Versione PROVVISORIA del contributo presentato al Convegno Annuale

La Filologia come sistema dinamico: qualche considerazione preliminare

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

[1] ILC-CNR, Italy - riccardo.delgratta(«»)ilc.cnr.it

[2] ILC-CNR & VeDPh, Italy - federico.boschetti(«»)ilc.cnr.it

[3] ILC-CNR, Italy - angelo.delgrosso(«»)ilc.cnr.it

[4] Università di Bologna, Italy - luigi.bambaci2(«»)unibo.it

ABSTRACT

In questo articolo introduciamo un approccio formale all'evoluzione dei documenti con particolare attenzione alla prospettiva filologica e alle problematiche tipiche connesse. Proponiamo un modello/framework 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, in particolare quando questa viene applicata all'analisi di documenti e testi di interesse storico-letterario.

PAROLE CHIAVE

Approccio evoluzionistico, modello formale, analisi documentale e testuale, sistema dinamico, filologia computazionale

1. INTRODUZIONE

I diversi tentativi fatti di formalizzazione dei metodi della Filologia testuale (o critica testuale) sono condizionati dalle specificità dei diversi domini di applicazione (Filologia Classica, Biblica, Romanza etc.). Il metodo del Lachmann è un chiaro esempio di questo problema: essendo modellato principalmente sulla tradizione dei testi greci e latini, esso risulta difficilmente applicabile al di fuori del dominio della Filologia Classica [7].

Risulta importante, quindi, promuovere una visione unitaria della Filologia testuale che evidenzi, da una parte, i fenomeni comuni alle diverse tradizioni testuali (es. fenomeni di corruzione involontaria o meccanica) e tenga conto, dall'altra, di fenomeni che si presentano differentemente in ciascun dominio, ma che sorgono da abitudini scribali comuni (innovazioni volontarie quali rielaborazioni esegetiche e stilistiche, interpolazioni dettate da motivazioni morali, teologiche e così via).

Il processo di formalizzazione non deve essere limitato alle entità, alle proprietà e alle relazioni che caratterizzano il testo filologico, ma deve anche tenere conto degli attori responsabili della creazione e della trasmissione di tali testi, siano essi tool automatici oppure agenti umani come autori, editori, scribi, traduttori, etc.

In questo articolo introduciamo un approccio formale all'analisi del testo prendendo in considerazione alcuni dei problemi menzionati sopra. In particolare, tenteremo di formalizzare il fenomeno della variazione testuale nel tempo attraverso un approccio di tipo evoluzionistico, dove per tale approccio intendiamo esattamente il processo dinamico che lega un documento a un altro, o due differenti versioni dello stesso documento.

2. UN SISTEMA DINAMICO, IN BREVE

Il più semplice sistema dinamico è quello della dinamica classica, formalizzato da Newton nei suoi *Principia Matematica*. Secondo tale sistema, in breve, se si conoscessero tutte le forze e tutti i dettagli di un sistema ad un dato istante t_0, allora si conoscerebbe il sistema nel suo complesso a qualsiasi istante t. Occorre notare che il presupposto è che, in ambito classico, descrivere un sistema significa descriverne ogni sua singola parte.

Nella sezione 3, definiamo alcune entità che ci aiutano a vedere alcuni aspetti della Filologia (ma anche della Linguistica Computazionale) come un sistema dinamico. Vedremo, nel corso dell'articolo, fino a che punto il parallelismo con un sistema dinamico classico sia pertinente.

3. IL MODELLO PROPOSTO, ALCUNE DEFINIZIONI UTILI

La prima definizione che proponiamo è quella di documento. La definizione proposta è coerente con quelle usate dai principali linguaggi di programmazione [4] a vari livelli di astrazione [1],[2],[3].

Per agevolare la comprensione da parte del lettore, parleremo di documenti testuali *elettronici.* Tale definizione di documenti e testi proposta è utile per la Filologia Digitale e Computazionale poiché modella aspetti noti quali le varianti, le congetture, la ricostruzione delle relazioni tra le fonti, ecc.

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

$$D = D(c,f,\{p\}) \ (1)$$

dove *c* è il contenuto informativo trasportato dal documento, *f* è il formato e *{p}* è un insieme di informazioni paratestuali aggiuntive.[1] Sia *c* che *{p}* possono essere vuoti, questo per consentire la possibilità del documento vuoto.[2] Per esempio, se D è un documento di testo piano[3] contenente una sequenza di caratteri a formare una frase, *c* è la frase, *f* è il formato plain text e *{p}* è vuoto. In documenti più complessi (file XML-TEI, per esempio) la separazione tra *c, f* e *{p}* non è così immediata. Anche se non saranno usati in questo articolo, è utile introdurre la definizione di documenti cartesiani. Un documento si dice cartesiano se è separabile nelle sue parti costituenti:

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

intendendo con *X* il prodotto cartesiano tra le varie parti. In un certo senso, nei documenti cartesiani o separabili, *c, f* e *{p}* sono tre entità indipendenti. Il senso della (2) è che D è identificato dalle proprie parti, esattamente come un punto in un piano cartesiano lo è dalle proprie coordinate.

Molte delle cose che diremo nel presente contributo sono indipendenti dal fatto che D sia cartesiano o meno, dato che lavoriamo ad un livello di astrazione più alto.

Una ulteriore sotto-definizione importante è quella di documento esteso. Un documento esteso è un documento "accompagnato" da altre informazioni:

$$D_{ex} = D(c,f,\{p\}) \ X \ \{m\} \ (3)$$

{m} può essere il periodo storico, l'autore, un riferimento a una versione precedente ecc. Nel caso di documenti digitali {m} può contenere, ad esempio, il nome del file e dove risiede fisicamente.

La seconda definizione è quella di azione o operazione. Una operazione *op* è "qualcosa che elabora un insieme di documenti e restituisce un singolo documento".[4] Ovvero:

$$op(\{D_1, \ D_2, .., D_n\}) = \{D_a\} \ (4)$$

Il senso della (4) è il seguente: *op* lavora su n documenti in input e crea un nuovo documento D_a in output. In caso di documenti cartesiani, anche le operazioni che agiscono su questi si considerano cartesiane:

$$op = \ op_c \ X \ op_f \ X \ op_{\{p\}} \ (5)$$

con l'ovvio significato di operazioni specifiche per parti specifiche: ad esempio op_c agisce solo su *c,* lasciando le altre inalterate. È come se op_c, op_f e $op_{\{p\}}$ agissero parallelamente su un documento D, ma la loro azione fosse limitata alla parte di competenza.[5]

Occorre aggiungere che il nostro approccio sarà quello di considerare D_a sempre un documento nuovo.[6] Consideriamo come D_0 il solito documento in formato plain text di cui sopra; supponiamo poi che D_0 contenga un insieme di frasi. Se *op* è, per esempio, l'aggiunta di una frase, D_1 è il risultato di $op(D_0) = D_1$. Anche se fisicamente D_1 è lo stesso file,[7]

---

[1] Le parentesi {} indicano il fatto che il paratesto p sia un insieme di informazioni. Dove è scritto {p} è da intendersi $\{p_1, p_2..., p_n\}$.

[2] Il documento vuoto deve essere supposto per permettere la chiusura del sistema formale rispetto alle operazioni (definite dopo). Un'operazione, per esempio un tool di estrazione, può non estrarre niente. Il suo risultato è appunto un documento vuoto.

[3] Creato con editor testuali come notepad, ad esempio. Oppure con *vim* su sistemi Linux.

[4] Ovvero un insieme con un solo elemento.

[5] Questo aspetto è molto importante all'interno delle cosiddette "teorie dei processi"; ma lo è anche all'interno dell'ingegneria del software. Un software disegnato per gestire documenti cartesiani ha aspetti diversi da uno nato per gestire documenti non cartesiani. Anche se i due software hanno lo stesso scopo.

[6] Approccio funzionale.

[7] Ovvio in caso di file testuali presenti su pc.

per il modello proposto D_0 e D_1 sono logicamente due entità distinte.

Possiamo introdurre il concetto di "evoluzione" di un documento. Per evoluzione si intende una struttura a grafo i cui nodi sono i documenti e i cui archi le operazioni, vedi Figura 1.

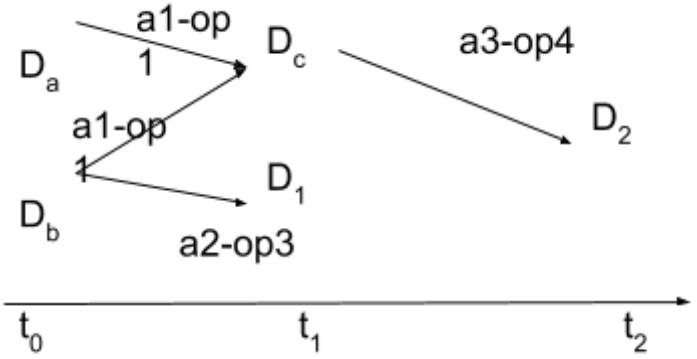

Figura 1. Evoluzione di differenti documenti a diversi istanti temporali

Le operazioni sono azioni astratte che vengono rese possibili attraverso l'intervento di agenti o attori (*ac*). Un attore (ac) è un tool automatico o un agente umano che esegue una o più operazioni. In Figura 1 le etichette degli archi stanno esattamente a indicare l'azione congiunta di un dato agente con una certa operazione. A titolo esemplificativo: a1-op1 significa che l'agente (a1) ha effettuato op1(D_a,D_b) 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.

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

$$\Delta t = t_2 - t_0 \ (6)$$

In ottica di ricostruzioni 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.
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:

$$B(t_o) = \{D_a,\ D_b\},\ B(t_1) = \{D_c,\ D_1\},\ B(t_2) = \{D_2\}\ (7)$$

Più genericamente, la (7) si può scrivere come:

$$B(\tau) = \{D_1,\ D_2, ...,\ D_n\}\ (8)$$

La (8) si legge "all'istante $t = \tau$ ci sono n documenti $D_1,\ D_2, ...,\ D_n$ disponibili.
Chiamiamo $B(\tau)$ spazio di lavoro o base-space a $t = \tau$.

Si noti che gli spazi di lavoro nella (7) sono in qualche modo collegati. Infatti, D_c in $B(t_1)$ dipende, attraverso a1-op1, da entrambi i documenti in $B(t_o)$ mentre D_1 solo da D_b. Similmente per $B(t_2)$ e $B(t_1)$.

Definiamo lo spazio di evoluzione H come una collezione di evoluzioni che collegano documenti appartenenti a diversi spazi di lavoro.[8] Sempre dalla solita Figura 1, uno spazio di evoluzione tra t_o e t_1 è il seguente:

$$ev_1 = \{(D_a -> D_c)_{(a1-op1)}\}, ev_2 = \{(D_b -> D_c)_{(a1-op1)}\}, ev_3 = \{(D_b -> D_1)_{(a2-op3)}\}$$

Gli oggetti *ev* sono "mappe vincolate" nel senso che di *ev* sono noti i valori iniziali e finali, il valore iniziale è un sottoinsieme anche proprio dello spazio di lavoro a un istante $t = t_0$, il valore finale è un documento sottoinsieme dello spazio di lavoro a $t = t_1$, $D_x \subset B(t_1)$.

[8] Ovvero a diversi istanti temporali.

$$ev(t_0) = \{D_0, ..., D_k\} \subseteq B(t_0);\ ev(t_1) = D_x \subset B(t_1)\ (9)$$

La (9) ci dice che se un ipotetico studioso, a $t = t_1$, sapesse con certezza come si è arrivati a D_c partendo da D_a, D_b, ovvero conoscendo l'operazione a1-op1, sarebbe in grado di replicare il processo che da $B(t_o)$ 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, la Figura riporta un solo arco da D_b a D_1 attraverso a2-op3. Un lavoro di ricostruzione potrebbe supporre la conoscenza di $B(t_o)$ ma non di ogni singola evoluzione *ev* che da $B(t_o)$ ha portato a D_1, in questo caso il processo è irreplicabile e solo speculativo.[9]

La Figura 2 mostra le relazioni tra spazi di lavori e spazi di evoluzione. I diversi colori identificano le differenti combinazioni.

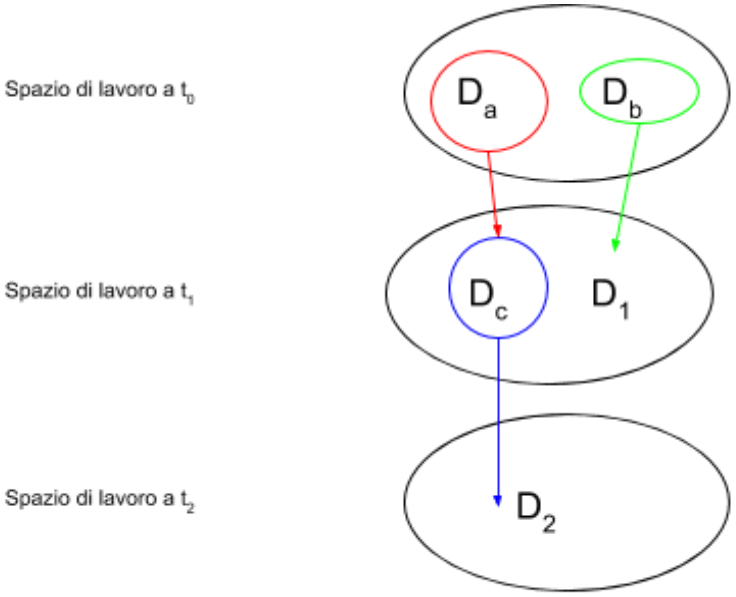

Figura 2: Rappresentazione grafica delle relazioni tra spazi di lavoro (in nero) e spazi di evoluzione (in diversi colori)

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

$$E = B\ X\ H\ (10)$$

Lo spazio totale 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 $\{D_1, ev_3\}$. Come si evince dalla Figura 2, il processo di "raggruppamento"[10] è iterabile: da D_2 mi collego a D_c e da D_c a D_a e D_b.

4. EVOLUZIONE COME DINAMICA

Proviamo a proiettare il modello proposto all'interno di un sistema dinamico. Focalizziamoci su un caso semplice:

$$D_0 \rightarrow D_1\ (11)$$

Come variabile dinamica prendiamo il solo contenuto *c*, ovvero si suppone che né *f* né *{p}* cambino. Un'operazione $op(D_0) \rightarrow D_1$ manda c_0 in c_1:

$$op(c_0) \rightarrow c_1\ (12)$$

Dove *op* è l'analogo delle forze in un sistema dinamico. Come le forze sono responsabili di "far muovere" un sistema da uno stato ad un altro, così le operazioni "fanno muovere" i documenti. Ci dobbiamo chiedere se sia possibile o meno

[9] Ovvero, se a t=t1, fosse noto solo lo spazio di lavoro B(t0)) ma non lo spazio di evoluzione, si potrebbe ipotizzare l'esistenza di una operazione $op_1(D_a, D_b)$ -> D_1, che è diversa da a2-op3(D_b)->D_1. In questo senso, il processo è speculativo.

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

scrivere delle equazioni (differenziali) che modellino esattamente l'evoluzione di c. Chiaramente la risposta alla domanda non è unica, dato che dipende da *op*. Un esempio semplice: se *op* è **togli K parole da c**, e c contiene 100 parole, sappiamo esattamente calcolare c dopo l'applicazione di *op.*[11]
Estrarre gli eventi da un dato testo è un ulteriore esempio, che può essere modellato come:

$$\{D_0, D_e\} \rightarrow D_1 \quad (13)$$

Dove D_0, D_e 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_e e D_0, e, al variare di D_e e D_0, 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, per distrazione, può saltare uno o più eventi. Questo semplice esempio mostra che quando l'agente è umano il modello deterministico dinamico non è applicabile. Torniamo alla (11) e 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. Noi riteniamo che un aspetto interessante di questo modello risieda proprio nel descrivere il comportamento di un agente umano, come spieghiamo nelle conclusioni.

5. 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. Questo, crediamo, non è un punto debole, ma piuttosto un di forza del modello proposto. L'idea di base è quella di modellare il più possibile 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[12] (e i bundles -cf. nota 10- lo sono) che funga da base e cornice a processi più complicati come quelli che richiedono ricostruzioni, replicabilità etc. Gli spazi di lavoro così come quelli di evoluzione e il loro raggruppamento (bundle) servono a questo scopo. Nell'ultima parte della Sezione 4 abbiamo accennato a due situazioni interessanti, entrambe collegate all'agente umano.

Un'ulteriore linea di ricerca riguarda lo studio del contesto esterno e della conoscenza di un agente umano sul "prodotto" di un processo. Nell'esempio dell'interpretazione di un testo fatta da due agenti diversi *a* e *b*, noi postuliamo che la differenza del risultato (la differenza è meccanicamente misurabile) sia una misura dell'impatto del contesto e della conoscenza specifica dei due agenti nel processo interpretativo. 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 è variata la sua conoscenza del fenomeno.
Tutto questo è, a nostro avviso, affascinante e merita ulteriori approfondimenti e sperimentazioni.

BIBLIOGRAFIA

[1] Liskov, B., e J. Guttag. 2001. *Program Development in Java: Abstraction, Specification, and Object-oriented Design*. Computer programming. Addison-Wesley.

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

[3] Friesen, J. 2019. *Java XML and JSON: Document Processing for Java SE*. Apress.

[4] Boschetti, Federico, e Angelo Mario Del Grosso. 2015. «TeiCoPhiLib: A Library of Components for the Domain of Collaborative Philology». A cura di Arianna Ciula e Fabio Ciotti. *Journal of the Text Encoding Initiative*, n. 8 (gennaio).

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

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

[5] Lambek, J. 2008. *From Word to Sentence: A Computational Algebraic Approach to Grammar*. Open access publications. Polimetrica

[6] Coecke, Bob, Edward Grefenstette, e Mehrnoosh Sadrzadeh. 2013. «Lambek vs. Lambek: Functorial vector space semantics and string diagrams for Lambek calculus». *Ann. Pure Appl. Log.* 164 (11): 1079–1100.

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