Risorse e sfide per la collazione automatica di testi medievali

Elena Spadini

L'utilizzo dell'informatica nelle pratiche ecdotiche ha una lunga storia, che ha inizio con il tentativo di automatizzare l'attività di collazione. A partire dagli anni '60, vengono identificate le operazioni di base di un pro- gramma di allineamento e riconosciuti gli aspetti più problematici della sua applicazione.[1] Questo contributo vuole offrire una panoramica dello stato dell'arte, con particolare attenzione alla collazione automatica di testi romanzi medievali.[2] In secondo luogo e brevemente, per completare l'analisi della situazione attuale, si propongono nuovi approcci ad alcuni degli annosi problemi già noti ai pionieri della filologia computazionale.

Collazione automatica e Gothenburg model

La collazione consiste nel comparare due o più testi al fine di individuare le loro differenze. In termini filologici, ciò che si compara sono le lezioni dei testimoni e tramite la collazione si ottengono le varianti tra di essi.

Il procedimento definito 'collazione automatica' prevede l'utilizzo di un computer in una o più delle sue fasi: non si tratta quindi di delegare alla macchina l'intero procedimento, come vedremo, ma di avere la sua assistenza in vari momenti. In questo articolo useremo 'collazione auto- matica' nell'accezione qui esplicitata e per brevità, rispetto a formule più trasparenti ma che appesantirebbero il dettato, come 'collazione semi- automatica' o 'collazione assistita'.

[1] E. Nury, E. Spadini, «From giant despair to a new heaven: The early years of automatic collation», *it - Information Technology*, vol. 62, 2 (2020), DOI:10.1515/itit-2019-0047. [2] Alcune delle considerazioni presentate potranno essere valide anche per testi di
tradizioni e epoche differenti.

Le operazioni costitutive di un programma di collazione automatica sono oggi ben definite. Chi voglia sviluppare un nuovo software può avvalersi di un modello e chi si accinga a studiare tali software può uti- lizzare lo stesso come griglia di analisi. Il modello in questione è quello definito a Göteborg,[3] nel 2009, dagli sviluppatori di Juxta e di CollateX e dai filologi del COST Action *Open Scholarly Communities on the Web* e del programma *Interedition*: esso è conosciuto come 'Gothenburg Model'.[4] Il modello stabilisce le cinque operazioni fondamentali che un programma di collazione automatica compie, seguendo il principio di progettazione della 'separation of concerns'. Ciò implica che le operazioni siano ripartite in sezioni separate dell'architettura software: ognuna di esse è addetta a un compito, non c'è sovrapposizione di inca- richi e, nel caso si voglia cambiare una parte, uno o più moduli possono essere sostituiti senza dover modificare l'intero ingegno.

Le operazioni fondamentali identificate dal modello di Göteborg sono la tokenizzazione, la normalizzazione, l'allineamento, l'analisi (o *feedback*) e l'*output*. Si passino brevemente in rassegna.

La tokenizzazione è il primo passo da compiere nella maggior parte dei trattamenti computazionali del testo e consiste nella divisione della sequenza di caratteri (lettere, punteggiatura, segni diacritici e spazi bian- chi) in unità minime. La dimensione e la natura dell'unità dipendono dal trattamento da eseguire: per la collazione, così come per molte altre attività, l'unità minima o *token* sarà la parola.[5]

Sulla seconda operazione, la normalizzazione, torneremo in seguito. Basti dire che si tratta di quella fase in cui il testo viene preparato per l'allineamento: il grado di normalizzazione può variare, dalla neutra- lizzazione delle differenze tra maiuscole e minuscole o delle varianti

[3] Utilizziamo la forma svedese del nome, che non ha una traduzione italiana, tranne quando occorre nell'espressione inglese *Gothenburg model*.

[4] Cf. R. Haentjens Dekker, G. Middell, «Computer-Supported Collation with CollateX: Managing Textual Variance in an Environment with Varying Requirements», in *Sup- porting Digital Humanities*, Copenhagen, University of Copenhagen, 2011; R. Haentjens Dekker et al., «Computer-Supported Collation of Modern Manuscripts: CollateX and the Beckett Digital Manuscript Project», *Digital Scholarship in the Humanities*, 3o/3 (2015), pp. 452-470; http://www.interedition.eu/wiki/index.php/About_microservices e https://wiki.tei-c.org/index.php/Textual_Variance (ultimo accesso 30 gennaio 2019).

[5] La complessità della tokenizzazione varia di lingua in lingua, ma generalmente biso- gna considerare che gli spazi bianchi e la punteggiatura non sono sufficienti per dividere le parole (si pensi ad esempio alle abbreviazioni). Inoltre, diverse tokenizzazioni sono possibili sulla stessa sequenza: la punteggiatura, ad esempio, può essere eliminata, costi- tuire un *token* a sé o un *token* insieme alla parola che precede.

ortografiche ad un livellamento più profondo, come ad esempio quello morfologico.

La terza operazione, l'allineamento, è il cuore del programma di col- lazione, la fase in cui il programma stabilisce quale *token* nel testo A corrisponde a quale *token* nel testo B.[6]

Il *feedback*, opzionale, permette al programma di correggere l'allinea- mento a partire dai risultati di un'analisi computazionale: ad esempio, il programma valuta la similarità tra i *token* e perfeziona l'allineamento in modo che *token* simili, seppure non uguali, corrispondano.[7]

L'ultima fase è quella *dell'output*, che consiste nella resa dei risultati in visualizzazioni quali la tabella e il grafo o in formati di codifica dei dati come XML o JSON.

L'architettura di CollateX e JuxtaCommons, i due programmi di colla- zione più adoperati negli ultimi anni, è basata sul modello di Göteborg.[8] Nello studio presentato in quest'articolo, è il primo ad essere usato: sia CollateX che Juxta sono software scritti nel linguaggio di programma- zione Java, ma solo per CollateX è stata sviluppata parallelamente una versione in Python,[9] un linguaggio più diffuso nella ricerca in filologia computazionale; inoltre, CollateX gestisce per impostazione predefi- nita le forme normalizzate dei *token*, caratteristica che, come vedremo, risulta fondamentale per il trattamento dei testi medievali.[10]

La collazione automatica di testi medievali

L'adozione della collazione automatica in filologia medievale è stata fre- nata da problemi concreti. Il primo di questi, forse il più importante, è la necessità di acquisire i testi, per poterli poi trattare e collazionare.[11]

[6] Faremo di seguito riferimento ai *token* come unità minime definite nella prima fase del programma di collazione; come abbiamo detto, essi corrispondono alle parole.

[7] È quello che succede attivando il parametro 'near match' della funzione 'collate' in CollateX, sul quale si tornerà di seguito.

[8] https://collatex.net/doc/ e http://juxtacommons.org/tech_info (ultimo accesso, 30 gennaio 2019).

[9] https://pypi.python.org/pypi/collatex (ultimo accesso, 30 gennaio 2019).

[10] La scelta di CollateX è dovuta anche alla possibilità di avere un supporto imme- diato da parte del suo principale sviluppatore, Ronald Haentjens Dekker.

[11] La questione non è nuova: cf. ad esempio S. Waite, «Two programs for comparing texts», in *La pratique des ordinateurs dans la critique des textes*, Paris, Ed. du CNRS, 1979, pp. 241-244:
244. Il metodo del '*double keying*' discusso da Waite è quello ritenuto anche in M. Piotrowski,
Natural language processing for historical texts, San Rafael, Morgan and Claypool, 2012, p. 48.

A seconda dell'opera, della sua lunghezza, della quantità di testimoni e del tipo di tradizione, l'acquisizione dei testi può richiedere un impegno diverso. Nonostante la ricerca sul riconoscimento delle scritture mano- scritte avanzi rapidamente, chi si accinga oggi a lavorare su un'opera tràdita da manoscritti medievali deve trascrivere, almeno in parte, ma- nualmente.[12] Le letterature romanze del medioevo – proponendo una larga generalizzazione –, sono ricche in opere, largamente tràdite e a tra- dizione attiva.[13] Trascrivere integralmente dieci o più manoscritti di un testo, magari lungo, a mano è un'operazione non solo onerosa, ma anche comunemente reputata inutile; mentre è in genere accettata l'idea di dedi- care una fase del lavoro, anche non breve, alla collazione. Ci si potrebbe allora chiedere quale delle due attività sia impegnativa in termini di tempo. La risposta, anche in questo caso, dipende dalle variabili elencate prima, ovvero dalla quantità di testimoni e dalla natura della tradizione; la riflessione di Tara Andrews, qui riportata, può ad esempio non esse- re valida per un romanzo arturiano in prosa del Duecento: a causa della varianza diffusa e di diverso tipo, una nuova trascrizione non potrà essere approntata copiando e cambiando la trascrizione di un testimone simile.

[W]itness transcriptions need not take any more time than manual collation of texts, and can usually be made to take considerably less. A full text transcription of any witness may be made simply by copying and altering the transcription of a similar witness, a process that is akin to the creation of a spreadsheet of vari- ants but simpler and easier in execution. There is a small risk that the readings of the similar text might influence the readings of the manuscript being tran- scribed, but that is more than offset by the fact that a full transcription removes the scholar's temptation to exclude a peculiar reading because it seems not to be worth the effort to set up a new variant location in a spreadsheet.[14]

[12] Se non altro per creare un 'training corpus' a partire dal quale la macchina possa poi imparare. Per il programma Transkribus, ad esempio, sono necessarie un centinaio di pagine di trascrizione per poter allenare il software e creare un modello, che si possa poi utilizzare sulle restanti pagine, cf. https://transkribus.eu/Transkribus/ (ultimo accesso 30 gennaio 2019).

[13] La prima variabile non è rilevante per il lavoro sulla singola opera, ma assume importanza in una riflessione più ampia su un'agenda di ricerca: lo sforzo che si può dedicare ad una singola opera è inversamente proporzionale alla quantità di opere di una tradizione letteraria, facendo astrazione dell'importanza culturale relativa di ognuna di esse.

[14] T. Andrews, «The Third Way: Philology and Critical Edition in the Digital Age», *Variants*, 10 (2013), pp. 61-76: 67.

Il vantaggio, dunque, non è da cercare nel tempo risparmiato grazie all'utilizzo di metodi computazionali. Bisognerà piuttosto guardare nella direzione che Wilhelm Ott, autore del più longevo programma di ecdo- tica digitale, TuStep, proponeva nel 1973:

To sum up: by means of this new tools, which we have in electronic data pro- cessing, new and higher standards are imposed not only on the results of others sciences, but also on critical editions ... The question whether it is possible or not to save time and / or money by these methods is only of secondary importance. The expenses necessary for future critical editions may possibly be even higher than they have been in the past when these tools were not yet available.[15]

Parlare di più alti standard per l'ecdotica potrebbe equivalere a met- tere in discussione i metodi non computazionali, e questo sarebbe un errore. Piuttosto, si può ritenere del proposito di Ott l'idea di nuovi stan- dard che si inseriscono nel panorama scientifico. Per quanto riguarda la collazione, la novità e il vantaggio si troverebbero nel tipo di dati che vengono generati durante la *recensio*: non griglie di collazione ad uso privato, ma trascrizioni che permettono la riproducibilità del tratta- mento, divenuto esplicito, e che sono riutilizzabili. Le trascrizioni dei manoscritti possono ad esempio essere riprese per la creazione di un corpus di analisi linguistica; il programma di collazione può essere ese- guito su quelle stesse trascrizioni a più riprese, tanto da colui che edita il testo quanto dal lettore.[16] Ci torneremo nelle conclusioni.

L'altro limite all'adozione di strumenti computazionali in filologia medievale è imposto dal tipo di variazione: la quantità di varianti che interessano aspetti formali della lingua, grafici e fonetici, è generalmente maggiore del numero di varianti di sostanza, che implicano un cambia- mento di significato o di stile rilevante. Vero è che quest'ultimo criterio, quello della rilevanza, non si può in alcun modo generalizzare: nell'eser- cizio della filologia stemmatica, la variante rilevante è monogenetica, le altre poligenetiche; ma, come ricordano Leonardi e Morato:

[l]a distinction entre substance et forme et leur rapport avec la monogenèse ou la polygenèse ne peuvent ... être établis a priori, mais requièrent une évaluation

[15] W. Ott, «Computer Applications in Textual Criticism», in *The Computer and Literary Studies*, a cura di A.J. Aitken et al., Edinburgh, Edinburgh University Press, 1973, pp. 199-223: 222.

[16] Per far sì che la riproducibilità sia reale, bisognerà mettere a disposizione tutti i pre- e post-trattamenti applicati ai dati, insieme ai testi e ai programmi.

empirique des mécanismes de transmission actifs pour chaque genre littéraire et chaque domaine linguistique et chronologique, si ce n'est pour chaque texte et chaque manuscrit.[17]

In ogni caso, la collazione automatica, effettuata sulle trascrizioni dei testi, produrrà una selva di varianti minori, ovvero una massa di infor- mazioni di poco o scarso rilievo ai fini della *recensio*. Lo constatano i filologi che vi hanno fatto ricorso:

[t]he main problem in the automatic comparison of different manuscripts of medieval texts is not so much to identify textual variants, but to distinguish between important and unimportant variants.[18]

La questione non si pone solo per la collazione, ma per buona parte dei trattamenti computazionali applicati a testi medievali. In un articolo del 2012, Karina van Dalen-Oskam applica metodi stilometrici di attribu- zione autoriale ad un corpus di manoscritti della *Scolastica* di Jacob van Maerlant per studiare l'attitude dei diversi copisti verso il testo; il proce- dimento è però ostacolato da differenze insignificanti:

the measurements on the Middle Dutch text are indeed highly influenced by irrelevant spelling differences. If we want to abstract from spelling, we need to do the measurements on normalised text, or on lemmatised text.[19]

Le soluzioni proposte da van Dalen-Oskam, la normalizzazione o la lemmatizzazione, sono quelle più adottate per ridurre il disordine di lingue non standardizzate ad un ordine che ne permetta la gestione tramite strumenti computazionali, generalmente pensati per lingue moderne con

[17] L. Leonardi, N. Morato, «L'édition du cycle de Guiron le Courtois. Établissement du texte et surface linguistique», in *Le Cycle de Guiron le Courtois. Prolégomènes à l'édi- tion intégrale du corpus*, Paris, Garnier, 2018, pp. 453-509: 472. Il ricco articolo di Leonardi e Morato insiste sulla necessità di una linea di divisione mobile tra forma e sostanza, tra monogenesi e poligenesi, linea che va spostata a seconda del corpus in esame: non si può ridurre d'ufficio la poligenesi ai fenomeni grafico-fonetici, al con- trario «l'objectif est d'identifier les phénomènes d'ordre morpho-syntaxique, lexical, discursif, qui se comportent comme des phénomènes polygénétiques, ou qui sont à tout le moins tendanciellement ou potentiellement polygénétiques» ivi, p. 475.

[18] L. Zeevaert, «Easy Tools to Get to Grips with Linguistic Variation in the Manuscripts of *Njáls Saga*», *Digital Medievalist*, 10/1 (2015), p. 50.

[19] K. van Dalen-Oskam, «The Secret Life of Scribes. Exploring Fifteen Manuscripts of Jacob van Maerlant's *Scolastica* (1271)», *Literary and Linguistic Computing*, 27/4 (2012), pp. 355-72: p. 359.

forme canoniche. Michael Piotrowski identifica le seguenti caratteristiche delle lingue pre-moderne, che ne impediscono il trattamento automa- tico: variazione diacronica, variazione sincronica e incertezza della tra- scrizione, in quanto risultato di un'attività soggetta ad errori. Per ovviare almeno ai primi due problemi, in linguistica computazionale si tende a normalizzare la lingua:

> in NLP [Natural Language Processing][20] canonicalization and normalization usually imply a mapping to a modern form. Spelling modernization would possibly be a more precise term for this approach.[21]

Una volta stabilite le corrispondenze tra la lingua storica e quella moderna si possono usare metodi, strumenti e risorse disponibili per la seconda e applicarli alla prima. Per identificare le corrispondenze, Piotrowski menziona la canonizzazione assoluta, che prevede l'utilizzo di una lista di forme corrispondenti, rigorosa ma onerosa da implemen- tare. Altri approcci discussi, detti di canonizzazione relativa, mirano alla creazione di forme canoniche diverse da quelle della lingua moderna e includono metodi che tengono conto della distanza di *edit*[22] e della simi- larità fonetica tra le varianti.[23]

Nella storia della collazione automatica diversi modi di gestire la varianza di 'spelling' (inteso come separazione delle parole, grafia, fone- tica, abbreviazioni e punteggiatura) si sono avvicendati. La scelta non è stata generalmente quella della modernizzazione, ma di altri tipi di normalizzazione, equivalenti agli approcci di canonizzazione relativa discussi da Piotrowski. Tutti vengono effettuati prima dell'allineamento vero e proprio e corrispondono dunque alla seconda fase (in alcuni casi alla prima e alla seconda) del modello di Göteburg, quella della nor- malizzazione. Tutti sono procedimenti di preparazione del testo, ai fini di ottenere risultati ottimali nella fase successiva: non solo riducendo

[20] L'inglese si può rendere in italiano con 'Trattamento automatico del linguaggio' o 'Elaborazione del linguaggio naturale'.

[21] Piotrowski, *Natural language processing for historical texts*, p. 69 (corsivo nell'ori- ginale).

[22] La 'edit distance' tra due *token* indica la quantità di operazioni (aggiunte, cancella- zioni e sostituzioni) sulle lettere necessarie per passare da una parola all'altra; general- mente, una soglia di '*edit distance*' è definita, sotto alla quale i due *token* vengono allineati, dunque considerati corrispondenti. Ad esempio, la '*edit distance*' tra 'legemmo' e 'leg- gemmo' (aggiunta di una lettera) è 1, così come quella tra 'scrisse' e 'scripse' (sostituzione di una lettera).

[23] Piotrowski, *Natural language processing for historical texts*, Capitolo 6.3.2.

il numero di varianti poco significative, ma anche limitando gli errori di allineamento che le varianti formali possono indurre. È importante notare che la normalizzazione e la preparazione del testo per l'allineamen- to non hanno conseguenze sulla trascrizione, ovvero sui dati raccolti dal filologo: la trascrizione rimane intatta e se ne crea una copia di lavoro da modificare per l'allestimento della collazione. Tramite meccanismi di codifica, le due copie possono convivere nello stesso file, come due facce della stessa medaglia: per la lettura della trascrizione se ne utilizzerà una e per l'allineamento l'altra. Il programma di collazione dovrebbe dare la possibilità di passare dall'una all'altra ad ogni momento; ciò implica, ad esempio, che i risultati dell'allineamento eseguito sulla copia nor- malizzata possano essere visualizzati secondo la lettera della versione originale.

Prima di passare in rassegna meccanismi specifici di normalizzazione, si trattino brevemente quelli più comuni, che sono rimasti quasi intatti negli ultimi cinquant'anni, seppur per motivi diversi. Nel 1966, Dom Jacques Froger scriveva:

> Avec la petite machine dont nous nous sommes servis, les moyens d'exprimer un texte se réduisaient à ceci :
>
> — des lettres exclusivement capitales, et par conséquent aussi les chiffres romains, puisqu'ils sont formés de lettres ;
>
> — les chiffres arabes ;
>
> — le point pour toute ponctuation ;
>
> — l'espace pour séparer les mots, les chiffres et de façon générale des élé- ments quelconques du texte.
>
> Dans des limites aussi étroites, on ne peut mettre à profit la différence entre majuscules et minuscules, entre romain et italique, entre lettres grasses ou mai- gres etc. On ne peut pas non plus faire usage de lettres qui portent des accents, puisqu'on ne dispose que de capitales toutes nues.[24]

Le limitazioni dell'*hardware* negli anni '60 imponevano l'utilizzo di convenzioni, come il segno $ per indicare una maiuscola o una virgola separata da spazi per il ritorno a capo. Froger decide di conservare la dif- ferenza tra maiuscole e minuscole, ma considera irrilevanti gli accenti e la punteggiatura.[25] Nel tempo sono venuti meno i limiti tecnici per

[24] Dom J. Froger, «La collation des manuscrits à la machine électronique», *Bulletin d'information de l'Institut de Recherche et d'Histoire des Textes*, 13 (1964-1966), pp. 135- 171: 144.

[25] Ivi, p. 145.

la rappresentazione di accenti, punteggiatura e maiuscole,[26] ma i pro- grammi di collazione hanno continuato a dare la possibilità di neutra- lizzare queste componenti del testo: a seconda della lingua, del corpus e del fine della collazione, gli accenti, la punteggiatura e la distinzione tra maiuscole e minuscole possono essere conservati o rimossi; nei dif- ferenti casi, la migliore strategia può essere individuata empiricamente tramite prove sullo stesso corpus con parametri diversi. Gli esempi rimontano agli anni '70: «[t]he editor may specify in the STRIP para- meter which punctuation marks and special symbols the module is to ignore», scrive Penny Gilbert nel 1979, presentando il sistema Collate per le edizioni di testi in prosa (da non confondere con quello elaborato da Peter Robinson dieci anni dopo).[27] Questi meccanismi di prepara- zione possono valere per tutti i testi, e non solo per quelli in lingue ad alto tasso di variabilità come quelle medievali.

Altri tipi di normalizzazione, che rispondono alle caratteristiche degli specifici *corpora* in questione, sono stati utilizzati nel tempo. Tra i primi casi di normalizzazione *ad hoc* ricordiamo il lavoro di Georgette Silva e Harold Love, che utilizzano come caso di studio il *Prelude* di Wordsworth, in cui prima della comparazione tra due versi «[i]n order to avoid the most common types of variation, such as liv'd/lived, or ;/, or capitalized/non-capitalized words, the subroutine first eliminates letters "e" and "s", the $-sign, blanks, and all punctuation marks from both lines».[28] Un approccio simile è quello di Waite: «if one of the texts systematically represents consonantal I and U by I and U while the other uses J and V for the same purpose, this change can readily be allowed for in the programs, so that only significant differences are marked».[29] In entrambi i casi, il filologo identifica i tratti formali non significativi e li elimina dal testo, oppure istruisce il programma per ignorarli, in modo da ridurre il rumore nei risultati.

Il programma che introduce meccanismi più sofisticati per gestire la varianza formale è Collate, sviluppato da Peter Robinson insieme a diversi

[26] Per la codifica dei caratteri, valgono le specificazioni di Unicode, cf. http://www. unicode.org/ (ultimo accesso 30 gennaio 2019).

[27] P. Gilbert, «The preparation of prose-text editions with the "Collate" system», in *La pratique des ordinateurs dans la critique des textes*, Paris, Ed. du CNRS, 1979, pp. 245-254: 247.

[28] G. Silva, H. Love, «The identification of text variants by computer», *Information Storage and Retrieval*, 5/3, (1969), pp. 89-108: 93. Il segno $ viene usato dopo la lettera per indicare che si tratta di una maiuscola; cf. ad esempio v. 90: «T$O GRATIFY THE PALATE WITH REPASTS» ivi, p. 94.

[29] Waite, «Two programs for comparing texts», p. 242.

collaboratori,[30] inizialmente concepito per l'edizione di un testo islan- dese medievale.[31] Il programma prevede tre modi per gestire la varianza formale: un meccanismo di normalizzazione che interviene prima del- l'allineamento e due procedimenti che avvengono al momento dell'al- lineamento o dopo. Questi ultimi sono il cosiddetto 'fuzzy match', che consente di allineare *token* simili ma non uguali;[32] e la funzione 'defvars', che fa ricorso ad un file esterno in cui sono registrati *token* graficamente dissimili ma equivalenti, come, ad esempio, gli avverbi di negazione 'mie' e 'pas' in antico francese.[33] La normalizzazione che avviene prima del- l'allineamento, invece, prevede l'utilizzo di un dizionario di varianti da scartare, approccio utilizzato ancora oggi. Il dizionario è costituito da una serie di righe: per ognuna di esse, la prima parola è la forma nor- malizzata, seguita da tutte le forme corrispondenti nei manoscritti. Per preparare il testo alla collazione, si crea una copia normalizzata dei testi- moni, controllando ogni parola nel dizionario e, se la parola è presente, sostituendo la forma originale con quella normalizzata. Il testo prodotto risulterà un'aberrazione filologica: ma si tratta, come detto, di una copia di lavoro allestita per la collazione; le corrispondenze tra forme originali e normalizzate sono presenti ad ogni momento e possono facilmente essere recuperate in ogni momento del processo. Un'altra questione apparen-

[30] P. Robinson, «The Collation and Textual Criticism of Icelandic Manuscripts (1): Collation», *Literary and Linguistic Computing*, 4/2 (1989), pp. 99-105. Annunciato nella mailing-list ('discussion group' o 'international seminar') *Humanist* nel 1991, lo svi- luppo di Collate ha dato vita a tre distribuzioni principali del software; le funzionalità e le caratteristiche qui discusse sono dunque frutto di un processo durato vent'anni. Cf. P. Robinson, «Rationale and Implementation of the Collation System», in *The Miller's Tale on CD-ROM. The Canterbury Tales Project*, Saskatchewan, Scholarly Digital Editions, 2004; https://dhhumanist.org/Archives/Virginia/v04/1240.html; https://scholarlydigitaleditions.blogspot.com/2014/09/the-history-of-collate.html>; https://scholarlydigitaleditions.blogspot.com/2014/09/collate-2-and-design-for-its-successor.html (ultimo accesso 30 gennaio 2019).

[31] Le varianti grafiche nel testo studiato sono davvero molte: «[t]he anarchy is such that I estimate that there are over fourteen million possible spellings of "FjolsviSr", the name of one of the protagonists of Fj [*Fjölsvinnsmál*]. Thankfully, we get only 97 different spellings, in the 243 occurrences of the name in the 37 manuscripts in which it occurs» Robinson, «The Collation and Textual Criticism of Icelandic Manuscripts», p. 100.

[32] Il parametro 'fuzzy match' di Collate corrisponde concettualmente al 'near match' di CollateX. Il secondo è basato sulla 'edit distance' tra i due *token* (cf. nota 20). Per il 'fuzzy match', il meccanismo è diverso; cf. Robinson, «The Collation and Textual Criticism of Icelandic Manuscripts», p. 103: «For FUZZYMATCH I devised a mathe- matical formula in which the variables are the number of letters in the two words, the letters themselves and their order».

[33] L'esempio proposto da Robinson è quello degli avverbi di congiunzione 'ok' e 'en' in islandese.

temente spinosa è quella delle forme normalizzate: per una lingua non standardizzata, non è facile, né corretto, scegliere una forma canonica. In effetti, come ricorda Robinson, «the normalisation adopted is quite unimportant, as it is only to be used for comparing readings between manuscripts. All that matters is that it be consistent».[34] L'importante non sarà dunque stabilire quale forma ritenere come canonica, ad esempio 'scripse' o 'scrisse'; ma assicurarsi che entrambe rimandino ad una forma canonica arbitraria ('scripse' o 'scrisse' o il disegno di una penna) e che possano essere quindi riconosciute dal programma come equivalenti.

Questo tipo di normalizzazione tramite dizionario, estremamente pre- ciso ma anche oneroso da implementare, è quello che oggi viene conside- rato standard. E che, insieme al costo della trascrizione, frena l'adozione della collazione automatica, come ribadito in un articolo recente: «l'at- tuabilità del piano di lavoro ... è contrastata dalle difficoltà insite nella codifica di un numero così ampio di testimoni, il cui testo deve essere standardizzato sufficientemente da essere collazionato da una macchina virtuale».[35]

Nuove risorse e nuove sfide

In questa sezione vedremo come le difficoltà discusse sopra possono essere almeno in parte sormontate grazie all'utilizzo di nuove risorse, occupandoci dello stato dell'arte recente per quanto riguarda la norma- lizzazione e la gestione delle varianti nella collazione di testi medievali.

La sfida da affrontare con nuove risorse è quella di mettere insieme le due componenti problematiche della collazione di testi medievali, ovvero la normalizzazione dei testi per ottenere un allineamento cor- retto e la grande quantità di varianti formali, e di utilizzare la prima per controllare la seconda: distinguere, quindi, categorie di varianti attra- verso l'informazione usata per la normalizzazione. Il procedimento sembra tautologico – le varianti formali sono quelle definite durante la fase di normalizzazione come varianti formali –, ma diventa vantag- gioso se le varianti sono identificate tramite un'annotazione linguistica automatica, come proposto di seguito.[36]

[34] Robinson, «The Collation and Textual Criticism of Icelandic Manuscripts (1): Collation», p. 100.

[35] S. Bertone, et al., «Verso il trattamento automatico della tradizione manoscritta del liber di Catullo», *Umanistica Digitale*, 2/3 (2018).

[36] Procedimenti simili sono studiati da Jean-Baptiste Camps e Lucence Ing, che ringrazio per aver messo a disposizione il loro lavoro in corso. Cf. J.-B. Camps, L. Ing,

Per proporre un esempio, dovremo innanzitutto specificare cosa inten- diamo per varianti formali: includeremo qui solamente le varianti grafi- che ('scripse' vs 'scrisse') e fonetiche ('bon' vs 'buon'). Prima di passare all'utilizzo dell'annotazione linguistica automatica, presentiamo il pro- cedimento, attraverso un esempio in cui la normalizzazione è effettuata manualmente.

Consideriamo il verso «soli eravamo e sanza alcun sospetto» (Dante, *Commedia*, ed. Petrocchi, Inf. V, 129) nei tre manoscritti di mano di Giovanni Boccaccio, che conservano «sança alcun sospecto» (Toledano) e
«sença alcun sospecto» (Riccardiano e Chigiano).[37] Per collazionare i testi, essi andranno normalizzati; definiremo quindi un dizionario di arbitrarie forme canoniche, in cui alle forme 'sança' e 'sença' corrisponde la forma canonica 'SENZA'; le copie di lavoro per la collazione registreranno dunque
«SENZA alcun sospecto» e «SENZA alcun sospecto». Il programma potrà allinearli correttamente e non riconoscerà nessuna differenza. In questo modo abbiamo ottenuto l'allineamento ottimale ed escluso la varianza gra- fica. Possiamo ora recuperare l'informazione usata per la normalizzazione e creare delle semplici regole: quando due o più *token* allineati hanno la stessa forma canonica ma una forma originale diversa, i *token* sono varianti formali; quando hanno diverse forme canoniche e diverse forme originali, i *token* sono varianti di sostanza; quando hanno la stessa forma canonica e originale, i *token* sono invarianti. I *token* 'sança' e 'sença' saranno quindi rico- nosciuti come varianti formali, perché la loro forma canonica 'SENZA' coin- cide nei testimoni ma le forme originali differiscono. L'applicazione delle regole appena descritte avviene dopo l'allineamento, durante la quarta fase, quella del *feedback*, ed è un esempio di analisi che serve ad arricchire l'*output*.

Collation assistée par ordinateur de témoins de textes en ancien français. 19 ottobre 2018,
<https://halshs.archives-ouvertes.fr/halshs-02023936/document>; L. Ing, *Outils numéri- ques pour corpus diachronique: quelques disparitions lexicales en œuvre dans le micro-corpus du Lancelot en prose*, mémoire de master 2 «Humanités numériques et computa- tionnelles», dir. J.-B. Camps e F. Duval, Université Paris Sciences & Lettres, 2018.

[37] Per uno studio dei manoscritti boccacciani della *Commedia*, cf. S. Tempestini, «Boccaccio copista e interprete della "Commedia". La "Commedia" nei codici Toledano 104.6, Riccardiano 10135, Chigiano L VI 213: alcuni dati sulla variantistica», in *Intorno a Boccaccio / Boccaccio e dintorni. Atti del Seminario internazionale di studi (Certaldo Alta, Casa di Giovanni Boccaccio, 9 settembre 2015)*, a cura di S. Zamponi, Firenze, University Press, 2015, pp. 89-107; S. Tempestini, «Boccaccio copista e editore della "Commedia". Per un'analisi della variantistica», *Critica del testo*, 21/2 (2019). Il corpus è parzialmente interrogabile all'indirizzo http://boccacciocommedia.unil.ch/ (ultimo accesso 30 gen- naio 2019); cf. S. Tempestini, E. Spadini, «Querying Variants: Boccaccio's "Commedia" and Data-Models», *Digital Medievalist*, XII (2019).

L'esempio è servito a mostrare il procedimento, ma se la normalizza- zione deve essere effettuata manualmente l'intero procedimento perde di interesse perché estremamente oneroso; di fatto, poi, le varianti formali sarebbero identificate in fase di normalizzazione. Il vantaggio risiede invece nell'utilizzare strumenti computazionali per la normalizzazione, in modo da automatizzare, almeno in parte, questo passaggio. Come detto, la normalizzazione serve a stabilire una forma canonica arbitra- ria: essa può essere costituita dall'equivalente moderno, da un'immagine, ma anche dalle proprietà linguistiche della parola; l'utilizzo del lemma e della categoria grammaticale come forma di normalizzazione è stato già applicato con successo.[38] Inoltre, si possono oggi utilizzare risorse che sono andate sviluppandosi negli ultimi decenni per l'analisi linguistica delle lingue medievali. Si tratta in particolare di modelli da usare per la lemmatizzazione e l'annotazione morfologica, disponibili ad oggi per varie lingue.[39] Per il francese antico, su cui ci concentreremo qui, due modelli per il software TreeTagger[40] sono a disposizione, basati sulle risorse del *Nouveau Corpus d'Amsterdam* e della *Base de français médiéval*, riuniti nel *Medieval French Language Toolkit*.[41]

Il procedimento rimane lo stesso rispetto a quello già descritto, salvo per l'utilizzo di strumenti automatici per l'annotazione linguistica durante la fase di preparazione del testo. Vediamolo in un altro esempio.

Immaginiamo tre manoscritti, che inizino in questo modo: A 'Grant fu la ioie'; B 'Or dist li contes que grans fu la joie', C 'Grans fu la joie'.

[38] Cf. van Dalen-Oskam, «The Secret Life of Scribes».

[39] Sono discusse risorse per arabo, cinese, olandese, inglese, francese, tedesco, latino, greco antico, portoghese e lingue nordiche in Piotrowski, *Natural language processing for historical texts*, Capitolo 8. Per lo spagnolo, cf. G. Boleda, «Extending the tool, or how to annotate historical language varieties», in *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 2011, pp. 1-9; J. Porta, et al., «Edit transducers for spelling variation in Old Spanish», in *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013; May 22-24; 2013; Oslo; Norway. NEALT Proceedings Series 18*, Linköping University Electronic Press, 2013, pp. 70-79.

[40] H. Schmid, «Probabilistic Part-of-Speech Tagging Using Decision Trees», in *Pro- ceedings of International Conference on New Methods in Language Processing*, 1994. Il pro- gramma è disponibile online all'indirizzo <http://www.cis.uni-muenchen.de/~schmid/ tools/TreeTagger/>. Il programma ha un corrispondente Python wrapper, che ne può facilitare l'integrazione con CollateX: <https://pypi.org/project/treetaggerwrapper/> (ultimo accesso, 30 gennaio 2019).

[41] <https://github.com/sheiden/Medieval-French-Language-Toolkit>. Il modello della *Base de français médiéval* è disponibile all'indirizzo http://bfm.ens-lyon.fr/spip.php?article324 (ultimo accesso 30 gennaio 2019).

Senza nessuna forma di normalizzazione, CollateX produce il risultato riprodotto nella TAB. 1.[42]

TABELLA 1
Risultati della collazione senza normalizzazione ('var' = varianti; 'inv' = invarianti).

A	B	C	D
Grant	Or	Grans	***var***
-	dist	-	***var***
-	li	-	***var***
-	contes	-	***var***
-	que	-	***var***
-	grans	-	***var***
fu	fu	fu	***inv***
la	la	la	***inv***
ioie	joie	joie	***var***

Se invece neutralizziamo la differenza tra maiuscole e minuscole, si risolve la posizione di 'Grans' in C, ma l'allineamento rimane poco sod- disfacente per A, come mostrato nella TAB. 2, perché il programma non identifica la vicinanza tra 'grant' e 'grans'.

TABELLA 2
Risultati della collazione dopo la neutralizzazione di maiuscole e minuscole ('var' = varianti; 'inv' = invarianti).

A	B	C	D
Grant	Or	-	***var***
-	dist	-	***var***
-	li	-	***var***

[42] I risultati riprodotti in questa e nelle successive tabelle sono ottenuti con CollateX 2.1.3 (Python). Si noti che la demo di CollateX disponibile all'indirizzo https://collatex.

-	contes	-	***var***
-	que	-	***var***
-	grans	Grans	***var***
fu	fu	fu	***inv***
la	la	la	***inv***
ioie	joie	joie	***var***

L'analisi linguistica automatica, invece, riconoscerà che 'Grant', 'grans' e 'Grans' sono aggettivi e che il lemma corrispondente è 'grant'; così come che 'ioie' e 'joie' sono sostantivi, il cui lemma corrispondente è 'joie'. La collazione si effettuerà quindi sui *token* normalizzati, ovvero sull'anno- tazione linguistica, come mostrato nella TAB. 3. I risultati saranno resi utilizzando invece le forme originali: si ottiene quindi la TAB. 4.

TABELLA 3
Risultati della collazione dopo la normalizzazione ('var' = varianti; 'inv' = inva- rianti): token normalizzati.

A	**B**	**C**	**D**
-	avverbio, *or*	-	***var***
-	verbo, *dire*	-	***var***
-	articolo, *il*	-	***var***
-	sostantivo, *conte*	-	***var***
-	congiunzione, *que*	-	***var***
aggettivo, *grant*	aggettivo, *grant*	aggettivo, *grant*	***inv***
verbo, *estre*	verbo, *estre*	verbo, *estre*	***inv***
articolo, *il*	articolo, *il*	articolo, *il*	***inv***
sostantivo, *joie*	sostantivo, *joie*	sostantivo, *joie*	***inv***

net/demo/ (ultimo accesso 30 gennaio 2019) utilizza invece la versione Java, che prevede la neutralizzazione delle maiuscole come normalizzazione di default.

TABELLA 4
Risultati della collazione dopo la normalizzazione (‘var’ = varianti; ‘inv’ = inva- rianti): token originali.

A	B	C	D
-	Or	-	***var***
-	dist	-	***var***
-	li	-	***var***
-	contes	-	***var***
-	que	-	***var***
Grant	grans	Grans	***inv***
fu	fu	fu	***inv***
la	la	la	***inv***
ioie	joie	joie	***inv***

Si noterà che si è passati da due a quattro righe invarianti. Non solo, quindi, '*Grant*', '*grans*' e '*Grans*' sono stati correttamente allineati, ma, come '*ioie*' e '*joie*', non sono più considerati varianti. Possiamo a questo punto applicare le regole enunciate sopra e ottenere il risultato rappre- sentato nella TAB. 5.[43]

TABELLA 5
Risultati della collazione dopo la normalizzazione e la distinzione delle varianti ('var sost' = varianti di sostanza; 'var form' = varianti di forma; 'inv' = invarianti).

A	B	C	D
-	Or	-	***var sost***
-	dist	-	***var sost***
-	li	-	***var sost***

[43] Una visualizzazione a mezzo di colori e altri espedienti grafici permetterebbe di cogliere le categorie espresse nell’ultima colonna a destra a colpo d’occhio. Export dei risultati in altri formati, come JSON e XML/TEI, sono configurabili.

-	contes	-	***var sost***
-	que	-	***var sost***
Grant	grans	Grans	***var form***[44]
fu	fu	fu	***inv***
la	la	la	***inv***
ioie	joie	joie	***var form***

Un altro breve esempio, tratto dal *Lancelot en prose*, prevede tre manoscritti che leggono rispettivamente: «A ge te conois mielz que tu ne conois moi»; B «Artu je te conois miels que tu ne fes moi», C «Artus je te conois mult miaus que tu ne fas moi».

La collazione senza nessun tipo di normalizzazione produce i risul- tati, per molti versi insoddisfacenti, presentati nella TAB. 6.

TABELLA 6

Risultati della collazione senza normalizzazione ('var' = varianti; 'inv' = invarianti).

A	**B**	**C**	**D**
ge	Artu	Artus	***var***
-	je	je	***var***
te	te	te	***inv***
conois	conois	conois	***inv***
mielz	miels	mult	***var***
-	-	miaus	***var***
que	que	que	***inv***
tu	tu	tu	***inv***
ne	ne	ne	***inv***

[44] L'aggettivo *grant* al femminile, seppur caso soggetto, può non mutare la termina- zione, dunque possiamo considerare la differenza di tipo fonetico e non morfologico.

conois	fes	fas	***var***
moi	moi	moi	***inv***

Reiterando il procedimento descritto sopra, ad ogni *token* viene asse- gnato un lemma e una parte del discorso, che faranno ufficio di forma canonica per l'allineamento, producendo infine i risultati visualizzati nella TAB. 7.

TABELLA 7
Risultati della collazione dopo la normalizzazione e la distinzione delle varianti ('var sost' = varianti di sostanza; 'var form' = varianti di forma; 'inv' = invarianti).

A	**B**	**C**	**D**
	Artu	Artus	***var sost***
ge	je	je	***var form***
te	te	te	***inv***
conois	conois	conois	***inv***
-	-	mult	***var sost***
mielz	miels	miaus	***var form***
que	que	que	***inv***
tu	tu	tu	***inv***
ne	ne	ne	***inv***
conois	fes	fas	***var sost***
moi	moi	moi	***inv***

L'approccio descritto tramite questi esempi permette di risolvere alcuni degli aspetti problematici della collazione di testi medievali, grazie alla sua capacità di limitare gli effetti dell'instabilità linguistica sui risultati dell'al- lineamento e alla possibilità di riutilizzare l'informazione per gestire la grande quantità di varianti tramite categorie.

La facilità di implementare questo tipo di soluzione dipende in parte dall'abilità del programma di gestire le informazioni per ogni *token* e, in particolare, di tenere in memoria la forma originale insieme alla forma normalizzata. CollateX offre un'architettura ideale, dato che per ogni *token* sono previste due proprietà per dare spazio alle forme canoniche e originali. L'efficienza del procedimento riposa invece soprattutto sul- l'accuratezza dell'analisi linguistica: se i valori per il software TreeTagger possono essere superiori al 95%,[45] essi possono diminuire nel caso delle lingue medievali proprio a causa dell'estesa varianza ortografica. Ulteriori ricerche e l'applicazione a corpora diversi serviranno per misurare l'effi- cacia delle proposte qui delineate.[46]

Conclusioni

In questo articolo si fornisce un panorama critico di approcci alla col- lazione automatica, menzionando implementazioni del passato e solu- zioni recenti. Per quanto riguarda i testi medievali, si è visto come l'in- stabilità linguistica – principalmente grafica e fonetica, ma non solo – mette a repentaglio la possibilità di usare strumenti automatici per la collazione. Gestire la diffusa varianza formale, declinata a seconda degli interessi di ricerca, è importante per ottenere un allineamento corretto e limitare il 'rumore' nei risultati.

Tra i meccanismi passati in rassegna, figura il 'near-' o 'fuzzy-match'. La necessità di trovare il giusto parametro, che potrebbe variare caso per caso anche all'interno dello stesso testo, rende però questo tipo di approccio difficilmente applicabile su larga scala.

L'altra possibilità esplorata, e che acquista interesse alla luce degli svi- luppi nel trattamento automatico di lingue storiche, è quella della nor- malizzazione. Essa può essere di diversa natura e permette dunque una grande flessibilità: ogni elemento del linguaggio può essere oggetto di normalizzazione, tanto i tratti fonetici quanto gli aspetti semantici, ad

[45] Cf. H. Schmid, «Improvements in Part-of-Speech Tagging with an Application to German», in *Proceedings of the ACL SIGDAT-Workshop*, Dublin, 1995.

[46] Insieme a Jean-Baptiste Camps e Lucence Ing stiamo sviluppando un'implementazione dell'approccio qui presentato; i primi risultati si trovano in: J.-B. Camps,
L. Ing e E. Spadini, «Collating Medieval Vernacular Texts. Aligning Witnesses, Classifying Variants», in *Digital Humanities Conference 2019 (Utrecht)*, https://hal.archives-ouver- tes.fr/hal-02268348. Cf. anche https://github.com/CondorCompPhil/falcon. Lavori preli- minari si trovano in https://github.com/elespdn/collation_spelling.

esempio tramite la neutralizzazione dei sinonimi. La normalizzazione può essere effettuata manualmente, come è avvenuto fino ad oggi, o auto- maticamente, come ormai è possibile. Nel primo caso, uno sforzo con- siderevole deve essere dedicato alla creazione di un dizionario di cor- rispondenze. Nel secondo caso, l'utilizzo di strumenti di trattamento della lingua per normalizzare e annotare il testo riduce l'impegno richie- sto e permette un ampio spettro di analisi: l'annotazione descritta nel- l'esempio qui sopra si riduce al lemma e alla categoria grammaticale, che permettono di identificare varianti grafiche e fonetiche, applicando semplici regole; un'annotazione più ricca, che comprendesse anche dati morfologici e semantico-lessicali, permetterebbe di creare nuove catego- rie di varianti corrispondenti.[47] Bisognerà senz'altro tenere a mente che gli strumenti per l'analisi linguistica della lingue medievali sono ancora in corso di sviluppo: si potrebbe immaginare un circolo virtuoso in cui i testi annotati e manualmente controllati possano servire come 'training corpus' per migliorare le risorse linguistiche, e così via.

L'utilizzo di una serie di software, uno dopo l'altro - in una 'pipeline' –, necessita poi di attenzione dal punto di vista dell'interoperabilità dei formati. L'utilizzo di standard e la semplificazione sono dunque fonda- mentali: formati come XML, CSV e TXT, a seconda della struttura dei dati, corrispondono ad entrambi i requisiti.

In questo stato dell'arte si sono discussi brevemente i limiti all'adozione della collazione automatica. Esistono però anche dei vantaggi, da con- siderare in relazione al circolo virtuoso appena evocato e al cambiamento di paradigma menzionato da Ott: la collazione automatica è riproducibile ed esplicita. Quest'ultimo punto si riferisce alla normalizzazione menta- le implicita che applica chi collaziona manualmente, registrando solo le varianti significative; la collazione automatica, tramite le varie operazioni descritte nel modello di Göteborg, rende invece esplicito ogni passaggio. Ognuno di essi può essere inoltre verificato, nel caso in cui si volesse effet- tuare un controllo, ma anche per cambiare approccio in corso di rotta: basterebbe rieseguire la collazione, con parametri diversi. I testi delle tra- scrizioni, inoltre, possono essere riusati, come detto, per finalità diverse.

Infine, consideriamo le osservazioni di Robinson che, in termini quasi provocatori, sottolinea:

[47] La proposta è abbozzata in Tempestini, Spadini, «Querying Variants: Boccaccio's "Commedia" and Data-Models». Nella stessa direzione si muovono anche le ricerche di Camps e Ing, che grazie all'utilizzo del software Pandora per l'annotazione linguistica ottengono risultati completi di analisi morfologica, da poter utilizzare per l'identifica- zione di varianti (cfr. nota 34).

you simply cannot judge what might or might not be significant until you have seen every word of every manuscript. The best guide to what was significant
was not theory but the manuscripts themselves.[48]

In un già citato articolo, Leonardi e Morato riflettono sulla distin- zione tra i fenomeni di superficie linguistica e i fenomeni testuali e tra le metodologie per trattare gli uni e gli altri nella pratica ecdotica; gli autori fanno riferimento all'edizione Segre del *Bestiaire d'amours*, in particolare all'identificazione di categorie di varianti tramite liste di occorrenze, e aggiungono:

C'est ce type d'analyse qui devrait être élargie et approfondie, de façon à rendre moins aléatoire la distinction dont il est question Or, la seule voie pour
tenter de mieux définir ce que nous attribuons à la surface linguistique est de recourir à l'expérience de la recensio, à savoir aux résultats de la collation et aux indications qui émergent de celle-ci concernant la mono- ou polygenèse et la distribution des variantes.[49]

Seppur gli autori non si riferiscono alla collazione automatica, proce- dimenti come quelli qui presentati si iscrivono nella direzione tracciata: lo studio della tradizione dei testi può trarre vantaggio da risultati della *recensio* espliciti, riproducibili e riutilizzabili secondo il paradigma deli- neato dalla filologia computazionale.

ABSTRACT

Collation is one of the first philological activities for which the use of computers has been considered. Already during the 60's, algorithms for semi-automatic col- lation came to light. This article pursues their development, focusing on a spe- cific type of materials: medieval vernacular texts. The linguistic resources avai- lable nowadays for these materials allow to design a collation pipeline, following the architecture proposed by the Gothenburg model: the text of each witness is annotated with linguistic information; the alignment is made on the lemma, in order to neutralize the orthographic variation; eventually, the linguistic annota- tion is used to identify categories of variants. While most of the steps in this pipe- line have been conceived earlier in the history of semi- automatic collation, they were partially carried out manually; the possibility of automatize them might influence the adoption of computers for the collation of medieval vernacular texts.

[48] Robinson, «The Collation and Textual Criticism of Icelandic Manuscripts», p. 101.

[49] Leonardi, Morato, «L'édition du cycle de Guiron le Courtois», p. 472.

Keywords
Collation, textual criticism, digital philology.

ABSTRACT

La collazione è una delle prime attività filologiche per le quali è stato propo- sto l'uso del computer. Fin dagli anni '60 assistiamo al fiorire di algoritmi per la collazione detta semi-automatica. In questo articolo se ne segue la storia, con particolare attenzione alle problematiche dei testi medievali in volgare. Le risorse linguistiche ad oggi disponibili per alcune lingue rendono possi- bile la progettazione di una *pipeline* computazionale, basata sul modello di Gotheborg: il testo di ogni testimone viene annotato con informazioni lingui- stiche; l'allineamento si effettua sui lemmi, per neutralizzare la diffusa varianza ortografica; infine, le informazioni linguistiche sono recuperate per categoriz- zare le varianti. Sebbene la maggior parte dei trattamenti nella *pipeline* siano stati adottati in precedenza nella storia della collazione semi-automatica, un passo avanti è rappresentato dalla proposta di una maggiore automatizzazione qui avanzata grazie alle risorse oggi disponibili.

Keywords
Collazione, filologia, filologia computazionale.