Umanistica Digitale – ISSN: 2532-8816 – n. 13, 2022
R. Del Gratta, A. Del Grosso, S. Zenzaro, F. Boschetti, L. Bambaci – *La filologia come sistema dinamico*
DOI: http://doi.org/10.6092/issn.2532-8816/13684

# La Filologia come sistema dinamico

[1]Riccardo Del Gratta, [2]Angelo Mario Del Grosso, [3]Simone Zenzaro, [4]Federico Boschetti, [5]Luigi Bambaci

[1,2,3,4]Istituto di Linguistica Computazionale «A. Zampolli», CNR, Pisa, Italia
[5]Università di Bologna, Bologna, Italia
[1]riccardo.delgratta@ilc.cnr.it
[2]angelo.delgrossoi@ilc.cnr.it
[3]simone.zenzaro@ilc.cnr.it
[4]federico.boschetti@ilc.cnr.it
[5]luigi.bambaci2@unibo.it

## Abstract

Introduciamo un approccio formale all'evoluzione del contenuto informativo veicolato da documenti umanistici, con particolare attenzione alla prospettiva filologica e alle problematiche tipiche ad essa connesse (studio della tradizione, confronto tra testimoni, selezione e scelta delle lezioni, edizione di un testo, etc). Proponiamo un modello matematico in grado di formalizzare diversi fenomeni complessi in vari ambiti di ricerca quali la Linguistica Computazionale, la Filologia Digitale e l'Ingegneria del Software, soprattutto quando questi vengono applicati all'analisi di documenti e testi di interesse storico-letterario.

In this article we introduce a formal approach to the evolution of documents with particular attention to the philological perspective and the typical related issues. We propose a mathematical model capable of formalizing various complex phenomena in various research fields such as Computational Linguistics, Digital Philology and Software Engineering, in particular when this is applied to the analysis of documents and texts of historical and literary interest.

## Introduzione

L'inizio della riflessione teorica sul metodo della filologia testuale (o critica testuale) risale al XVIII secolo ([35];[25]). Furono i filologi di quest'epoca, in particolare classici e del Nuovo Testamento, i primi a riconoscere la necessità di condurre un'indagine rigorosa e sistematica della documentazione manoscritta (recensio), prima di procedere alla normale attività di correzione (emendatio) che precede alla pubblicazione dei testi antichi.

La distinzione ottocentesca tra recensio ed emendatio ha determinato, com'è noto, una vera e propria "rifondazione scientifica" del metodo: prima di allora, l'attività editoriale era guidata quasi esclusivamente dai criteri cosiddetti "interni", come lo stile dell'autore e il "senso della lingua" (Sprachgefühle) dell'editore; solo in seguito, e grazie a tale distinzione, venne affermandosi l'importanza dei criteri "esterni", quali la datazione dei testimoni testuali (manoscritti, edizioni a stampa), la loro provenienza geografica e, soprattutto, la loro genealogia.

Il riconoscimento dell'esistenza di relazioni genealogiche tra testimoni testuali – e l'idea che queste possano essere ricostruite mediante l'applicazione di regole formali – è stato determinante per la fondazione della filologia come scienza moderna, e il suo posizionamento come disciplina accademica autonoma nel dominio delle scienze storiche. Il culmine di tale sviluppo coincise, di fatto, con la formazione del metodo del Lachmann o metodo genealogico-testuale, forse il più noto in filologia e certamente il più avanzato quanto alla definizione di procedure operative formali.

Lo sviluppo del metodo genealogico in filologia si svolse in parallelo all'affermazione della teoria dell'evoluzione darwiniana ([29]). Secondo tale metodo, la trasmissione di un'opera letteraria o storica può essere concepita, in effetti, in termini molto simili a quelli di "descent with modification" immaginati da Darwin per l'evoluzione degli organismi viventi: così come questi ereditano i tratti dai rispettivi antenati comuni e ne introducono di nuovi passando da una generazione alla successiva, così anche i testi, passando da una copia all'altra, si fanno portatori di variazioni ereditarie introdotte di volta in volta dagli attori del processo di trasmissione. Studiando le differenze (varianti) che intercorrono tra una copia e l'altra e, in particolare, seguendo le tracce (gli errori) lasciate dagli scribi o copisti nel corso di tale processo, il metodo genealogico promette di ricostruire – e di rappresentare mediante un grafo ad albero, lo stemma codicum – lo schema delle relazioni genealogiche, e di approssimarsi, grazie a questo, al cosiddetto Originale o Ur-text.

I principi del metodo ricostruttivo, formalizzati per la prima volta in modo rigoroso dal Maas ([27]), andarono presto incontro, com'è noto, ad aspre critiche demolitrici. Queste presero di mira tanto le procedure – il modello evolutivo ad albero – quanto i presupposti stessi del

metodo – l'esistenza, tipicamente romantica, di un solo originale d'autore e l'idea, forse positivistica, che questo sia per noi realmente attingibile.

La sfiducia nei confronti del metodo genealogico, da un lato, ha portato al sorgere di nuove correnti di pensiero ecdotico, come il bédierismo ([20]), disinteressate alla filologia ricostruttiva e anzi avverse a qualunque tentativo di sistematizzazione formale; dall'altro, in risposta alle criticità del metodo tradizionale che il dibattito teorico aveva nel frattempo messo in luce, ha promosso lo sviluppo di altre tecniche di indagine testuale, basate su criteri distribuzionali ([28]) o sui formalismi della logica algebrica ([40]). Tali tecniche, accomunate dalla critica al "soggettivismo" tipico delle valutazioni qualitative della ricerca umanistica, e improntate a un approccio più marcatamente quantitativo e "oggettivo", hanno senza dubbio contribuito al processo di formalizzazione del metodo, preparando il terreno agli sviluppi della disciplina successivi all'introduzione del personal computer ([23]; [35]; [22]).

È soprattutto grazie a questo, per concludere, che in anni più recenti si è assistito a un rinnovato interesse per la stemmatologia e, più in generale, per lo studio delle tradizioni testuali attraverso metodi formali.

Tra gli indirizzi di ricerca più feraci si ricorda in particolare quello della "critica testuale cladistica" ("cladistische tekstkritiek" [34], parzialmente tradotto in inglese in [33] pp. 60-70), ovvero dell'analisi di tradizioni testuali mediante algoritmi e software della bioinformatica. Le affinità menzionate sopra tra evoluzione degli organismi viventi e trasmissione dei testi e la riscoperta del background comune alle due discipline hanno gettato le basi, almeno a partire dai primi anni '90, per un'intensa collaborazione tra filologi e biologi, generando una vasta letteratura di studi filogenetici di tradizioni manoscritte ([26]; [30]; [17]; [31]).

Contemporaneamente alle applicazioni bioinformatiche, sono stati sviluppati altri metodi computazionali, per così dire, più direttamente filologici. Alcuni di questi sono tagliati su specifici casi di studio, come la tradizione del Nuovo Testamento ([36] [24], altri mirano a un'applicabilità più vasta ([36]; [37]) e sono disponibili anche sotto forma di pacchetti software ([21]).

In questo articolo introduciamo un approccio formale all'analisi e alla trasmissione del testo considerando prevalentemente gli aspetti di indagine filologica sopra introdotti. In effetti, non si propone una nuova definizione di documento elettronico, ma piuttosto una formalizzazione di un (meta)modello per lo studio della tradizione di un testo veicolato da molteplici documenti (testimoni) con particolare attenzione alle operazioni che introducono errori e varianti modellandone la dinamica.

In particolare, tenteremo di formalizzare il fenomeno della variazione del contenuto informativo nel tempo attraverso un approccio di tipo evoluzionistico, dove per tale approccio

intendiamo esattamente il processo dinamico che lega un documento (inteso come contenitore) a un altro, o che lega due differenti versioni dello stesso documento.

Prima di arrivare alla descrizione dell'approccio evoluzionistico forniamo, di seguito, alcune utili definizioni.

### Un Sistema dinamico, in breve

L'aggettivo dinamico è connesso al concetto di forza in Fisica. Per forza si può intendere una qualsiasi azione un agente esterno applichi al sistema osservato. In effetti, generalmente, si contrappone dinamico a statico, volendo enfatizzare proprio che le caratteristiche - e quindi lo stato - di un sistema cambiano nel tempo invece di restare costanti.

In secondo luogo, dinamico è connesso al termine dinamica, una parte della Fisica che studia il "moto" degli oggetti soggetti a forze (azioni) esterne. Abbiamo messo il "moto" volutamente tra apici, perché un sistema può cambiare anche rimanendo fermo. Si pensi, per esempio, ad una pallina di gomma perfettamente sferica che viene stretta tra due dita. La forma della pallina cambia: da perfettamente sferica a ellissoidale, più lunga che larga o viceversa.

Parlare di evoluzione dinamica di un sistema non significa necessariamente riferirsi alla variazione di posizione nello spazio degli oggetti del sistema, piuttosto all'osservazione dei cambiamenti di una (o più) proprietà. Queste variazioni vengono descritte in termini di "cambiamento di stato del sistema" denotando l'effetto che la variazione introduce nella configurazione iniziale dello stesso.

Il problema principale della dinamica risiede quindi nell'essere in grado di conoscere le forze (azioni) esterne per poter determinare il cambiamento di stato in modo deterministico.

Newton stesso, nel suo Philosophiae Naturalis Principia Mathematica, afferma[1,2] che se si conoscessero tutte le forze e tutti i dettagli di un sistema ad un dato istante iniziale, che denominiamo $t_0$, allora si conoscerebbe il sistema nel suo complesso a qualsiasi altro istante $t$.

Riprendendo un classico esempio, se di due palline da biliardo conoscessimo ogni proprietà (massa, velocità, direzione), allora sarebbe possibile conoscere con certezza cosa avviene di ogni singola pallina (e del sistema complessivo) dopo il loro urto.

Occorre inoltre notare che il presupposto essenziale di quanto precedentemente introdotto risieda (in ambito classico) nella descrizione di un sistema in ogni sua singola parte, ovvero che un dato sistema $S$ possa essere considerato la somma di tanti sottosistemi $s_i$.

1 Si fornisce una definizione volutamente non rigorosa.

2 Per ulteriori applicazioni di matematica complessa a fenomeni linguistici si veda [5], [6].

Ne deriva che la dinamica del sistema complessivo sia data dalla somma delle dinamiche dei sottosistemi:

$$sistema_{dopo} = X\left(sistema_{prima}\right) \quad (1)$$

Il senso della (1), intuitivamente è il seguente: se $X$ è un insieme di azioni esterne (di cui sappiamo tutto) che agiscono su un certo sistema (testi, lessici etc) che conosciamo prima dell'azione, allora possiamo avere informazioni sul sistema dopo aver applicato le azioni definite in $X$. Un sistema di questo tipo è un sistema deterministico.

Come meglio definito nella sezione seguente, faremo riferimento ad alcune entità utili alla descrizione di alcuni aspetti della Filologia (ma anche della Linguistica Computazionale) come un sistema dinamico.

## Un modello formale per l'evoluzione dei documenti

Per poter dare un'idea più precisa di sistema dinamico nell'ambito della filologia, occorre definire alcuni concetti preliminari. Innanzitutto vorremmo soffermarci su alcune motivazioni che ci spingono a credere che tale formalizzazione possa contribuire ad una maggiore comprensione delle entità, delle relazioni e dei processi alla base della filologia.

Definire la natura di un documento è un compito arduo e diversi approcci alla realizzazione di modelli hanno cercato di darne una interpretazione valida e generale ([9]-[12]; [14]). Per quanto riguarda i documenti digitali in ([8]) viene riportata l'evoluzione del paradigma di documento. In quel contesto si possono individuare tre modelli principali: sequenza di caratteri (stringhe), generalized markup (rappresentazione gerarchica), e tramite Resource Description Language (RDF - modello reticolare).

Il modello maggiormente adottato, definito da un linguaggio di markup, può essere generalizzato da un modello ad albero in cui la struttura gerarchica ha un ruolo preponderante nella definizione del documento stesso. A questo modello appartengono le linee guida TEI, con la loro definizione basata su XML, e l'implementazione del formato di rappresentazione delle pagine web ad oggetti, che è appunto chiamato Document Object Model (DOM). Proprio lo standard W3C per il DOM[3] definisce un documento come "la radice dell'albero del documento",[4] denotando la natura gerarchica dei dati che rappresenta. La struttura denotata

3 https://www.w3.org/TR/DOM-Level-3-Core/core.html

4 "It is the root of the document tree", cfr. https://www.w3.org/TR/DOM-Level-3-Core/core.html#i-Document

tramite RDF, invece, inferisce una descrizione in termini di grafo e generalizza il concetto di relazione che nel modello ad albero si limita alla relazione padre-figlio.

A questi modelli si affiancano quelli basati su tabelle come, ad esempio, la struttura delle basi di dati relazionali o quelle che definiscono indici di ricerca (ad esempio i documenti definiti dagli inverted index[5]).

Ancora altri esempi di modellazione procedono a partire da un contesto di semiotica e con una prospettiva interdisciplinare ([12]).

La presenza di un'ampia varietà di modelli per i documenti, da un lato denota la necessità di comprendere appieno la loro natura, dall'altra la difficoltà di trovare un modello che sia trasversale al dominio di riferimento dell'ambito di studio o del campo di applicazione. Guardando il problema da un altro punto di vista, potremmo concentrarci sull'osservazione e sullo studio dell'evoluzione di uno o più documenti astraendo dalle specificità delle possibili rappresentazioni. In questo senso possiamo pensare alla differenza che esiste fra la descrizione di una struttura dati in informatica e la definizione di un abstract data type (ADT). Così come è possibile ragionare sul comportamento e sulle proprietà di un tipo di dato in maniera astratta, è possibile ragionare sull'evoluzione di un documento indipendentemente dalla sua rappresentazione fisica o digitale ([18]).

Per questo motivo proponiamo una definizione matematica di documento sulla quale siano definite delle operazioni. Pur astraendo dai dettagli di rappresentazione, la definizione di documento che proponiamo è istanziabile in modelli concreti ed in particolare nei modelli di documento sopracitati. Nei fatti, la scelta del modello concreto di rappresentazione del documento dipenderà dal contesto di utilizzo dello stesso.

Il livello di astrazione del modello qui proposto permette inoltre di definire in maniera formale e non ambigua concetti di filologia quali le varianti, la ricostruzione delle relazioni fra le fonti, etc. Questa formalizzazione è utile in Filologia Digitale e Computazionale in quanto, oltre a fornire uno strumento utilizzabile a livello descrittivo, risulta eseguibile.

In questa fase, senza mancare di generalità, ci limiteremo alla classe dei documenti testuali digitali.

Assumiamo che il documento $D$ sia composto da tre parti, un contenuto informativo (che identifichiamo con $c$), un formato ($f$) che specifica come il contenuto informativo è disposto nel documento e un insieme di ulteriori informazioni, che chiamiamo informazioni paratestuali e che indichiamo con $\{p_1, \ldots p_k\}$ (e che abbreviamo con $\{p\}$).

5 I modelli definiti su schemi relazionali oppure su rappresentazioni di documenti adottati in ambito di text-retrieval denotano una centralità del dato (struttura della forma del contenuto) rispetto alla duale rappresentazione della struttura dell'espressione documento-centrica.

Un documento (testuale digitale) $D$ è una tripla $c, f, \{p\}$:

$$D = D(c, f, \{p\}) \quad (2)$$

In linea di principio queste tre componenti possono dipendere l'una dall'altra, oppure essere completamente identificabili e separabili. Per cui, tre ulteriori possibili definizioni di $D$, e che seguono dalla (2), sono le seguenti:

$$D = D(c(f), f, \{p\}) \quad (3)$$

$$D = D \quad (4)$$

$$D = D(c, f, \{p\}) = c \times f \times \{p\} \quad (5)$$

Un documento XML che riporti una formattazione inline è un esempio di (3). Infatti, si consideri il termine Roma il cui corrispondente frammento XML può (ad esempio) essere Roma, dove il tag  indica che il termine Roma va reso corsivo. Il contenuto informativo Roma dipende quindi sia dalla sequenza di caratteri Roma che dalla sua resa grafica.

Analogamente una pagina web veicola informazioni ($c$) sia attraverso font specifiche, tipo grassetto e italico ($f$), che grazie a note addizionali presenti in altre aree della pagina (ad esempio note a piè di pagina o laterali) ($\{p\}$).

Le pagine web sono dunque validi esempi di (4).

Infine, un testo che riporti in modalità stand-off la formattazione di Roma è un esempio di (5). Per rendersene conto basti considerare $f$ come lo schema (XML) del documento, che può apparire come segue:

<w id="1">Roma</w> (Livello di contenuto -c-)

<id="1">Corsivo</id> (livello paratestuale -{p}-)

Il livello di contenuto veicola il testo, mentre il livello paratestuale aggiunge informazioni sulla formattazione. La differenza sostanziale in questi due esempi è che, se volessimo passare dall'italico al grassetto, dovremmo sostituire i tag direttamente nel testo ($c$) per l'esempio relativo a (3), mentre per l'esempio in (5) è sufficiente cambiare in $\{p\}$. Questo aspetto è fondamentale quando introduciamo il concetto di separabilità e definiamo le operazioni sui documenti.

Il documento definito in (5) si definisce "separabile". Separabile indica che per descrivere $D$ occorre descrivere le sue parti. L'analogia più calzante è quella con i punti di un piano Cartesiano. Un punto $A$ in un piano cartesiano è descritto univocamente dalle sue coordinate ($X_A$,$Y_A$) ovvero $A \equiv \left( x_A , y_A \right)$.

Quindi, per identificare (descrivere) il punto A è sufficiente fornire le sue coordinate.

In un certo senso, nei documenti cartesiani o separabili, c, f e {p} sono tre componenti indipendenti. Come definito da (5), $D$ è identificato dalle proprie parti, esattamente come un punto in un piano Cartesiano lo è dalle proprie coordinate.

Se assumiamo nella (2) che $D$ non abbia contenuto informativo e informazioni paratestuali, ovvero l'insieme di tali informazioni è l'insieme vuoto $p = \emptyset$ , otteniamo il documento vuoto $D_e$:

$$D_e = D_e \left( c_e , f , \emptyset \right) \text{ (6)}$$

In $D_e$ la sola componente presente è il formato $f$ che è lasciato sotto specificato. File testuali creati con un qualsiasi strumento di editing testuale (come notepad oppure vim) e salvati senza inserire alcun carattere sono esempi di documenti vuoti.

Un'ulteriore sotto-definizione importante è quella di documento esteso. Un documento esteso è un documento "accompagnato" da altre informazioni ($m$) non necessariamente paratestuali e che non aggiungono niente al contenuto informativo del documento. Ad esempio, il periodo in cui un documento è stato scritto, oppure la lingua e, nel caso di documenti digitali, $\{ m \}$ può contenere, il nome del file e dove risiede fisicamente nella memoria del computer:

$$D_{ex} = D_{ex} \left( c , f , \backslash\backslash \{ p \backslash\backslash \} , \backslash\backslash \{ m \backslash\backslash \} \right) \text{ (7)}$$

Senza perdere di generalità possiamo supporre $D_{ex}$ separabile e scrivere la (7) come:

$$D_{ex} = D \left( c , f , \backslash\backslash \{ p \backslash\backslash \} \right) \times \backslash\backslash \{ m \backslash\backslash \} \text{ (8)}$$

La seconda definizione che presentiamo è quella di azione o operazione. Una operazione $op$ è "l'elaborazione di un insieme di documenti che restituisce un singolo documento".[6] Formalmente possiamo scrivere:

$$op \left( \backslash\backslash \{ D_1 , D_2 , \ldots , D_n \backslash\backslash \} \right) = \backslash\backslash \{ D_a \backslash\backslash \} \text{ (9)}$$

Il significato della (9) è il seguente: $op$ agisce su n documenti in input e crea un nuovo documento, $D_a$, in output.

6 Ovvero un insieme con un solo elemento.

In caso di documenti cartesiani, possiamo supporre che anche le operazioni definite sui documenti siano cartesiane:

$$op = o\,p_c \times o\,p_f \times o\,p_{\backslash\backslash\{p\backslash\backslash\}} \quad (10)$$

in modo che operazioni specifiche agiscano solo sulla componente di $D$

di competenza: $o\,p_c$ solo sul contenuto informativo, $o\,p_f$ sul formato e $o\,p_{\backslash\backslash\{p\backslash\backslash\}}$ sul paratesto, lasciando le altre componenti inalterate.[7]

Se torniamo all'esempio del file XML con formattazione stand-off, il cambio di valore da Corsivo a **Grassetto** nella parte paratestuale è il risultato di una operazione limitata alla parte paratestuale di un documento cartesiano, che è appunto $o\,p_{\backslash\backslash\{p\backslash\backslash\}}$.

Occorre aggiungere che il nostro approccio sarà quello di considerare $D_a$ sempre un documento nuovo quando c'è una variazione in almeno una delle tre componenti di $D$.

Consideriamo $D_0$ un documento in formato plain text; supponiamo poi che il contenuto di $D_0$ consti di un insieme di frasi. Se $op$ è, per esempio, l'aggiunta di una frase, denotiamo $D_1$ come $op\left(D_0\right) = D_1$ il documento risultante dall'aggiunta di una frase a $D_0$. Anche se $D_1$ fosse lo stesso file,[8] per il modello proposto $D_0$ e $D_1$sono logicamente due entità distinte che indicano lo stato del documento prima e dopo l'applicazione dell'operazione di aggiunta. Questo giustifica la frase a inizio paragrafo.

È necessario a questo punto introdurre anche il concetto di operazione identità $I_d$:

$$I_d : D \rightarrow D \quad (11)$$

Sia l'operazione identità che il documento vuoto sono necessari a rendere il modello formale proposto chiuso rispetto alle possibili operazioni che si possono effettuare. L'operazione identità denota intuitivamente il "non fare nessuna operazione", mentre il documento vuoto $D_e$ è necessario per coprire i casi in cui il risultato di una operazione è un documento vuoto.

7 Questo aspetto è molto importante all'interno delle cosiddette "teorie dei processi [19]"; ma lo è anche all'interno dell'ingegneria del software. A parità di scopo, un software progettato per gestire documenti cartesiani differisce da uno nato per gestire documenti non cartesiani.

8 Ovvio in caso di file testuali presenti su pc.

Per esempio, se $D_i$ contiene un testo italiano e $o\,p_i$ è "estrai le parole non italiane", è ovvio che il risultato di tale operazione sia un documento vuoto: $o\,p_i : D_i \rightarrow D_e$.

Un altro importante aspetto relativo all'insieme delle operazioni ammissibili, riguarda gli agenti che le eseguono. Essendo azioni astratte, le operazioni vengono rese possibili da agenti esterni che chiameremo attori. Quindi un attore ($ac$) rappresenta l'esecutore dell'operazione. Gli attori possono essere agenti umani, ad esempio uno scriba che corregge una parola di un manoscritto, ma anche strumenti informatici come ad esempio un convertitore di formato per un documento.

In linea con l'approccio evoluzionistico, possiamo visualizzare l'evoluzione di un documento come un grafo i cui nodi sono appunto i documenti. In accordo con la (9), le operazioni sono responsabili di trasformare un documento $D_0$ in $D_1$. E per quanto sopra, le operazioni sono fisicamente realizzate dagli agenti o attori. Nell'esempio di $op$ che aggiunge una frase a $D_0$, $op$ può essere fatta sia manualmente che automaticamente, in entrambi i casi c'è un agente che effettua l'operazione. Un modo, quindi più corretto di scrivere la (9) sarebbe $ac_{op} : D_0 \rightarrow D_1$, specificando che è la combinazione agente più operazione a far evolvere il documento.

Quindi, per evoluzione si intende una struttura a grafo diretto i cui nodi sono i documenti e i cui archi le operazioni (mediate da agenti) vedi Figura 1.

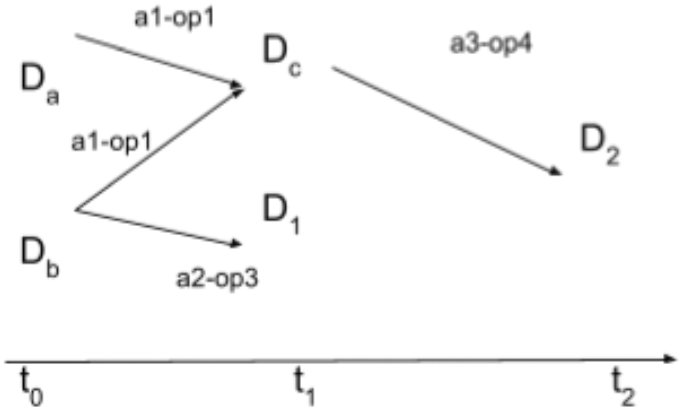


Figura 1 Evoluzione di differenti documenti a diversi istanti temporali

In Figura 1 le etichette degli archi indicano l'azione congiunta di un dato agente con una certa operazione. A titolo esemplificativo: *a1-op1* significa che l'agente ($a1$) ha effettuato $op1\left(D_a, D_b\right)$ per ottenere $D_c$.

Un grafo come quello in Figura 1 può essere applicato ad un processo di Linguistica Computazionale, dove da un insieme iniziale di documenti si passa a un output finale attraverso stati intermedi. In questo caso gli attori sono agenti automatici che effettuano operazioni di elaborazione automatica del linguaggio.

Nella stessa figura abbiamo messo una freccia del tempo per enfatizzare come da un gruppo di documenti iniziale il sistema evolva, attraverso documenti intermedi, verso un documento finale. Questa evoluzione ha una certa durata:

$$\Delta t = t_2 - t_0 \quad (12)$$

In ottica di ricostruzione dei rapporti tra le fonti, in Filologia, è molto utile permettere dei salti all'indietro, ovvero poter sapere, ad esempio all'istante t_1, cosa abbia reso possibile ottenere $D_c$ (in Figura 1).

Questo si può affrontare se ipotizziamo di poter effettuare delle "fotografie" del sistema in certi istanti. Sempre dalla Figura 1, abbiamo 3 istanti $t_0, t_1$ e $t_2$.

Consideriamo i documenti che abbiamo a disposizione a questi istanti:

$$B\left(t_0\right) = \backslash\backslash\{ D_a, D_b \backslash\backslash\}, B\left(t_1\right) = \backslash\backslash\{ D_C, D_1 \backslash\backslash\}, B\left(t_2\right) = \backslash\backslash\{ D_2 \backslash\backslash\} \quad (13)$$

Più genericamente, la (13) si può scrivere come:

$$B\left(\tau\right) = \backslash\backslash\{ D_1, D_2, \ldots, D_n \backslash\backslash\} \quad (14)$$

La (14) si legge "all'istante $t = \tau$ ci sono *n* documenti $D_1, D_2, \ldots, D_n$ disponibili.

Chiamiamo $B\left(\tau\right)$ spazio di lavoro, o ***base-space***, a $t = \tau$.

Si noti che gli spazi di lavoro nella (13) sono collegati tra loro. Infatti, $D_c$ in $B\left(t_1\right)$ dipende, attraverso $a1 - op1$, da entrambi i documenti in $B\left(t_0\right)$ mentre $D_1$ solo da $D_b$. Similmente per gli altri spazi di lavoro $B\left(t_2\right)$ e $B\left(t_1\right)$.

Definiamo lo spazio di evoluzione $H$ come una collezione di evoluzioni che collegano documenti appartenenti a diversi spazi di lavoro.[9]

9 Ovvero a diversi istanti temporali.

Sempre dalla Figura 1, uno spazio di evoluzione tra $t_0$ e $t_1$ contiene i seguenti elementi:

$$H=\left\{ev_1=\backslash\backslash\{ D_a \rightarrow D_c \backslash\backslash\}_{a1-op1}, ev_2=\backslash\backslash\{ D_b \rightarrow D_c \backslash\backslash\}_{a1-op1}, ev_3=\backslash\backslash\{ D_b \rightarrow D_1 \backslash\backslash\}_{a2-op3}\right\} \quad (15)$$

Gli elementi $ev$ di $H$ sono mappe vincolate nel senso che di tali oggetti sono noti gli stati iniziali e finali. Il valore iniziale è un sottoinsieme (anche proprio) dello spazio di lavoro a un dato istante (per esempio $t_0$), il valore finale è un sottoinsieme dello spazio di lavoro a $t=t_1, D_x \subset B(t_1)$:

$$ev(t_0)=\backslash\backslash\{ D_0,\ldots D_k \backslash\backslash\} \subset B(t_0); ev(t_1)=D_x \subset B(t_1) \quad (16)$$

Il significato della (15) è il seguente: se un ipotetico studioso, a $t=t_1$, sapesse con certezza come si è arrivati a $D_c$ partendo da $D_a$ e $D_b$, ovvero fosse nota l'operazione $a1-op1$, sarebbe in grado di replicare il processo che a partire dai documenti nello spazio di lavoro a $t=t_0, B(t_0)$, ha portato a $D_c \subset B(t_1)$.

Sfortunatamente, questo è un caso molto raro e spesso si possono solo supporre sia le fonti che le operazioni che da esse hanno portato al documento in esame. Nel caso di $D_1$, per esempio, la Figura 1 riporta un solo arco (da $D_b$ a $D_1$ attraverso $a2-op3$) ma non è noto, a priori, se tale evoluzione è unica oppure no.

Un lavoro di ricostruzione potrebbe supporre l'esistenza di $B(t_0)$ e di $D_b$ come documento di $B(t_0)$. Ma, in mancanza di ulteriori informazioni, non è possibile affermare con certezza che $D_b$ sia l'unico documento appartenente a $B(t_0)$ da cui si ottiene $D_1$. Ovvero che $a2-op3$ sia l'unica operazione responsabile dell'evoluzione di $D_1$. In questo caso, a differenza dell'esempio descritto nel paragrafo precedente, il processo è irreplicabile e solo speculativo. Infatti, anche nel caso in cui si sapesse che $B(t_0)$ consiste in solo due documenti, $D_a$ e $D_b$, la mancanza di informazioni ulteriori sullo spazio di evoluzione può far supporre l'esistenza di altre operazioni responsabili dell'evoluzione di $D_1$ e diverse da $a2-op3$, per esempio $op_x(D_a, D_b) \rightarrow D_1$ e $op_i(Da) \rightarrow D_1$.

La Figura 1 si presta a un diversa lettura: non rappresentazione di evoluzioni, ma connessioni tra spazi di lavoro a diversi istanti temporali. La Figura 2, infatti, mostra le relazioni tra spazi di lavoro e spazi di evoluzione. I diversi colori identificano le differenti combinazioni.

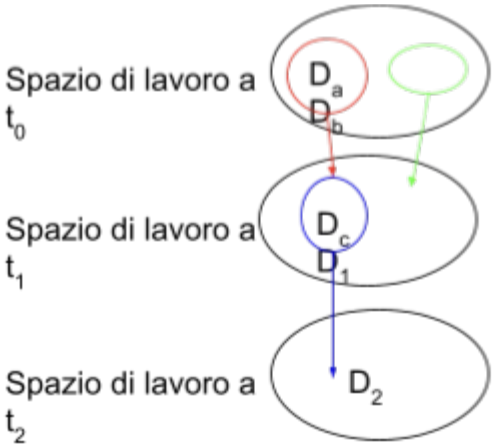


Figura 2 Rappresentazione grafica delle relazioni tra spazi di lavoro (in nero) e spazi di evoluzione (in diversi colori)

Definiamo lo spazio totale E come prodotto cartesiano tra uno spazio di lavoro e lo spazio delle storie evolutive:

$$E = B \times H \quad (17)$$

Lo spazio totale $E$ mette in relazione ogni documento in uno spazio di lavoro con le sue storie evolutive. Gli elementi di E sono coppie documento-evoluzioni: per esempio la coppia $\{D_1, ev_3\}$ mette in relazione il documento $D_1 \in B(t_1)$ con l'elemento $ev_3$ di $H$.[10] Come si evince dalla Figura 2, il processo di "raggruppamento"[11] è iterabile: da $D_2$ è possibile collegarsi a $D_c$ e da $D_c$ a $D_a$ e $D_b$.

## Evoluzione di documenti come sistema dinamico

Il modello, descritto nella sezione precedente, è stato pensato all'interno delle teorie di processi [19] e come tale ha insito sul concetto di cambiamento di stato, ovvero di dinamica.

10 Ricordiamo che $ev_3 = (D_b \rightarrow D_1)_{a2-op3}$

11 Raggruppamento è la traduzione di *bundle*. E bundle è la teoria matematica (topologica) da cui questa parte del modello deriva.

Proviamo ora a inquadrare il modello proposto come un sistema dinamico. Consideriamo un documento $D_0$ che evolve in un documento $D_1$.

In accordo all'equazione (9):

$$op : D_0 \rightarrow D_1 \text{ (18)}$$

Senza perdita di generalità possiamo supporre che solo il contenuto subisca variazioni sotto l'operazione $op$ nella (18):

$$op : c_0 \rightarrow c_1 \text{ (19)}$$

$$op : f_0 \rightarrow f_0 \text{ (20)}$$

$$op : \{ p_0 \} \rightarrow \{ p_0 \} \text{ (21)}$$

Un parametro essenziale per un sistema dinamico è il tempo t.

Il secondo principio della Dinamica afferma che la forza (o la risultante delle forze) che agisce su un corpo è direttamente proporzionale all'accelerazione del corpo stesso:

$$f = m \cdot a \text{ (22)}$$

Nella (22) l'accelerazione è la variazione della velocità nel tempo $a(t) = \Delta(v) / \Delta(t)$, per cui, introducendo la dipendenza temporale anche nella forza f, la (18) diventa:

$$f(t) = m \cdot \Delta(v) / \Delta(t) = m \cdot \frac{dv(t)}{dt} \text{ (23)}$$

dove $\frac{dv(t)}{dt}$ è la derivata (variazione infinitesimale) della velocità rispetto al tempo. Dato che la velocità v è la variazione infinitesimale dello spazio rispetto al tempo, la (23) si scrive:

$$f(t) = m \cdot \frac{d^2 s}{d t^2} \text{ (24)}$$

Le (23) e (24) prendono il nome di equazioni differenziali, la cui soluzione dipende da $f(t)$. In prima approssimazione possiamo scrivere:

$$s(t_1) \sim s(t_0) + A(f) \Delta(t) \text{ (25)}$$

Dove $A(f)$ è un "qualcosa che dipende da f".[12]

12 In questo momento non ci interessa come A dipenda da f (e da t).

Nel nostro modello, il tempo (*t*) interviene esplicitamente solo per definire gli spazi di lavoro. Comunque, implicitamente, il tempo (*t*) parametrizza le evoluzioni dei documenti. E le parametrizza attraverso la durata delle operazioni. Le nostre *unità temporali* corrispondono al numero di volte che un'operazione viene applicata.

Consideriamo ora nelle (19-21) *op* come l'analogo delle forze in un sistema dinamico. Come le forze sono responsabili della successione di stati del sistema, così le operazioni cambiano i documenti. In questo caso è lecito domandarsi se il sistema formale che abbiamo definito permetta di modellare l'evoluzione del contenuto del documento iniziale così come solitamente avviene in fisica attraverso l'utilizzo di equazioni differenziali. Chiaramente la risposta alla domanda non è univoca perché dipende da *op*. Prendiamo in considerazione due esempi di operazione per verificarlo. Se l'operazione fosse **rimuovi K parole da c** e c contenesse 100 parole, sappiamo esattamente calcolare c dopo l'applicazione *op*. Infatti possiamo scrivere un'equazione simile alla (25):

$$c(t_1) \sim c(t_0) + A(op)\Delta(t) \quad (26)$$

Dove abbiamo identificato con $A(op)$ l'applicazione di *op* al contesto c.

Quindi, nel caso di op=**rimuovi K parole da c**:

$$c_1 = c_0 - K, c_2 = c_1 - K = c_0 - 2K, \ldots c_n = c_0 - nK \quad (27)$$

Estrarre gli eventi, per esempio date, persone, da un dato testo è un ulteriore esempio, che può essere modellato come:

$$op : \{D_0, D_{eventi}\} \rightarrow D_1 \quad (28)$$

Dove $D_0, D_{eventi}$ sono rispettivamente il documento da cui si devono estrarre gli eventi e un documento che contiene la lista degli eventi. $D_1$ conterrà la lista di eventi estratti.

Se l'agente che esegue *op* è un agente automatico, è possibile sapere il numero di eventi in $D_1$ dati $D_0$ e $D_{eventi}$ e, al variare di $D_{eventi}$, si può sempre studiare come varia la lista in $D_1$. Ma se l'agente è umano questo non è sempre vero.

Un agente umano, ad esempio per distrazione, può saltare uno o più eventi producendo documenti finali potenzialmente diversi tra loro. Questo semplice esempio mostra che quando l'agente è umano il modello deterministico dinamico non è applicabile.

In effetti, la modellazione del processo, come descritta dalla (18), quando l'agente esecutore di *op* è umano, necessita di considerazioni aggiuntive sia in termini di riflessioni (di ricerca) sia di tecnicismi matematici da applicare al modello.

Da un punto di vista matematico la (18) può essere riformulata come segue:

$$op : D_0 \rightarrow D_1(\lambda) \quad (29)$$

dove $\lambda$ è un parametro che descrive (qualitativamente, in questo stadio) l'agente, ovvero la sua conoscenza di un dato processo (*op*).

Proviamo a capire meglio con un esempio e torniamo alla (18) vedendola alla luce della (29).

Supponiamo che *op* sia una interpretazione di un testo. Possono verificarsi due situazioni notevoli. La prima è che se due agenti diversi interpretano lo stesso testo, il risultato sarà (molto probabilmente) diverso. La seconda è che se lo stesso agente interpreta lo stesso testo a distanza di tempo, anche in questo caso, il risultato potrebbe essere diverso.

In questo esempio, $\lambda$ "modella", nel primo caso, la diversa conoscenza che i due agenti hanno del testo che devono interpretare, nel secondo il fatto che la conoscenza del testo da interpretare cambia nel tempo.

In entrambi i casi abbiamo che i documenti finali prodotti sono diversi:

$$op : D_0 \rightarrow D_1(\lambda) \quad (30)$$

$$op : D_0 \rightarrow D_1^{'}(\lambda^{'}) \quad (31)$$

$$D_1(\lambda) \neq D_1^{'}(\lambda^{'}) \quad (32)$$

Il parametro $\lambda$ nella (29) misura la conoscenza di un processo da parte di un agente umano.

Due questioni ci sembrano quindi interessanti: (i) come dipende $\lambda$ dal contesto in cui un'operazione viene effettuata? (ii) come dipende $\lambda$ dalle altre conoscenze intrinseche che ha l'agente umano?

Nell'esempio dell'interpretazione di un testo fatta da due agenti diversi *a* e *b*, postuliamo che la differenza (meccanicamente misurabile) del risultato sia una misura dell'impatto del contesto in cui avviene il processo interpretativo e della conoscenza specifica dei due agenti nel processo stesso. Anche nel caso di interpretazioni fatte dallo stesso agente, la differenza delle interpretazioni misura l'impatto del contesto e della conoscenza nel processo interpretativo, ma aggiunge anche una co-evoluzione del processo interpretativo e del rapporto agente-contesto/conoscenza. Una interpretazione cambia perché è cambiato il rapporto dell'agente con il contesto esterno ed anche la sua conoscenza del fenomeno.

## Conclusioni e lavori futuri

In questo articolo abbiamo proposto un modello di gestione dell'evoluzione dei documenti e abbiamo provato a proiettarlo su un sistema dinamico classico, evidenziandone i limiti di applicazione, soprattutto quando l'agente è umano.

Abbiamo definito un documento come la tripla contenuto, formato, informazioni paratestuali, fornendo esempi di documenti separabili e non separabili. Abbiamo descritto le operazioni come azioni (processi) che partono da un insieme di documenti iniziali per crearne uno finale. Infine, per inquadrare l'aspetto dinamico abbiamo introdotto il concetto degli spazi di lavoro definendoli come collezioni di documenti disponibili a un certo momento temporale. L'evoluzione dei documenti è vista come un percorso tra spazi di lavoro. L'uso dei bundles (cf. nota 10) permette di collegare un singolo documento in uno spazio di lavoro a documenti e spazi di lavoro precedenti.

L'idea di base è stata quella di modellare le parti meccaniche di un processo (variazioni di formato, aggiunta di livelli paratestuali, variazioni del contenuto informativo etc.), ma, allo stesso tempo, di predisporre un apparato matematico complesso[13] (e i bundles lo sono) che funga da base e cornice a processi più complicati come quelli che richiedono ricostruzioni, replicabilità etc.

L'agente umano aggiunge un aspetto non deterministico al modello che necessita di ulteriori riflessioni di ricerca. In questa prima fase, vogliamo gettare le fondamenta matematiche su cui sviluppare le implicazioni che un agente umano introduce e quindi limitare questo aspetto ad una "modellazione qualitativa".

Includere il non determinismo al modello consente ulteriori indagini ad esempio per quanto riguarda le metriche necessarie a misurare le differenze tra $D_1$ e $D_1^{'}$ nella (32).

## References

[1] Liskov, Barbara, and John Guttag. 2001. *Program Development in JAVA: Abstraction, Specification, and Object-Oriented Design*. Computer programming. Pearson Education

13 Per i curiosi l'equazione differenziale è dc/dt=-K che permette di calcolare il numero di parole in entrambe le direzioni temporali.

[2] Carrano, Frank, and Walter J. Savitch. 2011. *Data Structures and Abstractions with Java*. 3rd ed. USA: Prentice Hall Press.

[3] Friesen, Jeff. 2019. *Java XML and JSON Document Processing for Java SE*. https://doi.org/10.1007/978-1-4842-4330-5

[4] Boschetti, Federico, and Angelo Mario Del Grosso. 2014. "TeiCoPhiLib: A Library of Components for the Domain of Collaborative Philology." Edited by Arianna Ciula and Fabio Ciotti. *Journal of the Text Encoding Initiative*, no. 8.

[5] Lambek, Joachim. 2008. *From Word to Sentence: A Computational Algebraic Approach to Grammar*. Polimetrica sas.

[6] Coecke, Bob, Edward Grefenstette, and Mehrnoosh Sadrzadeh. 2013. "Lambek vs. Lambek: Functorial Vector Space Semantics and String Diagrams for Lambek Calculus." *Annals of Pure and Applied Logic* 164 (11): 1079–1100.

[7] Weitzman, Michael. 1985. "The Analysis of Open Traditions." *Studies in Bibliography* 38: 82–120.

[8] Schloen, David, and Sandra Schloen. 2014. "Beyond Gutenberg: Transcending the Document Paradigm in Digital Humanities." *Digital Humanities Quarterly* 8 (4). http://www.digitalhumanities.org/dhq/vol/8/4/000196/000196.html.

[9] Buckland, Michael K. 1997. "What Is a "document"?" *Journal of the American Society for Information Science* 48 (9): 804–809.

[10] Buckland, Michael. 1998. "What Is a Digital Document." *Document Numérique* 2 (2): 221–230.

[11] Lund, Niels Windfeld, and Roswitha Skare. 2010. "Document Theory." In *Encyclopedia of Library and Information Sciences*, 1632–1639.

[12] Buzzetti, Dino. 2002. "Digital Representation and the Text Model." *New Literary History* 33 (1): 61–88.

[13] Ciula, Arianna, and Øyvind Eide. 2017. "Modelling in Digital Humanities: Signs in Context." *Digital Scholarship in the Humanities* 32 (suppl. 1): i33–i46.

[14] Franke, Helena. 2005. "What's in a Name? Contextualizing the Document Concept." *Literary and Linguistic Computing* 20 (1): 61–69.

[15] Schmidt, Desmond. 2010. "The Inadequacy of Embedded Markup for Cultural Heritage Texts." *Literary and Linguistic Computing* 25 (3): 337–356.

[16] Ferilli, Stefano. 2011. *Automatic Digital Document Processing and Management: Problems, Algorithms and Techniques*. Berlin-Heidelberg: Springer.

[17] Buzzoni, Marina, Eugenio Burgio, Martina Modena, and Samuela Simion. 2016. "Open versus Closed Recensions (Pasquali): Pros and Cons of Some Methods for

Computer-Assisted Stemmatology." *Digital Scholarship in the Humanities* 31: 652–669.

[18] Del Grosso, Angelo Mario, Emiliano Giovannetti, and Simone Marchi. 2017. "Il Modello a Microkernel Di Omega Nello Sviluppo Di Strumenti per Lo Studio Dei Testi: Dagli ADT Alle API." In *AIUCD 2017 Book of Abstracts*, 199–205.

[19] Paul, Ralph. 2015. "Developing and Evaluating Software Engineering Process Theories." In *Proceedings of the 37th International Conference on Software Engineering (ICSE '15)*, 1:20–31. IEEE Press.

[20] Bédier, Joseph. 1928. "La Tradition Manuscrite Du Lai de l'Ombre. Réflexions Sur l'art d'éditer Les Anciens Textes (Premier Article)." *Romania* 54 (214): 161–196.

[21] Camps, Jean-Baptiste, and Florian Cafiero. n.d. "Stemmatology: An R Package for the Computer-Assisted Analysis of Textual Traditions." In *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities CRH-2*, 65–74. Vienna. https://doi.org/10.5281/zenodo.1117389.

[22] Dearing, Vinton A. 1959. *A Manual of Textual Analysis*. Berkeley: University of California Press.

[23] Froger, Jacques. 1968. *La Critique Des Textes et Son Automatisation*. Paris: Dunod.

[24] Gurry, Peter J. 2017. *A Critical Examination of the Coherence-Based Genealogical Method in New Testament Textual Criticism*. Leiden: Brill. https://doi.org/10.1163/9789004354548.

[25] Kenney, Edward J. 1974. *The Classical Text – Aspects of Editing in the Age of the Printed Book*. Sather Classical Lectures 44. Berkeley-Los Angeles-London: University of California Press.

[26] Lee, Arthur R. 1989. "Numerical Taxonomy Revisited: John Griffith, Cladistic Analysis and St. Augustine's Quaestiones in Heptateuchum." *Studia Patristica* 20: 24–32.

[27] Maas, Paul. 1927. *Textkritik*. 1st ed. Leipzig: B.G. Teubner Verlagsgesellschaft.

[28] Quentin, Henri. 1926. *Essais de Critique Textuelle (Ecdotique)*. Paris: A. Picard.

[29] Robins, William. 2006. "Editing and Evolution." *Literature Compass* 3: 89–120.

[30] Robinson, Peter. 1997. "A Stemmatic Analysis of the Fifteenth-Century Witnesses to The Wife of Bath's Prologue." In *Canterbury Tales Project Occasional Papers*, by Peter Robinson and N.F. Blake, II:69–132. London.

[31] Roelli, Philipp, ed. 2020. *Handbook of Stemmatology*. Berlin-Boston: De Gruyter. https://doi.org/10.1515/9783110684384.

[32] Salemans, Ben J.P. 2000. *Building Stemmas with the Computer in a Cladistic, Neo-Lachmannian, Way: The Case of Fourteen Text Versions of Lanseloet Van Denemerken*. Nijmegen: Katholieke Universiteit Nijmegen.

[33] Salemans, Ben J.P. 1996. "Cladistics or the Resurrection of the Method of Lachmann." In *Studies in Stemmatology*, by Pieter Th. van Reenen, Margot van Mulken, and Janet Dyk, I:3–70. John Benjamins Publishing.

[34] Salemans, Ben J.P. 1987. "Van Lachmann Tot Hennig: Cladistische Tekstkritiek." *Gramma* 11: 191–224.

[35] Timpanaro, Sebastiano. 2004. *La Genesi Del Metodo Del Lachmann*. Torino: UTET Università.

[36] Wisse, Frederik. 1982. *The Profile Method for the Classification and Evaluation of Manuscript Evidence, as Applied to the Continuous Greek Text of the Gospel of Luke*. Eerdmans.

[37] Zarri, Gian Piero. 1969. "Il Metodo per La «Recensio» Di Dom Quentin Esaminato Criticamente Mediante La Sua Traduzione in Un Algoritmo." *Lingua e Stile* 4: 161–182.

[38] Poole, Eric. 1979. "L'analyse Stemmatique Des Textes Documentaires." In *La Pratique Des Ordinateurs Dans La Critique Des Textes*, by J. Glenisson, Jean Irigoin, and R. Marichal, 151–161. Paris: CNRS Éditions.

[39] Poole, Eric. 1974. "The Computer in Determining Stemmatic Relationships." *Computers and the Humanities* 8 (4): 207–216.

[40] Greg, Walter Wilson. 1927. *The Calculus of Variants – An Essay on Textual Criticism*. Oxford: Clarendon Press.