?) #numero hadith
          :\s+?
          '*Narrate(d|s|\s)\s?
          (?P<last_trasm_en>.*?)   #ultimo trasmettitore
          ((:\s\n)|(:\s)|\n)
          (?P<mat_en>.*)$  #matn
          """,
                re.VERBOSE | re.MULTILINE)
#re per gli hadith NON segmentabili(14 occorrenze):
p4 = re.compile(r"""
          ^Volume\s*(?P<vol_en>\d+?) #numero volume
          ,*\s*?
          Book\s*(?P<book_en>\d+?) #numero libro
          ,*?\s*?
          Number\s*(?P<num_en>\d+[a-z]?) #numero hadith
          :\s*?
          (?P<mat_full>.*)$  #matn NON SEGMENTATO
          """,
                re.VERBOSE | re.MULTILINE | re.DOTALL)
id_en = 0
for line in had_all :
    id_en +=1
    s_had = p3.search(line)
    if s_had:
        f.write('\t' + r'<hadith_en id_en="' + str(id_en) + r'" ' +
r'id_cor="' + '>' + '\n')
        f.write('\t\t' + '<source_info>' + '\n')
        f.write  ('\t\t\t'  +'<vol>'  +  s_had.group(r'vol_en')  +
'</vol>' + '\n')
        f.write  ('\t\t\t'  +'<book>'  +  s_had.group('book_en')  +
'</book>' + '\n')
        f.write  ('\t\t\t'  +'<num>'  +  s_had.group('num_en')  +
'</num>' + '\n')
        f.write('\t\t' + r'</source_info>' + '\n')
        f.write('\t\t'          +          r'<last_trasm>'          +
s_had.group(r'last_trasm_en') + r'</last_trasm>' + '\n')
        f.write('\t\t'  +  r'<mat_en>'  +  s_had.group(r'mat_en')  +
r'</mat_en>' + '\n')
        f.write('\t' + r'</hadith_en>' + '\n')
```

```
    else :
        alt_had = p4.search(line)
        if alt_had:
            f.write('\t' + r'<hadith_en id_en="' + str(id_en) + r'"
' + r'id_cor="" + '>' + '\n')
            f.write('\t\t' + '<source_info>' + '\n')
            f.write ('\t\t\t' +'<vol>' + alt_had.group(r'vol_en') +
'</vol>' + '\n')
            f.write ('\t\t\t' +'<book>' + alt_had.group('book_en') +
'</book>' + '\n')
            f.write ('\t\t\t' +'<num>' + alt_had.group('num_en') +
'</num>' + '\n')
            f.write('\t\t' + r'</source_info>' + '\n')
            f.write('\t\t' + r'<last_trasm>' + r'NOT SEGMENTED--SEE
MATN' + r'</last_trasm>' + '\n')
            f.write('\t\t'          +          r'<mat_en>'          +
alt_had.group(r'mat_full') + r'</mat_en>' + '\n')
            f.write('\t' + r'</hadith_en>' + '\n')
        else :
            print line

f.write('</HADITH_ENGLISH>\n')
f.close()
```

A.3 *ChainViewer*

```
#ChainViewer 0.1
#Marco Boella<marco.boella@alice.it> - 21 may 2012

# Import codecs and RE
import codecs, re

# Import graphviz
import sys
sys.path.append('..')
sys.path.append('/usr/lib/graphviz/python/')
sys.path.append('/usr/lib64/graphviz/python/')
import gv

# Import pygraph
from pygraph.classes.graph import graph
from pygraph.classes.digraph import digraph
from pygraph.algorithms.searching import breadth_first_search
from pygraph.readwrite.dot import write

# Input and output files' operations
```

```
f = codecs.open('HadExtractedPROGR.xml',encoding = 'utf-8')
##f = codecs.open('HadExtractedONE.xml',encoding = 'utf-8')
##f = codecs.open('HadExtractedTWO.xml',encoding = 'utf-8')

indata = f.read()
f.close()

#Get data from HadExtracted.xml (it builds a nested list with id_ar,
type, trasm
reg_ex1 = re.compile(r'\s+?(?=<hadith id)')
indata_list = reg_ex1.split(indata)
reg_ex2 = re.compile(r'<hadith id_ar="(?P<id_ar>\d+?)"')
reg_ex3                    =                    re.compile(r'<trasm
type="(?P<type>.+?)">(?P<trasm>.+?)</trasm>?')
final_list=[]
for line in indata_list [1:] :
    m = reg_ex2.search(line)
    id_ar = m.group('id_ar')

    extrT = reg_ex3.findall(line)
    n_list = []
    for elem in extrT:
        eleml = list(elem)
        n_list.append(eleml)
    prov_list =[id_ar, n_list]
    final_list.append(prov_list)

# Graph creation
gr = graph()

# Extract from nested list a list of unique transmitter's names
trasm_list = []
for ntytr in final_list:
    for tytr in ntytr[1:]:
        for tr in tytr:
           trasm_list.append(tr[1])
trasm_list= set(trasm_list)


# Add nodes
gr.add_nodes(["MATN", "COMPILER"])
gr.add_nodes (trasm_list)

# Extract from nested list a list of transmitter's names and type of
transmission
tytr_list=[]
for ntytr in final_list:
    for tytr in ntytr[1:]:
        tytr_list.append(tytr)
edges_list=[]
##print tytr_list[0]
```

```
for elem in tytr_list:
    first_edge = ("COMPILER",elem[0][1], elem[0][0])
    edges_list.append(first_edge)
    n = 1
    le = len(elem)
    while n < le:
        internal_edge = (elem[n-1][1], elem[n][1], elem[n][0])
        edges_list.append(internal_edge)
        n+=1
    last_edge = (elem [n-1][1],"MATN","-")
    edges_list.append(last_edge)
edges_list= list(set(edges_list))

##for tytr in elem[1:]:
##        internal_edges = [elem????,tytr[1],tytr[0]] #da qui
##print len(edges_list)
##set(edges_list)
##print edges_list
##print


#Add edges
for edge in edges_list:
    gr.add_edge((edge[0], edge[1]), label= edge[2])

#Print IDLE info
print '===========Running Chain viewer:============'
print 'Total of analised hadith: ' + str(len(final_list))
print    'Total    of    transmitters    involved    (NODES):    '    +
str(len(trasm_list))
print 'Total of relations (EDGES): ' + str(len(edges_list))
print
print 'Creating chains\'graphs...'


# Draw as PNG
dot = write(gr)
gvv = gv.readstring(dot)
gv.layout(gvv,'dot') #others: dot neato fdp sfdp twopi circo
gv.render(gvv,'png','provamarco.png')

# Then, draw the breadth first search spanning tree rooted in
Compilator
st, order = breadth_first_search(gr, root="MATN")
gst = digraph()
gst.add_spanning_tree(st)

dot = write(gst)
gvv = gv.readstring(dot)
```

```
gv.layout(gvv,'dot')
gv.render(gvv,'png','provamarco1.png')

print 'Graphs created.'
print
print 'Featuring...'
##trasm_list = list(trasm_list)
##trasm_list.sort()

for name in trasm_list:
    print name
```

A.4 *CrossQuran*

```
import codecs, re

def no_lines(a):
    a=a.replace('\n','')
    return a

def RE_protection(b):
    b=b.replace(r"+",r"\+")

    return b

def read_file(filen):
    f = codecs.open(filen, encoding="utf-8")
    t = f.read()
    f.close()
    return t


def split_in_periods(fulltext):
    q=fulltext.split('.')
    return q

def split_in_words(fulltext):
    q = fulltext.split()
    return q

def find_ref(patterns, text):
    ref_increm = []
    pat_var = r'%s'
    n = 0
    x = patterns[0]
```

```
    for pattern in patterns:

        p = re.compile(pat_var % pattern)
        found_ref = p.search(text)
        if found_ref:
            ref_increm.append(pattern)

    return ref_increm

##def find_ngrams(patterns, text):
##    n = 0
##    x = patterns[n]
##    pat_var = r'%s'
##
##    if  x re.compile


def print_results(text_to_find,found,text_to_investigate):
    print 'numero versetti token: ', len(text_to_find)
    print 'numero versetti type: ', len(set(text_to_find))
    print 'Corrispondenze trovate: ', len(found)
    print        '%        Corano        in        Hadith:        ',
float(len(''.join(found))*100)/float(len(text_to_investigate))
    for res in sorted(found): print res

##=========================

textA = read_file('quran_trans.txt')
textB = read_file('matn.txt')
textA = no_lines(textA)
textA = RE_protection(textA)
####sent = split_in_periods(textA)
textAtokens = split_in_words2(textA)
print textAtokens[:100]
##matches = find_ref(sent,matn)
##print_results(sent,matches,matn)
```

Ringraziamenti

Ringrazio il prof. Giuliano Lancioni che è stato per me un eccellente tutor e un mentore prezioso, geniale nel saper bilanciare insieme dotte conoscenze arabistiche linguistiche e informatiche. A lui devo, oltre a molte altre intuizioni, l'idea su cui molta parte di questa tesi è basata, vale a dire la possibilità di interpretare computazionalmente la struttura originaria dei *ḥadīṯ*.

Un ringraziamento anche al prof. Leonardo Capezzone per aver coordinato il mio curriculum di dottorato con passione e intelligenza, e per avermi suggerito l'idea della lettura anacronistica di una collezione di *ḥadīṯ* come base di dati.

Un pensiero particolare va al compianto Djamel Khouloughli, studioso di grammatici arabi ma anche uno dei pionieri della linguistica computazionale per l'arabo Una delle poche persone che si emozionava con me e non si addormentava a parlare di espressioni regolari.

Questo lavoro mi ha dato più occasioni per conoscere e apprezzare il prof. Arieh Levin, per il suo immenso valore intellettuale, ma anche per l'affetto e l'amicizia dimostratimi.

Un ringraziamento speciale a Marino, che mi è sempre stato accanto e mi ha aiutato moltissimo pungolandomi intellettualmente con la sua mente di scienziato puro.

Ringrazio infine i miei genitori e gli amici più cari, Enrica, Luisa, Anna, Alessio e tutti gli altri per la gioia, la vicinanza e anche il divertimento che mi hanno donato in questi anni.

Marburg, 3 maggio 2013