Umanistica Digitale – ISSN: 2532-8816 – n. 16, 2023
L. Clemenzi, F. Fusco, D. Fusi, G. Lombardi – *Testi in maschera: nuovi strumenti per la sicurezza e l'analisi linguistica di* corpora *giuridici*
DOI: http://doi.org/10.6092/issn.2532-8816/15608

Testi in maschera: nuovi strumenti per la sicurezza e l'analisi linguistica di *corpora* giuridici

Laura Clemenzi

Università degli Studi della Tuscia
laura.clemenzi@unitus.it

Francesca Fusco

Università degli Studi di Padova
francesca.fusco@unipd.it

Daniele Fusi

Università degli Studi di Venezia Ca' Foscari - Venice Centre for digital and public humanities (VeDPH)
daniele.fusi@unive.it

Giulia Lombardi

Università di Genova
giulia.lombardi@edu.unige.it

Abstract

Il progetto Atti Chiari, volto a raccogliere il primo grande *corpus* italiano di atti di parte, presenta stringenti requisiti di ordine legale e numerose peculiarità sul piano della lingua e dei contenuti, che hanno reso necessario progettare e implementare una serie di processi e di strumenti *ad hoc*.

In particolare, al fine di eliminare ogni dato personale dai documenti, senza tuttavia distruggerne il tessuto linguistico e comprometterne la leggibilità, si è creata una procedura di pseudonimizzazione funzionale anche alla successiva indicizzazione e ricerca. La molteplicità dei metadati derivanti da questo processo e delle relative fonti converge poi in un sistema di ricerca basato su un motore specificamente disegnato per trattare testi in qualsiasi formato dotati di grandi quantità di annotazioni, anche relative a strutture testuali eterogenee e liberamente sovrapponibili. La combinazione di tutte queste strutture e dei loro metadati in una ricerca è resa

possibile da un approccio più astratto, dove il testo viene in certo modo smaterializzato in un insieme di oggetti dotati di metadati aperti, risultando in una modellazione modulare riflessa anche in una procedura di indicizzazione.

Parole chiave: AIUCD2022, linguistica giuridica, motore di ricerca, scrittura forense, pseudonimizzazione, Pythia, TEI

The Atti Chiari project, collecting the first large Italian corpus of judicial acts, presents strict legal requirements as well as many peculiarities in terms of language and content; to meet them, a number of processes and tools have been designed and implemented. The first issue is the requirement to remove any personal data from the documents, without however destroying their linguistic form, nor compromising their readability. To this end, a specific pseudonymisation procedure has been created which is also functional for subsequent indexing and linguistic research. The multiplicity of metadata resulting from this process and its sources then converge in a search system based on a search engine specifically designed to handle texts in any format with large amounts of annotations, even heterogeneous and freely overlapping text structures. The integration of these structures and their associated metadata in a search is enabled through a more abstract approach, in which the text is transformed into a collection of objects with open metadata, resulting in a modular modelling also reflected in a indexing procedure.

Keywords: AIUCD2022, legal linguistics, legal writing, pseudonymization, Pythia, search engine, TEI

1. Il progetto Atti Chiari[1]

Il Progetto di rilevante interesse nazionale (PRIN) 2017 "La chiarezza degli atti del processo (AttiChiari): una base di dati inedita per lo studioso e il cittadino", a cui collaborano linguisti e giuristi degli atenei di Genova, Firenze, Lecce e Viterbo, si è posto l'obiettivo di creare una nuova risorsa per una scrittura efficace degli atti processuali.[2] Dapprima è stato allestito un *corpus* sincronico di atti di parte rappresentativo, per tipologie testuali e provenienza geografica, delle diverse prassi di scrittura degli avvocati; successivamente è stata avviata la realizzazione di una base di dati interrogabile.[3]

Dal punto di vista scientifico, il progetto contribuisce a un significativo avanzamento della linguistica giuridica e della linguistica delle varietà dell'italiano; la base di dati interrogabile consente infatti una descrizione approfondita e dettagliata delle proprietà testuali, stilistiche, retoriche, pragmatiche, morfosintattiche e lessicali della varietà giudiziaria degli atti di parte,

[1] Il testo è stato concordato e rivisto da tutti gli autori; tuttavia, ai fini dell'attribuzione della paternità delle singole parti di cui si compone, vanno attribuiti, nell'ordine, a Laura Clemenzi il paragrafo 1, a Giulia Lombardi il paragrafo 2, a Francesca Fusco il paragrafo 3, a Daniele Fusi i paragrafi 4-13.

[2] Progetto finanziato dal Ministero dell'Istruzione dell'Università e della Ricerca, Prot. 2017BSECYX.

[3] Per maggiori dettagli sugli obiettivi del progetto e sulle procedure adottate, si rinvia agli interventi raccolti nel volume curato da Gualdo e Clemenzi ([21]). Altre informazioni riguardanti la composizione e le attività del gruppo sono disponibili nel sito ufficiale del progetto, https://attichiari.unige.it.

ancora poco frequentata negli studi linguistici.[4] La scelta di affrontare questo tema unendo competenze giuridiche e linguistiche dà al progetto «una forte impronta innovativa», e l'obiettivo di migliorare la comunicazione giudiziaria a favore del cittadino – che risulta, anche quando indiretto, il destinatario finale – rappresenta «una spinta d'impegno civile».[5]

Nella prospettiva della ricerca linguistica, nella fase di progettazione del motore di interrogazione della base di dati, come si spiegherà anche più avanti (v. *infra* par. 9), si è posta l'esigenza di poter filtrare le parole attraverso dati provenienti dai documenti originali (es. per corsivi, grassetti, ecc.), da processi automatici (es. il *Part Of Speech tagging*, in sigla POS, per il riconoscimento delle parti del discorso), da annotazioni mirate (es. per i forestierismi), e di poter filtrare i documenti attraverso i metadati, raccolti separatamente, relativi agli atti (es. tipo e sede dell'organo giurisdizionale, tipologia e data dell'atto, anno di nascita dell'avvocato che ha redatto l'atto, ecc.). Il motore, del quale in questo articolo verranno illustrate le principali caratteristiche, consente di utilizzare tutte queste informazioni: ad esempio, per cercare i participi presenti al fine di studiare un tratto tipico del linguaggio giuridico, l'uso del participio presente con valore verbale, è possibile chiedere al sistema di combinare la ricerca di "verbform" e di "tense" corrispondenti rispettivamente al modo participio e al tempo presente; o per cercare gli anglicismi al fine di verificarne l'impiego in base all'età degli avvocati, è possibile chiedere al sistema di filtrare le parole attraverso il marcatore dei forestierismi e al contempo di filtrare i documenti attraverso il metadato dell'anno di nascita (Figura 1).[6]

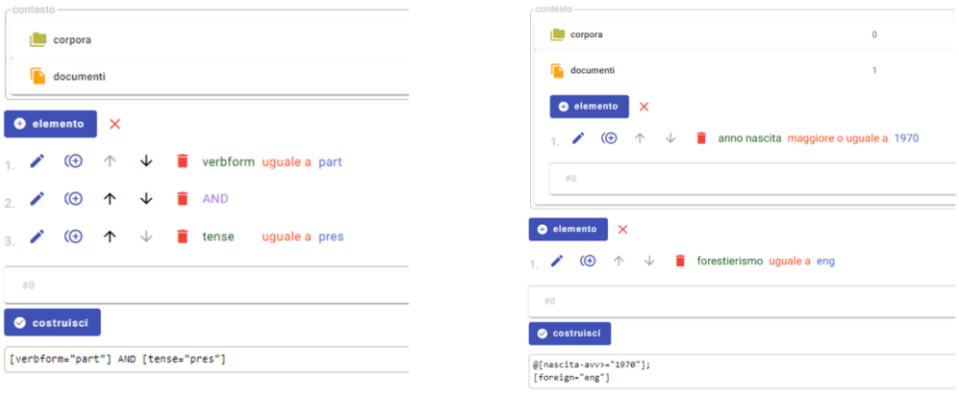


Figura 1: Esempi di ricerche nel corpus Atti Chiari con il motore di ricerca e l'interfaccia di interrogazione assistita: i controlli in alto costruiscono un'interrogazione a partire dai suoi termini, selezionando nomi, valori e operatori da appositi elenchi. La corrispondente interrogazione in forma testuale viene automaticamente costruita (sotto) ed eseguita. A destra, la stessa interfaccia mostra ulteriori filtri selezionati per i documenti in base ai loro metadati.

[4] Per alcuni primi studi sulla lingua degli atti di parte, cfr. [30], [26], [27], [4], [7], [22], [10], [3]. Alcune prime indagini sugli atti del *corpus* del progetto Atti Chiari sono in [11] e in [17].

[5] [23], 17-18.

[6] Per altri esempi di fenomeni linguistici ricercabili nella base dati Atti Chiari, cfr. [6].

I risultati sono sempre delle liste di concordanze nel formato KWIC (*Key Word In Context*): in ogni riga viene mostrato un contesto in cui ricorre la parola cercata, evidenziata al centro; cliccando sulle occorrenze della parola è possibile vedere contesti più ampi (Figura 2).

amm-vt-prf-sigl000-201604_01	di	accertamento	una	verifica	periodica	tendente	a	valutare	la	corretta	funzionalita
civ-ge-app-cit342-200911_01	composto	dalle	singole	unita	immobiliari	appartenenti	a	proprietari	diversi	li	nel
civ-ge-app-cit342-200911_01	sentenza	impugnata	n	i	fatti	costituenti	titolo	esecutivo	e	res	iudicata
civ-ge-app-cit342-200911_01	petitum	e	la	causa	pretendi	vertente	sull'	accertamento	del	medesimo	fatto
civ-ge-app-cit342-201102_01	per	le	altre	una	gratuita	derivante	da	non	meglio	precisati	usi
civ-ge-app-cit342-201301_01	in	pagamento	unitamente	ad	altre	riguardanti	forniture	diverse	dalla	linea	aloe
civ-ge-app-cit342-201301_01	doc	ns	produzioni	in	cuanto	avente	ad	oggetto	i	predetti	astucci

-- civ-ge-app-cit342-200911_01

B) Come si vede, quindi, sono stati in parte omessi ed in parte travisati dalla sentenza impugnata n. 2071/5795 i fatti costituenti titolo esecutivo e res iudicata, già da soli sufficienti a definire il contenzioso oggetto della sentenza impugnata, che non poteva disattendere le valutazioni già effettuate dal precedente giudicante e divenute – ripetesi –res iudicata per mancata impugnazione, né sostituirsi alle parti nel sollevare narrati profili di impugnazione, peraltro non intervenuta.

Figura 2: Estratto dei risultati della ricerca dei participi presenti nel corpus Atti Chiari. La visualizzazione per colonne mostra ogni parola trovata al centro, con intorno il suo contesto. A destra, il testo corrispondente, renderizzato in HTML (*HyperText Markup Language*) con evidenziazione del risultato selezionato.

2. I testi del *corpus*

Gli atti che compongono il *corpus* sono testi attraverso i quali, nei diversi momenti dell'*iter* processuale, gli avvocati argomentano, provano e perorano la causa dell'assistito di fronte al giudice. Sulla base delle classificazioni di Mortara Garavelli ([25], 19–34) e di Sabatini ([30]), si possono definire "testi applicativi", "mediamente vincolanti", ma compositi sotto diversi punti di vista: innanzitutto, all'interno degli scritti difensivi si trovano facilmente non solo citazioni di testi normativi e interpretativi appartenenti al linguaggio giuridico, ma anche referti medici, perizie tecniche, scambi epistolari, conversazioni informali e articoli informativi di vario tipo; in secondo luogo, ciascun atto si presenta suddiviso al suo interno da una sezione più spiccatamente narrativa nell'esordio, da una sezione argomentativa nella motivazione, e da una sezione prescrittiva nelle richieste istruttorie.[7] Si noti che non è raro rintracciare all'interno degli atti di parte proposizioni esclamative e interrogative quali espressioni di quella "modalità dialogica" che spinge gli avvocati a riprodurre nello scritto le dinamiche tipiche dell'udienza con il giudice.[8]

Ad oggi il *corpus* Atti Chiari è composto da circa 1,2 milioni di *token*, per un totale di 318 atti raccolti dalle sedi di Genova, Firenze, Viterbo e Lecce.[9] Gli avvocati che hanno collaborato al progetto, inviando uno o più atti, sono nati tra il 1932 e il 1990 e per più dei due terzi sono di sesso maschile; si tratta di professionisti iscritti ai Fori di Arezzo, Bari, Benevento, Bologna,

[7] Cfr. [32].

[8] Cfr. [22], 630.

[9] La raccolta e l'elaborazione dei dati qui presentati si devono all'intero gruppo di ricerca.

Brindisi, Firenze, Genova, La Spezia, Lecce, Lecco, Lucca, Milano, Modena, Napoli, Padova, Parma, Pisa, Pistoia, Prato, Roma, Savona, Torino, Treviso, Velletri, Viterbo.

Gli atti raccolti sono stati redatti tra il 1992 e il 2021 e afferiscono a tutti e tre i gradi di giudizio: 223 atti, pari al 70,1%, al primo grado; 83 atti, pari al 26,1%, al secondo grado; 12 atti, pari al 3,8%, al terzo grado.

292 atti, pari al 91,8%, sono relativi a procedimenti in materia civile (il dato include anche gli atti incardinati in alcune sezioni specializzate: 22 per la sezione famiglia, 24 per la sezione impresa, 17 per la sezione lavoro); dei restanti 26 atti, 22, pari al 6,9%, sono relativi a procedimenti in materia penale, mentre 4, pari all'1,3%, sono relativi a procedimenti in materia amministrativa.

Per ciascuna materia il gruppo di ricerca ha individuato le principali tipologie di atti; in totale nel *corpus* ne sono rappresentate oltre 50.

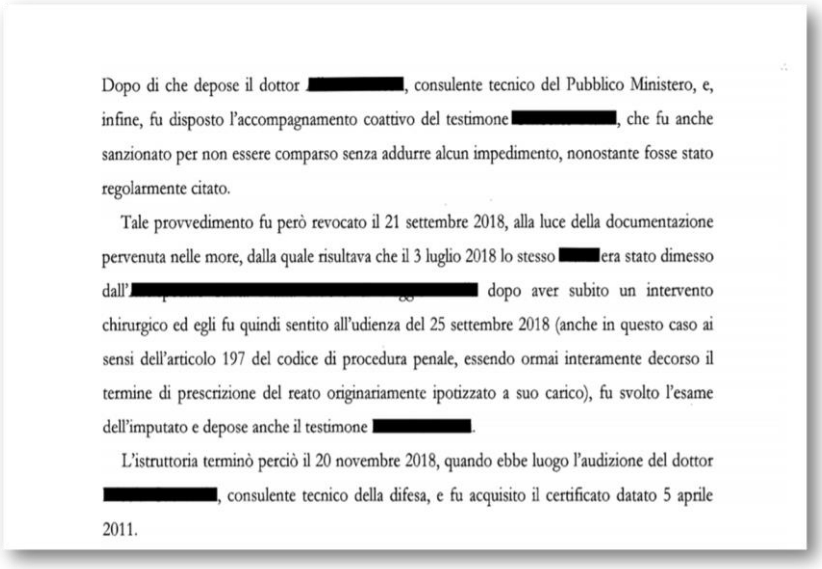
Le statistiche qui offerte sono state calcolate a partire dai metadati raccolti dal gruppo di ricerca, utilizzabili per filtrare le ricerche da condurre attraverso il motore (per alcuni esempi, v. Figura 1 e Figura 14).

3. Le esigenze del progetto e i requisiti del programma

I testi che compongono il *corpus* si contraddistinguono per contenere al loro interno dati personali (spesso anche sensibili), la cui diffusione violerebbe il diritto alla riservatezza delle parti, di eventuali terzi coinvolti e dei procuratori costituiti. Preliminarmente a qualsiasi tipo di studio e come requisito stesso per ottenere l'accesso agli atti si deve quindi procedere ad anonimizzare i documenti in modo da rendere irriconoscibili le vicende narrate e i soggetti coinvolti.

A oggi, le prassi di anonimizzazione in uso in Italia per riprodurre e diffondere testi giuridici che contengono dati personali (come ad esempio i provvedimenti giudiziari) consistono di norma nella mera eliminazione di tali dati tramite l'omissione o la cancellatura con tratti neri (Figura 3 e Figura 4), oppure nella loro sostituzione con asterischi, *omissis*, lettere, o altri segni grafici (Figura 5).[10]

[10] Si rinvia sul tema a [2], da cui sono tratti gli esempi nella Figura 3, nella Figura 4 e nella Figura 5. Si segnala qui che, in controtendenza, il Consiglio di Giustizia Amministrativa della Regione Sicilia con la sentenza n. 1134/2020 ha deciso di sostituire gli *omissis* con nomi di fantasia.

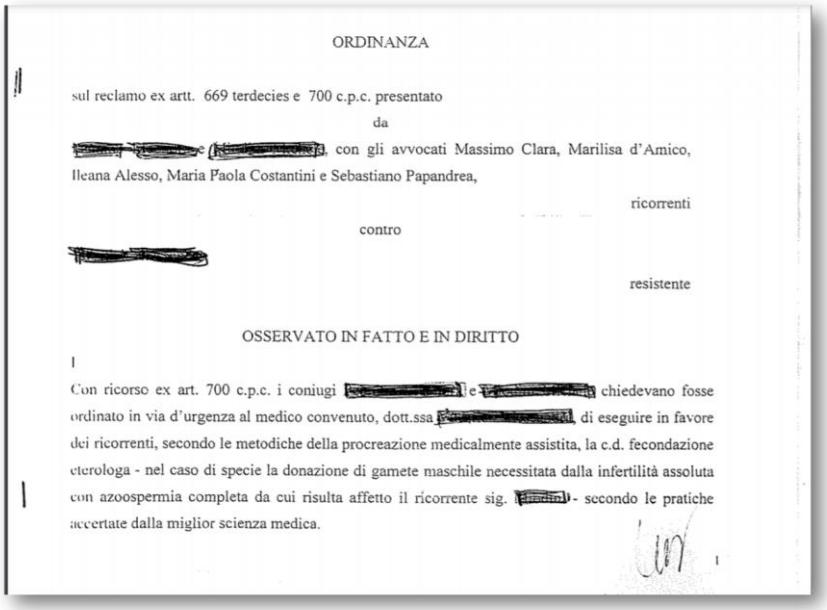


Dopo di che depose il dottor [redacted], consulente tecnico del Pubblico Ministero, e, infine, fu disposto l'accompagnamento coattivo del testimone [redacted], che fu anche sanzionato per non essere comparso senza addurre alcun impedimento, nonostante fosse stato regolarmente citato.

Tale provvedimento fu però revocato il 21 settembre 2018, alla luce della documentazione pervenuta nelle more, dalla quale risultava che il 3 luglio 2018 lo stesso [redacted] era stato dimesso dall'[redacted] dopo aver subito un intervento chirurgico ed egli fu quindi sentito all'udienza del 25 settembre 2018 (anche in questo caso ai sensi dell'articolo 197 del codice di procedura penale, essendo ormai interamente decorso il termine di prescrizione del reato originariamente ipotizzato a suo carico), fu svolto l'esame dell'imputato e depose anche il testimone [redacted].

L'istruttoria terminò perciò il 20 novembre 2018, quando ebbe luogo l'audizione del dottor [redacted], consulente tecnico della difesa, e fu acquisito il certificato datato 5 aprile 2011.

Figura 3: Esempio di atto anonimizzato tramite l'oscuramento dei dati personali con tratti neri ([2], 21).



ORDINANZA

sul reclamo ex artt. 669 terdecies e 700 c.p.c. presentato

da

[redacted], con gli avvocati Massimo Clara, Marilisa d'Amico, Ileana Alesso, Maria Paola Costantini e Sebastiano Papandrea,

ricorrenti

contro

[redacted]

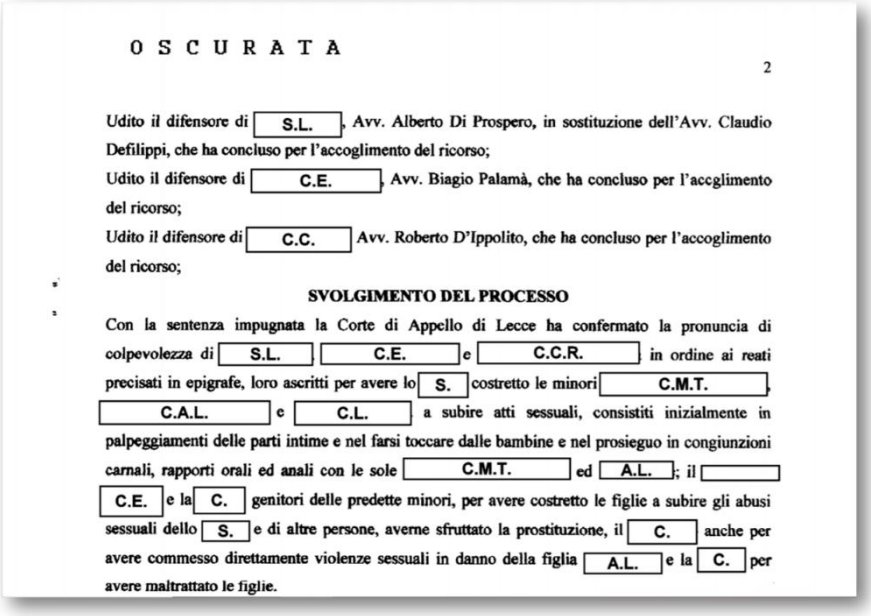
resistente

OSSERVATO IN FATTO E IN DIRITTO

I

Con ricorso ex art. 700 c.p.c. i coniugi [redacted] chiedevano fosse ordinato in via d'urgenza al medico convenuto, dott.ssa [redacted], di eseguire in favore dei ricorrenti, secondo le metodiche della procreazione medicalmente assistita, la c.d. fecondazione eterologa - nel caso di specie la donazione di gamete maschile necessitata dalla infertilità assoluta con azoospermia completa da cui risulta affetto il ricorrente sig. [redacted] - secondo le pratiche accertate dalla miglior scienza medica.

Figura 4: Esempio di atto anonimizzato tramite l'oscuramento dei dati personali con tratti di penna ([2], 21).



O S C U R A T A

2

Udito il difensore di S.L., Avv. Alberto Di Prospero, in sostituzione dell'Avv. Claudio Defilippi, che ha concluso per l'accoglimento del ricorso;

Udito il difensore di C.E., Avv. Biagio Palamà, che ha concluso per l'accglimento del ricorso;

Udito il difensore di C.C. Avv. Roberto D'Ippolito, che ha concluso per l'accoglimento del ricorso;

SVOLGIMENTO DEL PROCESSO

Con la sentenza impugnata la Corte di Appello di Lecce ha confermato la pronuncia di colpevolezza di S.L. C.E. e C.C.R. in ordine ai reati precisati in epigrafe, loro ascritti per avere lo S. costretto le minori C.M.T., C.A.L. e C.L. a subire atti sessuali, consistiti inizialmente in palpeggiamenti delle parti intime e nel farsi toccare dalle bambine e nel prosieguo in congiunzioni carnali, rapporti orali ed anali con le sole C.M.T. ed A.L.; il C.E. e la C. genitori delle predette minori, per avere costretto le figlie a subire gli abusi sessuali dello S. e di altre persone, averne sfruttato la prostituzione, il C. anche per avere commesso direttamente violenze sessuali in danno della figlia A.L. e la C. per avere maltrattato le figlie.

Figura 5: Esempio di atto anonimizzato tramite la sostituzione dei nomi e cognomi dei soggetti coinvolti con le relative iniziali ([2], 26).

Tali modalità anonimizzatorie vanno tuttavia a collidere con l'esigenza del linguista di avere testi massimamente leggibili e quanto più possibile completi, al fine di poter analizzare appieno le strategie usate dagli avvocati nel riferirsi alla parte assistita, alla controparte e agli altri soggetti del processo, sia all'interno di uno stesso atto, sia, in un'ottica di studio di tipo "verticale" e intertestuale, negli altri atti relativi allo stesso giudizio. Oscurando nomi, toponimi, date e ogni altro dato personale, verrebbe difatti meno la possibilità di individuare e distinguere le parti processuali e di ricostruire le vicende narrate: sarebbe quindi impossibile dipanare l'intreccio delle voci scriventi.[11]

Pertanto, al fine di garantire al contempo la piena leggibilità dei testi e la tutela dei dati personali in essi contenuti, nell'ambito del progetto Atti Chiari si è scelto di trattare gli atti raccolti con una tecnica di pseudonimizzazione.

Secondo la definizione di Elger et al. ([13], 233), «Pseudonymisation [...] is the step where a pseudonym or code is added to […] de-identified data». Si tratta di una tecnica esplicitamente prevista anche dal *Regolamento generale sulla protezione dei dati* (Reg. U.E. n. 2016/679), che la definisce all'art. 4, c. 5, come «il trattamento dei dati personali in modo tale che i dati personali non possano più essere attribuiti a un interessato specifico senza l'utilizzo di informazioni aggiuntive, a condizione che tali informazioni aggiuntive siano conservate separatamente e soggette a misure tecniche e organizzative intese a garantire che tali dati personali non siano

[11] Cfr. [16], 30.

attribuiti a una persona fisica identificata o identificabile»; essa è più volte richiamata all'interno del Regolamento proprio come misura di «garanzia adeguata» della riservatezza dei dati.[12]

Per le esigenze del PRIN Atti Chiari è stato ideato un metodo di pseudonimizzazione ([20]) ispirato ai modelli in Douglass et al. ([12]), Elger et al. ([13]) e Dalianis ([9]) per la pseudonimizzazione delle cartelle cliniche, e in Oksanen et al. ([29]) per la pseudonimizzazione degli atti del tribunale finlandese.

Il programma elaborato sostituisce, previa leggera marcatura manuale dei testi (cfr. *infra* par. 5),[13] i dati personali in essi contenuti con dati fittizi della stessa categoria, attingendo a liste predefinite per prenomi maschili e femminili, cognomi e toponimi, e modificando casualmente sequenze numeriche e alfanumeriche (quali date, targhe, fax, numeri di telefono, ecc.); sono garantite inoltre sostituzioni costanti e coerenti all'interno di uno stesso testo o all'interno di più testi afferenti allo stesso giudizio.

Si tratta di un sistema che garantisce la coerenza concettuale-semantica tra i dati originali e quelli fittizi e la coerenza morfosintattica dei dati fittizi con il contesto (il dato nuovo corrisponde in maniera univoca all'originale in tutte le occorrenze del testo e ne conserva il genere grammaticale per non alterare la morfosintassi della frase in cui è inserito),[14] permettendo così uno studio linguistico ottimale del testo.

Inoltre, dal momento che lo studio che si intende condurre sui testi è non solo di tipo linguistico, ma anche giuridico, si possono prevedere insiemi di metadati diversi a seconda degli scopi: se, ad esempio, l'analisi linguistica presuppone l'inserimento di metadati relativi al paratesto, quella giuridica richiede che la sostituzione delle date non pregiudichi la ricostruzione cronologica dei fatti (cfr. *infra* par. 4).

4. Annotazione e pseudonimizzazione

La pseudonimizzazione, in sostanza, trasforma le eliminazioni in sostituzioni, fornendo dati casuali al posto di dati che necessitano di essere oscurati. Pertanto, a differenza della mera anonimizzazione, questo processo implica una vasta gamma di scelte nell'indirizzare tali sostituzioni a seconda della natura dei testi e degli scopi del *corpus*. Si è dunque aggiunto al processo un ulteriore livello intermedio: anziché sostituire direttamente ogni porzione di testo con qualcos'altro, si inizia semplicemente con l'annotarlo. Ciò mantiene inalterato il testo, fatta eccezione per l'aggiunta di nuove informazioni per sue singole parti.

[12] Cfr., nello stesso regolamento, gli artt. 6, c. 4, 25, c. 1, 32, c. 1, 40, c. 2, 89, c. 1, oltre ai *considerando* 26, 28, 29, 75, 78, 85, 156.

[13] In questo contributo i termini *marcatura* e *annotazione* sono usati come sinonimi, ma più precisamente, nella linguistica dei *corpora*, per *marcatura* si intende «la codifica di metadati contestuali e oggettivi relativi ai testi» (es. titolo, autore, anno, suddivisione in paragrafi, presenza di immagini, ecc.), mentre con *annotazione* si fa riferimento a «informazioni di tipo linguistico-interpretativo […] corrispondenti ai diversi livelli dell'analisi linguistica» ([15], 19; cfr. anche [8], 73 e ss.).

[14] Cfr. [16], 30-34.

Lo scopo di questa annotazione è innanzitutto una classificazione predefinita dei dati personali: ogni porzione di testo viene contrassegnata di volta in volta come antroponimo, toponimo, data, numero, ecc. La strategia per rimuovere informazione parte quindi dalla loro aggiunta, e questo approccio è giustificato dalla necessità di attribuire flessibilità al processo, adattandolo a diversi livelli di granularità, a seconda delle diverse esigenze. Ad esempio, per un'analisi linguistica si possono semplicemente sostituire i numeri relativi a importi di denaro, numeri di telefono, ecc. con stringhe di numeri casuali della stessa lunghezza. Lo stesso procedimento può essere applicato alle date, qualunque sia la loro espressione (anno-mese-giorno, solo anno, solo mese, sia come nome che come numero, ecc.).

Tuttavia, se in seguito fosse necessario fornire una nuova versione del *corpus* per un pubblico diverso, come studiosi di diritto, questo tipo di sostituzione puramente casuale pregiudicherebbe aspetti essenziali nell'esposizione dei fatti: ad esempio, un importo casuale di denaro potrebbe risultare ridicolmente alto o basso; oppure, un fatto accaduto prima di un altro potrebbe essere collocato in una data successiva.

Simili effetti, potenzialmente indesiderati o distruttivi, rimangono possibili finché tutte queste sostituzioni siano eseguite su base puramente casuale; procedendo col sostituire direttamente e senza altri criteri il testo originale, senza una fase intermedia di annotazione, il *corpus* perderebbe irrimediabilmente informazione: una volta sostituita una data o un numero, non sarebbe più possibile tornare indietro.

Se invece si marca ogni somma di denaro o ogni data come tale, non vi saranno difficoltà nel ripetere il processo sugli stessi testi, ma con una configurazione diversa. Se quindi da una prospettiva puramente linguistica non ha senso cercare di preservare l'ordine di grandezza di una quantità numerica, o la cronologia relativa delle date, per un uso giuridico si potrebbe invece chiedere al programma di sostituire ogni importo di denaro con un valore casuale, ma dello stesso ordine di grandezza; o di sostituire ogni data con una casuale, mantenendo però la medesima distanza relativa da tutte le altre date nel documento.[15]

Suddividere dunque il processo in due fasi consente di annotare i testi originali una volta, per poi produrre un numero virtualmente illimitato di *output*, ciascuno mirato a scopi specifici.

5. Annotazioni di estensione

L'approccio illustrato sopra può essere esteso anche all'annotazione di caratteristiche che, pur non connesse al problema della rimozione dei dati personali, risultano assai utili dal punto di vista del loro successivo utilizzo.

[15] Come in molti altri casi, il sistema opera qui in base a dei parametri: per quanto riguarda specificamente le date, di default il programma le modifica sottraendo all'anno un valore casuale compreso fra due estremi (ad es. 5-20 anni), ma mantenendo il valore costante in tutte le date del documento in modo da conservare la successione cronologica dell'originale; ovviamente laddove si ritenesse che non vi fosse pregiudizio per la sensibilità dei dati, in funzione di usi diversi come uno puramente giuridico, le date potrebbero anche essere conservate intatte, o modificate con dei parametri costanti invece che casuali. All'opposto, si potrebbero invece sostituire con valori del tutto randomici. Quest'ultimo approccio naturalmente non sarebbe adatto laddove si volessero preservare le distanze cronologiche relative fra date diverse.

Ad esempio, contrassegnare una parte di un testo in lingua straniera specificandone la lingua (attribuendole un codice ISO 639) risulterebbe magari superfluo rispetto agli scopi della pseudonimizzazione. Tuttavia, una simile operazione è altamente auspicabile dal punto di vista linguistico, in quanto consente al programma di effettuare sostituzioni intelligenti, come ad esempio scegliere il nome di una città con lo stesso genere grammaticale dell'originale, con l'effetto di preservare la coerenza sintattica della frase (anzitutto in vista delle concordanze). Inoltre, a beneficio della ricerca successiva, anche solo a scopo di indicizzazione del testo, è essenziale poter distinguere tra lingue diverse; e dato che le espressioni latine sono un aspetto caratteristico di questo tipo di documenti, ciò risulta estremamente utile per chi vuole approfondire questo aspetto specifico.[16]

Sempre in questo ambito, in alcuni casi poi lo stesso comportamento del programma può essere personalizzato per adattarsi meglio alle esigenze linguistiche. Ad esempio, il programma di solito sceglie pseudonimi che iniziano con la stessa lettera di quelli originali, al fine di preservare il contesto fonosintattico originale. Anche in questo caso, si tratta di un espediente utile in special modo per questo tipo di testi, dove spesso si abusa delle consonanti "eufoniche" (es. "ed Asti" come "ed Empoli").

I diversi tipi di annotazione sono dunque attentamente progettati per adattarsi a una serie di scenari di utilizzo, garantendo al contempo la possibilità di generare *output* diversi in base ai requisiti dell'analisi. Questa annotazione è molto leggera e poco intrusiva: la porzione di testo da annotare è semplicemente racchiusa tra parentesi graffe, preceduta da un marcatore terminato da due punti, secondo la sintassi {marcatore:testo}, come nell'esempio mostrato in Figura 6.[17]

Di seguito sono elencati i marcatori utilizzati nel progetto Atti Chiari per identificare e sostituire i dati personali negli atti raccolti:

- a-f-f (*anthroponym*, *female*, *first name*) per antroponimi femminili;
- a-m-f (*anthroponym*, *male*, *first name*) per antroponimi maschili;
- a-l (*anthroponym*, *last name*) per i cognomi;
- j-f (*juridic person*, *female*) per le persone giuridiche di genere grammaticale femminile;
- j-m (*juridic person*, *male*) per le persone giuridiche di genere grammaticale maschile;
- t (*toponym*) per i toponimi;
- ad (*address*) per gli indirizzi;
- m (*e-mail*) per gli indirizzi e-mail;

[16] Si vedano ad esempio gli studi di Fusco ([17], [18]), rispettivamente sui forestierismi e i latinismi presenti negli atti del *corpus* del progetto Atti Chiari, resi possibili proprio dalla scelta di marcare le parti di testo in altre lingue.

[17] L'estratto rappresentato nella Figura 6 è un facsimile: per motivi di *privacy*, i dati originali, contrassegnati e inseriti tra parentesi graffe, sono stati sostituiti con dati fittizi a scopo esemplificativo.

- d (*date*) per le date;
- n (*number*) per cifre (es. numeri di telefono, somma di denaro, estensione di appezzamenti di terreno, ecc.);
- u per stringhe alfanumeriche (es. codici fiscali, abbreviazioni di province, targa, ecc.);
- x per i dati che non rientrano in nessuna delle categorie precedenti (sostituiti con ###).

Per ciascuna categoria di dati personali identificata da un marcatore, il programma sceglie in modo casuale una sostituzione da un elenco di migliaia di termini della stessa categoria (nomi, cognomi e toponimi maschili e femminili).[18] Pertanto, contrassegnando "Bellini" come "cognome" (a-l), come mostrato in Figura 6, si indica al programma di rimuovere questo elemento e sostituirlo con un altro cognome estratto casualmente dall'elenco fornito, avente la stessa iniziale "B".

GIUDICE DI PACE DI {t:TERMOLI}

ATTO DI CITAZIONE IN OPPOSIZIONE

A DECRETO INGIUNTIVO {f-lat:EX} ART. 645 C.P.C.

E CONTESTUALE ISTANZA DI SOSPENSIONE DELLA PROVVISORIA ESECUTIVITA'

La sottoscritta Avv. {a-f-f:Gianna} {a-l:Barbieri} del foro di {t:Termoli}, C.F.: {u:BRBGNN87S46G045T}, che rappresenta e difende ad ogni effetto di legge, in virtù di delega posta in calce al presente atto la **SIG.RA {a-l:BELLINI} {a-f-f:GIANCARLA}** nata a {t:Vicenza} il {d:24.11.1972} e residente in {t:Termoli}, Via {ad:Garibaldi, n. 4}, C.F.: {u:BLL GNC 72P52 R557X}, ed elettivamente domiciliata presso lo studio del predetto avvocato in {t:Termoli}, Via {ad:XX Settembre, n. 56}, con indicazione del n. fax al {n:0435/4530202}, PEC: {m:barbieri@pec.lambfa.it},

PREMESSO CHE

-in data {d:19.11. 2015} il Giudice di Pace di {t:Termoli} emetteva a favore di **{j-f:Beta NPL} S.p.a** *decreto ingiuntivo provvisoriamente esecutivo* n. {n:1234/2015} per la somma di € {n:3.400,00} oltre interessi di mora e spese della procedura;

-tale decreto, munito di formula esecutiva in data {d:03.12.2015}, veniva notificato in data {d:12.12.2015} alla Sig.ra {a-f-f:Giancarla} {a-l:Bellini};

- il predetto decreto è ingiusto ed illegittimo e avverso lo stesso si propone formale opposizione per i seguenti

MOTIVI

Figura 6: Versione marcata del facsimile di un atto giudiziario.

[18] Ai fini del progetto, non è stato necessario creare sottocategorie di toponimi (cioè distinguere i nomi delle città dai nomi dei paesi).

Naturalmente, per mantenere la coerenza nelle sostituzioni all'interno del testo, tutte le occorrenze dello stesso termine precedute dallo stesso marcatore vengono sostituite con lo stesso termine fittizio in tutto il documento. Così, nell'esempio sopra, il programma ha sostituito il cognome "Bellini" con "Bandino" in tutte le sue occorrenze all'interno del procedimento (v. *infra* Figura 7).[19] Questa mappatura tra i nomi originali e quelli pseudonimizzati garantisce coerenza non solo all'interno di un unico documento, ma anche nel caso di più documenti relativi allo stesso procedimento.

Quanto a stringhe numeriche e alfanumeriche (come numero di targa, numeri di telefono, ecc.), esse vengono di norma sostituite con stringhe numeriche e alfanumeriche casuali della stessa lunghezza; a meno che non si desideri preservare il loro ordine di grandezza, rimuovendo comunque i valori originali (cfr. *supra* par. 4). Allo stesso modo, poiché la coerenza dei riferimenti temporali all'interno del testo risulta fondamentale per utilizzare gli atti anche per studi giuridici (che comportano la ricostruzione dei fatti e degli eventi del processo), il comportamento del programma per le date può essere configurato per preservare le distanze cronologiche relative.

Rispetto infine alla sicurezza di questo trattamento, va aggiunto che la mappatura tra nomi e pseudonimi a esso funzionale non può essere in alcun modo esportata: si tratta sostanzialmente di una legenda temporanea, con una vita transitoria che si esaurisce con la fine del processo, e non sarà più recuperabile in seguito.[20] Infatti, data la finalità del progetto e la natura dei testi, una procedura di de-pseudonimizzazione[21] non solo non è necessaria, ma nemmeno auspicabile, proprio a garanzia dell'irrecuperabilità dei dati personali. Inoltre, il ricercatore può eseguire l'intero processo senza lasciare la propria macchina, il che costituisce un altro requisito essenziale per evitare che qualsiasi documento non trattato sfugga al suo controllo (cfr. *infra* par. 8). A tal fine, il programma è implementato come uno strumento a linea di comando multipiattaforma, in grado di elaborare in blocco un qualsiasi numero di documenti. Nel contesto di tale procedura, gli unici documenti autorizzati ad uscire da questa "zona di sicurezza" sono quindi i testi pseudonimizzati.

6. Trattamento delle abbreviazioni

Si è visto come l'*output* del processo derivi dalla gestione di ciascuna categoria di dati in base alla sua natura e alle finalità del loro trattamento. Questo ha una stretta corrispondenza lato *software*: proprio come si forniscono diversi marcatori per diverse categorie di dati, diversi moduli *software*, ciascuno specializzato nella gestione di alcune categorie, sono composti insieme in un'architettura flessibile, piuttosto che utilizzare un algoritmo monolitico. In questo modo, il semplice scambio di un modulo con un altro può portare a un risultato completamente diverso.

Infine, nell'ambito di questa flessibilità, si può citare un altro adattamento alle peculiarità dei documenti qui trattati: ancora una volta un'esigenza linguistica è responsabile di una seconda fase di annotazione, qui automatica anziché manuale.

[19] Per ulteriori esempi cfr. [16], 33–34.

[20] Cfr. [24].

[21] Cfr. [28], [13].

Infatti, una caratteristica importante di questi documenti piuttosto tecnici è la loro ricchezza di abbreviazioni, che non solo rendono il testo meno leggibile per utenti non specializzati, ma rischiano anche di compromettere ulteriori elaborazioni linguistiche come l'individuazione delle frasi.

Le frasi, come qualsiasi altra struttura testuale, sono profondamente integrate nel motore di ricerca qui adottato, dove rappresentano oggetti ricercabili, proprio come semplici "parole", con una serie di operatori contestuali e posizionali pensati proprio allo scopo di combinare diverse strutture nella stessa ricerca. Tuttavia, l'algoritmo di individuazione delle frasi si basa principalmente (sebbene non esclusivamente, nel caso ad esempio di una fonte XML[22]) sulla punteggiatura. Quindi, in questi testi, molto ricchi di abbreviazioni che spesso includono punti, il semplice approccio basato sulla punteggiatura non sarebbe sufficiente: poiché la maggior parte delle abbreviazioni termina con un punto, esso non potrebbe essere considerato come un marcatore di fine frase. Per evitare questo inconveniente senza ulteriori aggravi per i ricercatori addetti alla marcatura, il programma utilizza un elenco di abbreviazioni da individuare nei testi. Ciò consente di marcarle automaticamente, in modo che in seguito gli eventuali punti in esse contenuti non abbiano alcun effetto sull'algoritmo di rilevamento delle frasi. Un ulteriore vantaggio di questo approccio è poi rendere i testi più leggibili per utenti non specializzati, che potrebbero non essere in grado di sciogliere le loro numerose abbreviazioni. Proprio per questo l'annotazione viene arricchita con un elenco dei suoi possibili scioglimenti, che possono essere successivamente mostrate all'utente finale, ad esempio come un *popup* visualizzato passando il *mouse* sulla sigla.[23]

7. Pseudonimizzazione e conversione

Nel contesto di queste procedure automatiche, l'unico compito del ricercatore consiste dunque nell'annotare il testo originale direttamente in un'applicazione di videoscrittura (di solito MS Word). Da questo punto in poi, tutto il processo è automatico, anche se altamente personalizzabile in base ai dati e alle finalità. Un esempio di rappresentazione del risultato del processo di pseudonimizzazione è mostrato nella Figura 7, nella quale si offre un estratto del documento HTML generato dalla trasformazione dell'*output* TEI (*Text Encoding Initiative*) del programma:

[22] In XML (*eXtensible Markup Language*) può ben accadere che l'informazione desunta dal semplice testo sia integrata da quella derivante dalla marcatura. Ad esempio, si pensi al caso del titolo di una sezione, tipicamente racchiuso in un elemento *head*: in tal caso, di norma il titolo non è terminato da un punto, ma rappresenta comunque un testo distinto rispetto alla prima frase del testo che lo segue.

[23] Naturalmente, decidere quale scioglimento dovrebbe essere scelto tra i possibili candidati richiederebbe un'elaborazione basata su procedimenti di *Natural Language Processing* (NLP) che aggiungerebbe un sovraccarico peraltro poco utile in questo contesto, dove lo scopo principale nel marcare le abbreviazioni è anzitutto evitare effetti negativi sull'individuazione delle frasi.

GIUDICE DI PACE DI TARCENTO

ATTO DI CITAZIONE IN OPPOSIZIONE

A DECRETO INGIUNTIVO EX ART. 645 C.P.C.

E CONTESTUALE ISTANZA DI SOSPENSIONE DELLA PROVVISORIA ESECUTIVITA'

La sottoscritta Avv. Gradita Bertanzetti del foro di Tarcento, C.F.: TUWZWL08K38R553U, che rappresenta e difende ad ogni effetto di legge, in virtù di delega posta in calce al presente atto la SIG.RA **BANDINO GOFFREDA** nata a Verrès il 24/11/1965 e residente in Tarcento, Via Khepri Santori, 84, C.F.: IEE QGR 53E69 Z778I, ed elettivamente domiciliata presso lo studio del predetto avvocato in Tarcento, Via Giosia Iovine, 51, con indicazione del n. fax al 6259/8989756, PEC: fk5172@tiscali.it,

PREMESSO CHE

-in data 19/11/2008 il Giudice di Pace di Tarcento emetteva a favore di **Brillante S.p.a** *decreto ingiuntivo provvisoriamente esecutivo* n. 3079/4358 per la somma di € 4.287,51 oltre interessi di mora e spese della procedura;

-tale decreto, munito di formula esecutiva in data 3/12/2008, veniva notificato in data 12/12/2008 alla Sig.ra Goffreda Bandino;

- il predetto decreto è ingiusto ed illegittimo e avverso lo stesso si propone formale opposizione per i seguenti

MOTIVI

Figura 7: Documento pseudonimizzato (reso in HTML), con diversi colori a rappresentare diverse categorie di annotazione.

Anche questo breve esempio dovrebbe essere sufficiente a evidenziare come un documento pseudonimizzato formalmente non risulti affatto diverso da uno non trattato, tranne che per una certa quantità di nomi personali piuttosto inconsueti negli pseudonimi. Questo è l'effetto degli elenchi di nomi utilizzati dal programma, che coprono un ampio arco cronologico e geografico per garantire un elevato numero di scelte; ed è stato volutamente preservato, proprio per meglio assicurare sull'effettiva pseudonimizzazione del documento, che a prima vista non appare affatto evidente. Proprio tale ambiguità rappresenta in effetti la più chiara indicazione della bontà di questo trattamento.

Inoltre, un ulteriore vantaggio offerto da questo approccio, che aggiunge informazioni solo per poi rimuoverle, è anche il fatto che il sistema di pseudonimizzazione diventa in grado di rimodellare il documento di partenza, con la sua struttura puramente tipografica, in un documento strutturato semanticamente. Infatti, avvalendosi delle diverse fonti di metadati incluse nel proprio *input*, il sistema non solo pseudonimizza il documento, ma lo converte anche

in un documento TEI, rimappando in elementi XML le categorie di dati e le caratteristiche tipografiche dei documenti del programma di videoscrittura.

Pertanto, il processo completo prevede una fase di decodifica del formato originale, una fase di pseudonimizzazione secondo un insieme variabile di regole, e infine la generazione di un documento TEI, accompagnato da eventuali rese tipografiche in HTML, per fornire ai ricercatori addetti alla marcatura un immediato riscontro sul loro lavoro. Tutto ciò è incluso nel flusso che porta dal documento originale a un testo pseudonimizzato, affiancato da un distinto contenitore di metadati, che seguirà poi il percorso verso un archivio centrale e l'indicizzazione del testo.

8. Flusso di dati

Il flusso di dati che emerge dalla discussione precedente può dunque essere riassunto nella Figura 8.

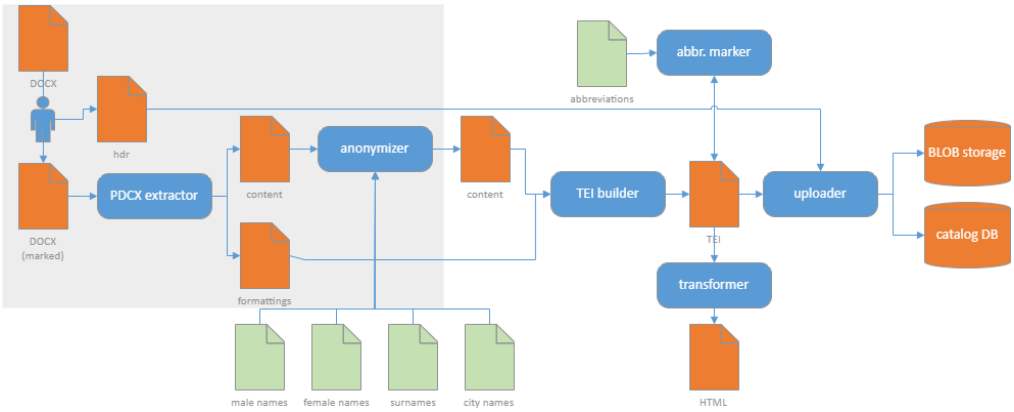


Figura 8: Prima parte del flusso generale dei dati: il riquadro grigio delimita l'area protetta, dalla quale nessun dato personale può uscire. Da sinistra a destra: i documenti di videoscrittura vengono filtrati per estrarre contenuti e stili rilevanti, pseudonimizzati avvalendosi di varie liste onomastiche, convertiti in TEI, marcati automaticamente per le loro abbreviazioni, trasformati in HTML per agevolare il controllo da parte dei ricercatori addetti alla marcatura e caricati in un deposito remoto centralizzato.

Il flusso prende origine da un documento di videoscrittura, tipicamente in formato DOCX,[24] quale che sia la sua origine: potrebbe provenire da MS Word o da qualsiasi altro applicativo, o anche essere il risultato di OCR (*Optical Character Recognition*) per documenti cartacei più datati. A questo punto, i ricercatori intervengono per introdurre la leggera marcatura sopra illustrata,

[24] DOCX è uno standard ISO/IEC 29500 basato su XML compresso con ZIP. Oltre a essere in grado di codificare un'ampia serie di funzionalità, essendo quindi un buon obiettivo per le conversioni da diversi formati di *word processor*, la sua natura standard garantisce che si possa elaborarlo indipendentemente dall'applicazione che lo ha prodotto.

producendo così un'altra versione del documento, uguale all'originale tranne che per le sue annotazioni aggiuntive.

Allo stesso tempo, il ricercatore addetto alla marcatura produce un piccolo file che include alcuni metadati relativi al documento elaborato. Per facilitarne la modifica, si tratta di un semplice foglio di calcolo, con una colonna per il nome e un'altra per il valore di ogni metadato. Successivamente, questo foglio verrà automaticamente convertito in un formato più pratico (JSON), e utilizzato dai sistemi di caricamento automatico di documenti nell'archivio centrale.

A questo punto, un componente incorporato nello pseudonimizzatore estrae il testo e la sua formattazione essenziale dal formato DOCX in un dialetto XML appositamente progettato. Questo codifica sia il semplice testo che un sottoinsieme di indicazioni di formattazione estratte dal documento originale.

In effetti, le fonti dei metadati di un documento sono molteplici; anzitutto, la marcatura manuale applicata per l'oscuramento delle informazioni personali e per l'annotazione di altri aspetti utili all'analisi, cui si aggiunge quella automatica delle abbreviazioni; in secondo luogo, il formato di elaborazione testi (DOCX), in cui vengono unificati tutti i documenti, e da cui viene estratto solo un sottoinsieme minimo di informazioni tipografiche; e infine l'eventuale utilizzo di sistemi di POS *tagging*, che consentono di ottenere – pur con inevitabile approssimazione – ulteriori metadati relativi alla lemmatizzazione e alla classificazione morfologica di ciascuna parola. Tutti questi metadati, raccolti da diverse fonti, devono poi trovare la loro strada nell'indice che alimenterà il motore di ricerca.

In questa fase, il testo in chiaro viene elaborato dallo pseudonimizzatore, che si basa su una serie di elenchi comprendenti migliaia di voci per sostituire nomi personali (distinti secondo il loro genere), cognomi e toponimi. L'*output* di questo processo è un testo depurato di ogni dato personale, suscettibile di oltrepassare i confini del perimetro di sicurezza costruito attorno al documento originale, che invece non lascia mai la macchina del ricercatore che lo ha annotato.

A questo punto, i dati sulla formattazione originale vengono recuperati insieme al testo trattato per la pseudonimizzazione, e uniti in un documento TEI prodotto da un altro componente *software*, mappando i vari metadati ricevuti in una serie di marcatori XML.

Una volta che i marcatori si trovano al loro posto, il testo viene elaborato dal rilevatore di abbreviazioni, che si basa su un elenco esterno di abbreviazioni con le loro espansioni. Il risultato è un'altra versione del documento TEI, dotato di marcatori aggiuntivi.

Il documento TEI così prodotto può quindi essere trasformato tramite XSLT (*eXstensible Stylesheet Language Transformations*) in HTML, che consente ai ricercatori addetti alla marcatura di avere un controllo finale del documento in una forma più leggibile e meno dispersiva, opportunamente formattata per un'esperienza di lettura ottimale. Un *output* simile verrà poi utilizzato per presentare il testo agli utenti finali nel contesto dell'applicazione web che incorpora il motore di ricerca.

Infine, il ricercatore utilizza un altro strumento sviluppato per caricare testo e metadati di ogni documento elaborato in un archivio centralizzato, accessibile via web, protetto con un sistema

di autorizzazione standard.[25] Questo archivio rappresenta la sede finale del *corpus*, inclusi i documenti nel loro formato TEI pseudonimizzato, e i loro metadati, archiviati separatamente. Il documento TEI non contiene infatti metadati nella sua intestazione, per proteggerlo ulteriormente da ogni possibile identificazione di persone reali.

9. Requisiti del motore di ricerca

Al pari del pretrattamento dei testi finora illustrato, anche la loro indicizzazione ai fini di ricerca deve adattarsi ai peculiari requisiti del progetto. A tal scopo, nell'ultima parte del flusso di lavoro che porta dai documenti Word ai loro *output* TEI pseudonimizzati è stato introdotto un nuovo motore di ricerca *open source* (*Pythia*).[26] Dato che descrivere il motore richiederebbe un articolo a sé, qui ci si limiterà a una panoramica delle sue caratteristiche più utili per questo progetto.

Anzitutto, il progetto richiede un approccio basato sulle concordanze[27], combinato con un indice riutilizzabile sotto forma di database, che includa i documenti stessi e i loro metadati. Tuttavia, molti dei motori di ricerca di testo più utilizzati si concentrano principalmente sull'individuazione di un documento in un enorme *corpus* tramite il calcolo di un punteggio di pertinenza, piuttosto che sul dettaglio delle occorrenze di ciascuna espressione ricercata nel suo contesto. Inoltre, nella maggior parte dei casi l'indice ha un formato proprietario, sicché non risulta utilizzabile al di fuori dello specifico sistema che lo ha creato.

In secondo luogo, è necessario integrare nella ricerca testuale un insieme virtualmente illimitato di metadati, che potrebbero anche risultare quantitativamente maggiori dei dati testuali stessi, quale che sia la loro fonte, tipicamente composita: metadati dei documenti, attributi tipografici (es. corsivo, allineamento di paragrafo, ecc.) estratti dal loro formato originale, informazioni aggiuntive fornite da sistemi di terza parte come POS *tagger*, ecc.

Peraltro, tutti questi metadati si riferiscono a diversi livelli di analisi, hanno modelli diversi, e si estendono su diverse aree del testo. Ad esempio, un testo poetico non strofico è strutturato in frasi per la sua sintassi, e in versi per il suo metro: entrambe sono strutture linguistiche, ma poggiano su diversi livelli di analisi, e come tali spesso risultano disallineate e variamente sovrapposte. Se poi si volesse considerare a livello di impaginazione la loro colometria, si introdurrebbe una terza struttura, non necessariamente corrispondente a nessuna delle due.

Pertanto, le porzioni di testo corrispondenti a ciascuna struttura spesso non sono solo nidificate, ma anche sovrapposte, andando così oltre le semplici capacità di codifica di un documento basato su XML. Tuttavia, sarebbe utile poter combinare nella stessa ricerca dati così eterogenei: ciò aggiungerebbe anche la possibilità di utilizzare strutture che abbracciano più parole, come

[25] È possibile accedere al codice sorgente completo di questo sistema e alla relativa documentazione tecnica su https://github.com/vedph/simple-blob. Per quanto riguarda il motore di ricerca, entrambi i sistemi vengono generalmente distribuiti come servizi containerizzati con *Docker*. È dunque possibile utilizzarli su una macchina locale o su un server, semplicemente avviando uno *script*.

[26] V. https://github.com/vedph/pythia per il codice sorgente e la documentazione.

[27] Si tratta di un aspetto da tempo evidenziato dalla letteratura scientifica, ma tutt'altro che ubiquitario negli strumenti più comunemente usati a scopo di ricerca testuale, spesso progettati per scopi e volumi di dati affatto diversi: cfr. ad es. [5] e [31].

frasi o versi, per consentire ricerche contestuali in un contesto più significativo, definito ad esempio dalla sintassi o dal ritmo, piuttosto che da un meccanico conteggio delle parole (un certo numero di parole prima o dopo quello cercato, anche quando appartengono a frasi o versi diversi).

Infatti, il processo di indicizzazione in questo e in altri progetti deve fungere anche da una sorta di *hub*, raccogliendo e adattando in un modello uniforme dati provenienti da fonti completamente diverse e indipendenti: ad esempio, file Excel con metadati del documento; file Word che corrispondono ai testi e da cui si possono ricavare alcune informazioni di formattazione; marcatori TEI che derivano per trasformazione dall'annotazione manuale a scopo di pseudonimizzazione, nonché dall'estrazione di metadati dai documenti; marcatori POS derivanti dall'elaborazione in strumenti di terza parte[28]; ecc. Di fatto, grazie anche all'introduzione di marcature POS per questo *corpus* i metadati finiscono per rappresentare l'85% dei dati relativi alle parole nell'indice.

Occorre dunque un modo per incorporare tutti questi metadati in un indice capace di presentare una superficie uniformemente ricercabile, il che implica la necessità di un più elevato livello di astrazione. È proprio questa astrazione che consente di trattare allo stesso modo non solo le parole e i loro metadati, ma anche strutture testuali più estese (come frasi o – anche se non è il caso di questo specifico progetto – versi e strofe),[29] pure dotate dei loro metadati.

10. Architettura dei motori di ricerca

Per soddisfare tali requisiti, l'architettura del motore di ricerca è stata progettata proprio su un più alto livello di astrazione: l'idea centrale è appunto un modello del testo[30] che prevede una sorta di sua smaterializzazione ([19]).

[28] In particolare, per questo *corpus* viene utilizzata una serie di componenti NLP basati su *Universal Dependencies* tramite *UDPipe* (https://lindat.mff.cuni.cz/services/udpipe/), in modo da iniettare in ogni *token* i risultati di un POS *tagger* per la lingua italiana. Anche questi componenti sono modulari al pari degli altri, e vengono quindi inseriti nella *pipeline* del sistema di indicizzazione tramite il profilo di configurazione del *corpus*.

[29] Tali esempi si comprendono meglio nel contesto che ha dato origine al progetto *Pythia*, appunto costituito proprio da un sistema di analisi linguistica e metrica (Chiron), capace di fornire ai testi analizzati letteralmente milioni di metadati altamente specializzati, da interi versi (o frasi, nel caso di prosa ritmica) fino a tratti subfonematici ([19], [20]). Più in generale comunque, una simile architettura può rappresentare uno strumento al servizio di modellazioni più complesse e strutturate connesse al testo indicizzato, che possono articolarsi anche su livelli di analisi multipli: si pensi ad esempio a modelli come CRMtex, ispirati a ontologie di ampia diffusione come CIDOC-CRM, che pongono il testo alla base di una serie di piani di analisi di varia granularità all'interno del complesso fenomeno della scrittura (cfr. [14]).

[30] Benché intuitivamente risulti meno evidente, qualsiasi sistema che manipoli un testo non può che fare delle assunzioni più o meno implicite rispetto al suo modello. Naturalmente, questo rappresenta sempre un'approssimazione, utile rispetto all'oggetto della propria ricerca e gli scopi che ci si prefigge. Nondimeno, tenendo ben presente, come ricorda Buzzetti [1], che altro è una

Tradizionalmente, un tipico sistema di ricerca testuale si concentra su sequenze di caratteri (*token*), variamente estratte e filtrate da un testo, magari accompagnate da metadati aggiuntivi. La ricerca si concentra sul confrontare tali sequenze con quella dell'espressione desiderata, limitando semmai ulteriormente i risultati con l'aiuto dei metadati (Figura 9).

Inoltre, in alcuni casi tra questi metadati c'è anche la posizione nel documento, che consente ricerche più granulari, come quelle richieste nelle concordanze, semplicemente calcolandone la differenza rispetto ad altri *token*; benché non di rado in alcune implementazioni questa funzionalità assomigli più a un'aggiunta secondaria che a una caratteristica progettata assieme al nucleo del sistema.

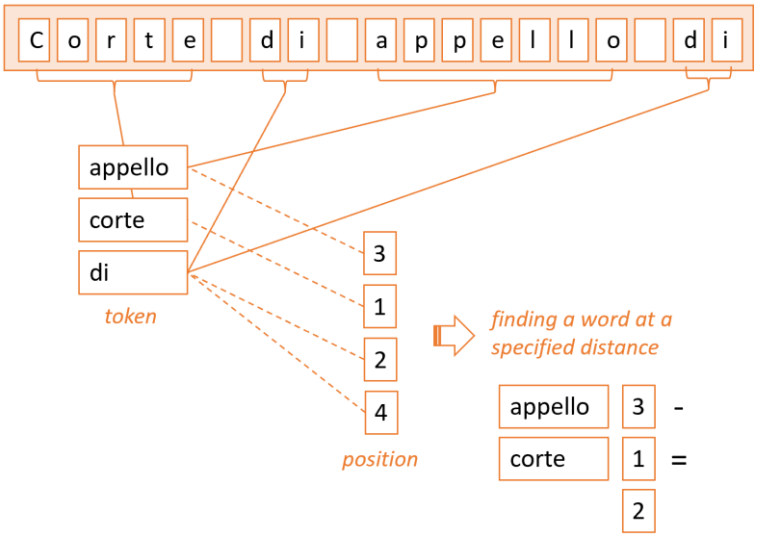


Figura 9: Modello di un motore di ricerca testuale focalizzato sul testo come sequenza di caratteri: il testo viene suddiviso in sottosequenze di caratteri (*token*), soggette a filtro per eliminare caratteristiche non utili alla ricerca; ogni *token* riceve un numero rappresentante la sua posizione nel documento, utilizzata per trovare co-occorrenze di parole a una data distanza.

In *Pythia* invece, sebbene ovviamente i meccanismi di base siano gli stessi, l'attenzione si sposta da questa sequenza di caratteri a un insieme aperto di oggetti, ciascuno dotato di un qualsiasi numero di metadati. In questo senso, *corpora* (cioè gruppo di documenti), documenti, frasi, versi, parole e altre strutture simili sono tutti oggetti. Il modello di tali oggetti peraltro è molto semplice: un oggetto qui è solo un contenitore di proprietà scalari. Alcuni di questi oggetti sono esterni al documento, come un *corpus* o il documento stesso; altri invece sono interni. Questi ultimi fra le loro proprietà includono anche la posizione nel documento, che consente ricerche granulari e tradizionali visualizzazioni KWIC.

mappa e altro il territorio da essa rappresentato, la modellazione qui è un aspetto essenziale e dunque molto più esplicito che in altre tecnologie di ricerca testuale.

In questa architettura, la ricerca consiste dunque nel trovare tutti gli oggetti i cui metadati corrispondono ai criteri specificati, eventualmente all'interno di uno specifico sottoinsieme dell'indice. I metadati sono quindi i punti di connessione attraverso i quali gli utenti accedono agli oggetti.

Questa modellazione ha l'effetto di modificare il processo di ricerca come le sue caratteristiche: non si tratta più di abbinare sequenze di caratteri (Figura 10), quanto piuttosto di trovare uno o più dei punti terminali rappresentati dalle proprietà di ciascun oggetto, memorizzati in una *black box* la cui struttura interna non è rilevante per la ricerca, e che rappresentano non solo parole (*token*), ma anche altre entità, testuali o meno (Figura 11).

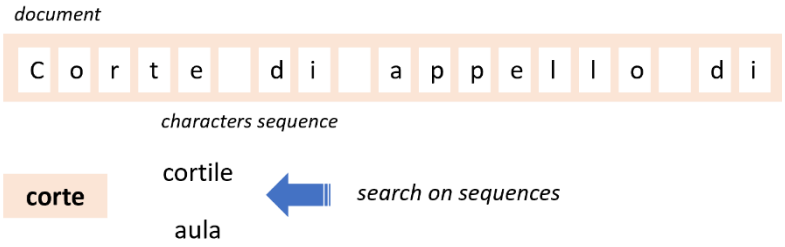


Figura 10: Ricerca tradizionale.

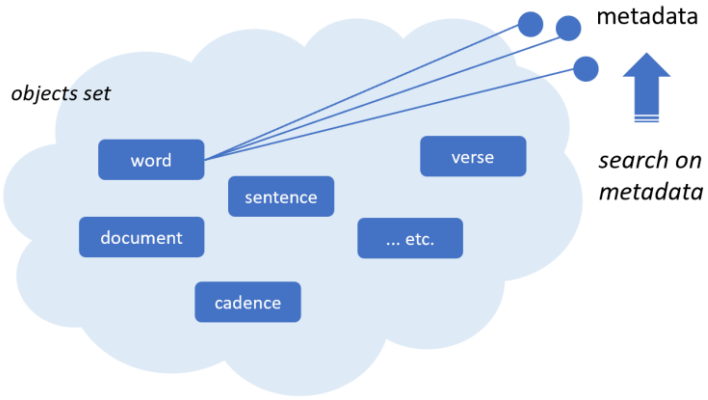


Figura 11: Ricerca in Pythia.

Inoltre, va notato che in una simile architettura il testo non è più l'unica fonte di metadati; a questo livello di astrazione, in cui non si tratta più direttamente di sequenze di caratteri, qualsiasi fonte di metadati relativa al nostro testo può essere inserita fra le proprietà degli oggetti, e quindi ricercata al pari di qualsiasi altra. In effetti, l'elenco dei metadati non è chiuso: un metadato è solo un valore denominato di qualsiasi tipo.

Pertanto, dal punto di vista della ricerca un documento di testo è innalzato a un superiore livello di astrazione: piuttosto che una sequenza di caratteri, è un insieme di oggetti, molti dei quali hanno una posizione specifica al suo interno. Tuttavia, questa posizione è solo uno qualsiasi dell'insieme virtualmente illimitato di metadati, e non rappresenta più una sorta di aggiunta a posteriori, ma una caratteristica radicata nell'architettura di base.

La Figura 12 illustra schematicamente questo principio: un testo di partenza viene rimodellato come un insieme di oggetti, tutti definiti allo stesso modo (le caselle grigie); la differenza è solo nei loro metadati. Dato che tra di essi si trova anche la loro posizione, ove applicabile, il risultato è un insieme di oggetti non ordinato, il che costituisce un modello adatto alla maggior parte dei tipi di database standard.

A sua volta, ciascuna di queste caselle ha un insieme aperto di metadati, modellato come coppia nome=valore. Ad esempio, nel caso del *token* "Fillide" (antroponimo femminile), i metadati mostrati in figura si riferiscono ad aspetti assai vari, come numero di caratteri e sillabe, classificazione morfologica, valore del testo filtrato ("fillide"), e persino grassetto, che si trovava applicato a quella parola nel documento Word originale.

A un livello linguistico più alto, poi, anche le frasi sono caselle; nella figura ce ne sono due, corrispondenti alle due frasi del testo di esempio. Queste caselle comunque non sono diverse da tutte le altre: includono pertanto i loro metadati e la loro posizione, stavolta rappresentata come un segmento delimitato da una posizione iniziale e finale, piuttosto che come un singolo punto.

In questo scenario, una ricerca testuale corrisponde in definitiva all'esame di qualsiasi sottoinsieme dei metadati esposti da ciascuno di questi oggetti, che non sono necessariamente parole: possono essere altre strutture testuali (es. frasi o versi), o anche documenti o gruppi di documenti (*corpora*).

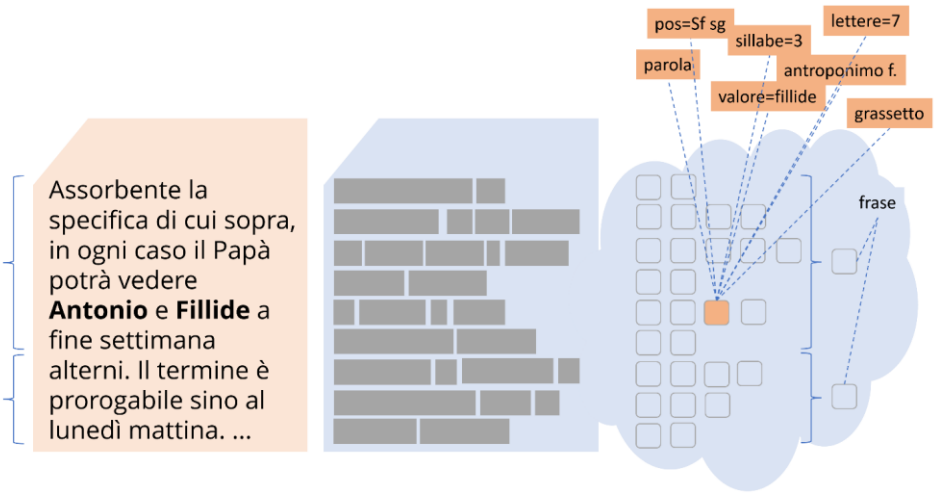


Figura 12: Rimodellamento di un testo in un insieme di oggetti con metadati, che forniscono il collegamento alle funzioni di ricerca.

Di conseguenza, anche un semplice sguardo all'interfaccia della ricerca assistita, che consente di costruire una interrogazione in modo visuale invece di avvalersi della sintassi del motore, illustra la diversità di questo modello (Figura 13). Il valore testuale della parola come sequenza di caratteri qui è solo uno dei tanti metadati associati all'oggetto che la rappresenta, sicché ogni elemento dell'interrogazione è una coppia nome/valore, dove il nome indica un qualsiasi metadato: il valore testuale, ma anche lingua, numero di sillabe, grassetto, classe morfologica, ecc. Una serie di operatori consente poi di unire questo nome a un valore.

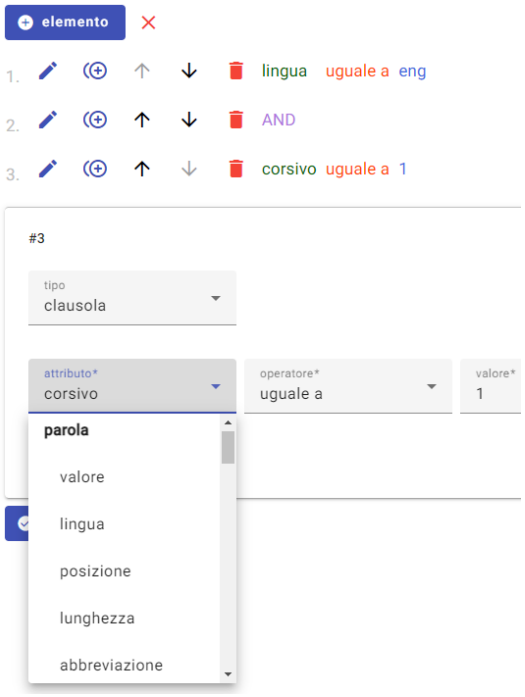


Figura 13: Interfaccia di costruzione visuale di una interrogazione.

Fra gli operatori non si trova solo il tradizionale insieme di operatori logici e di raggruppamento; ma anche un insieme di operatori *posizionali* appositamente progettati, che consentono agli utenti di trovare oggetti all'interno di altri oggetti, o parzialmente sovrapposti ad altri oggetti, o corrispondenti all'inizio o alla fine di altri oggetti, ecc. Senza entrare in dettagli, ci si può limitare a elencare gli operatori disponibili con la loro rappresentazione sintattica (in parte ispirata ai selettori CSS):[31]

[31] Va comunque notato che l'intero linguaggio di *query* fornito da *Pythia* è solo uno di quelli possibili. Questo è definito con una semplice grammatica basata su ANTLR; dato che la tecnologia di archiviazione sottostante è un database relazionale, il suo compito consiste semplicemente nel

- = uguale a
- <> non uguale a
- *= compreso
- ^= a partire da
- $= termina con
- ?= espressione con caratteri jolly
- ~= espressione regolare
- %= *fuzzy matching* (con l'aggiunta di un valore di soglia compreso tra 0 e 1)
- operatori numerici (trasformando i valori dei metadati in numeri, consentendo così confronti numerici):[32] ==, !=, >, >=, <, <=.
- operatori di collocazione: NEAR, NOT NEAR, BEFORE, NOT BEFORE, AFTER, NOT AFTER, INSIDE, NOT INSIDE, OVERLAPS, LALIGN (*left aligned*), RALIGN (*right aligned*).

Proprio in virtù dell'approccio orientato ad oggetti, che non necessariamente corrispondono a parole, gli operatori di collocazione possono essere utilizzati non solo per individuare la posizione relativa di una parola rispetto all'altra, ma anche rispetto ad altre strutture. Ad esempio, se si volesse ricercare una parola all'inizio o alla fine di una frase (o di un verso, o di qualsiasi altra struttura), si potrebbe semplicemente cercare quella parola da un lato, e una qualsiasi struttura frase dall'altro, unendole con un operatore LALIGN o RALIGN con una distanza massima uguale a zero. Questa non è altro che la formalizzazione di un rapporto posizionale fra due entità, una parola e una frase, che vogliamo trovare allineate in modo che la parola sia in testa o in coda alla frase.

Peraltro, la visualizzazione grafica della *query* esemplificata sopra è solo uno degli effetti della stessa modellazione per oggetti. Un'altra visualizzazione che ben si presta a rappresentarli è ad esempio costituita dalla possibilità di esplorare la distribuzione di ogni parola in gruppi di volta in volta definiti dai metadati scelti. Ad esempio (Figura 14), attingendo dall'elenco dei metadati si può visualizzare la distribuzione di una parola come *accertamento* per materia, organo giudicante, data di nascita dell'avvocato (criterio utile per determinare differenze linguistiche riconducibili

tradurre il linguaggio di interrogazione in istruzioni SQL. Quindi, nulla impedisce di sostituire questa lingua con un'altra, o anche di bypassare il linguaggio di interrogazione stesso, e utilizzare direttamente SQL, il che può risultare utile quando si integri l'indice in *software* di terze parti. Per ulteriori informazioni sulla sintassi standard e sugli esempi di interrogazione è possibile fare riferimento alla documentazione *online* nel *repository* all'indirizzo https://github.com/vedph/pythia.

[32] Sebbene questa sia un'implementazione relativamente banale per un RDBMS sottostante, va notato che in molti sistemi di ricerca a tutto testo questa non è sempre una caratteristica data per scontata. Ad esempio, tradizionalmente nei sistemi basati su *Lucene* i numeri devono essere formattati come stringhe a lunghezza fissa per consentire confronti numerici significativi.

all'età e alla formazione dello scrivente), ecc. L'interfaccia consente di aggiungere qualsiasi altro metadato ai criteri, arricchendo così l'insieme dei grafici di un nuovo elemento.

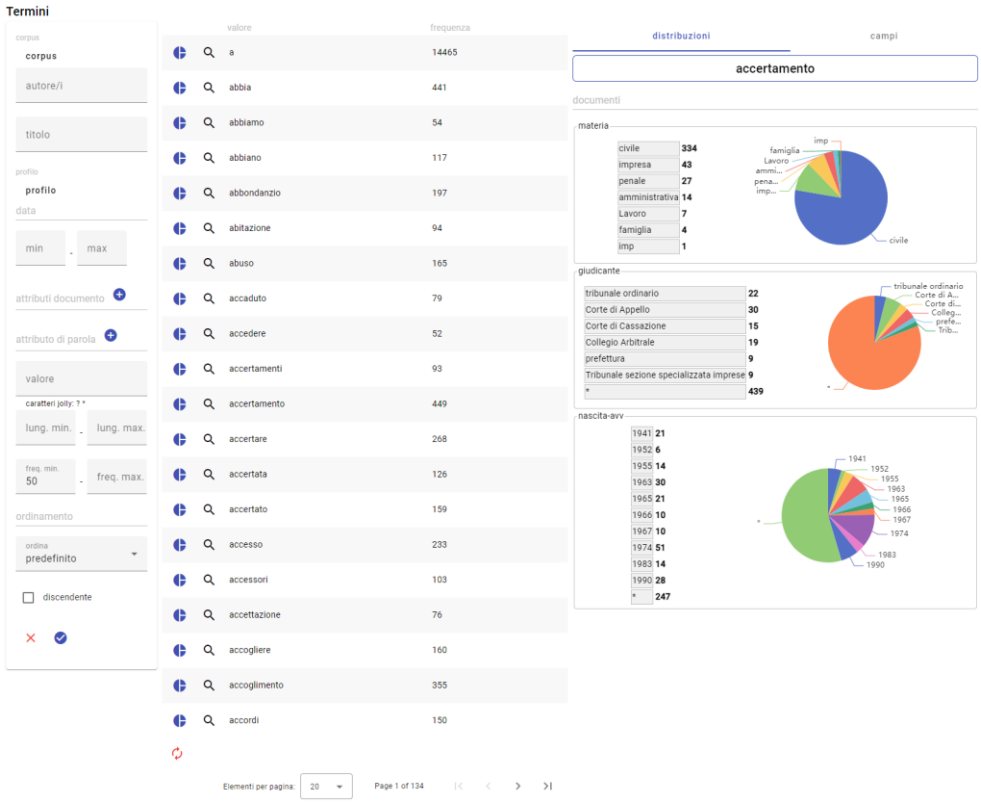


Figura 14: Elenco delle parole indicizzate con frequenza totale e distribuzione delle frequenze secondo gruppi definiti interattivamente selezionando vari metadati fra quelli assegnati a documento e parole.

11. *Pipeline*

Questa sorta di smaterializzazione del testo consente dunque di ricercare un insieme aperto ed estremamente vario di metadati, indipendentemente dall'oggetto a cui appartengono o dalle loro fonti, spesso molteplici. In questo ambito, il sistema di indicizzazione deve pertanto consentire la stessa apertura della ricerca.

L'architettura che soddisfa tale requisito concatena appunto diversi tipi di moduli *software* riutilizzabili in una *pipeline*, che gestisce l'intero flusso di elaborazione dei dati, quali che siano la loro origine (*file system*, *database*, risorse web, ecc.) e il loro formato digitale (testo semplice, XML come TEI, database, ecc.) sino al *parser* che modella gli indici, inclusi filtri, tokenizzatori, estrattori

e calcolatori di metadati, rilevatori di strutture di testo, ecc. Tutti questi componenti sono definiti con le loro eventuali opzioni in un documento JSON esterno, che configura la *pipeline*, i cui moduli possono essere anche prodotti da terze parti e integrati come *plugin*.

In generale, questi componenti rientrano in una serie di categorie, elencate qui nel loro tipico ordine di utilizzo:

1. *source collector*: componenti che producono un elenco di documenti a partire da una fonte. Questo consente di astrarre la fonte dei documenti dalla loro indicizzazione: ad esempio, un *collector* basato su *file system* enumera i file dalle loro cartelle, mentre uno basato su *cloud* li enumera a partire da un'origine dati remota su internet. È quindi possibile semplicemente sostituire un componente con l'altro per far sì che il resto della *pipeline* operi ora a partire da una macchina locale e ora a partire da una remota, senza alcuna differenza.

2. *literal filter*: filtri applicati ai valori letterali dei termini utilizzati nelle *query* di ricerca, sì da armonizzarli con quelli filtrati durante l'indicizzazione.

3. *text filter*: filtri applicati al testo nel suo insieme. Questi possono fornire sostituzioni globali, pretrattamenti TEI, ecc.

4. *attribute parser*: componenti che estraggono attributi (metadati) dai documenti. Questi metadati verranno quindi inseriti nell'oggetto rappresentante il documento. Ad esempio, un *parser* XML utilizza una combinazione di *XPath* ed espressioni regolari per estrarre metadati da qualsiasi documento XML; un *parser* Excel utilizza una serie di parametri per estrarre i metadati da un foglio di calcolo; ecc.

5. *document sort key builder*: componenti che creano una chiave di ordinamento per i documenti, in modo che vengano ordinati in un modo specifico. Questo consente di definire il proprio schema di ordinamento (in genere basato su una combinazione di metadati del documento) quando si presentano i documenti nei risultati di una ricerca, o si visualizza l'elenco dei documenti stessi.

6. *date value calculators*: componenti che calcolano un valore numerico approssimativo a partire dalla data dei documenti, come espressa fra i loro metadati. Il valore della data è in questo senso un metadato privilegiato poiché viene spesso utilizzato in *corpora* di documenti storici.

7. *tokenizer*: componenti che spezzano il testo in *token*.

8. *token filter*. Componenti che non solo filtrano il testo del *token*, ma possono anche supplire metadati: ad esempio, tali filtri possono fornire il conteggio di sillabe o caratteri di ciascun *token*; allo stesso modo, si potrebbero supplire dati da *POS tagger*, analisi metriche, ecc. Tutti questi metadati saranno comunque ricercabili, al pari del valore testuale del *token*, che in un motore tradizionale invece rappresenta il dato privilegiato, quando non persino unico.

9. *structure parser*: componenti che rilevano strutture testuali di qualsiasi estensione (es. frase, verso, ecc.) in un documento. Un esempio è il *parser* di frasi XML, che combina l'approccio basato sulla punteggiatura sopra citato per individuare i limiti delle frasi con

i dati impliciti nell'uso di determinati marcatori XML. Poiché il formato di *input* in Atti Chiari è TEI, qui l'algoritmo di divisione delle frasi generico (basato principalmente sulla punteggiatura) viene combinato con un approccio configurabile, che tiene conto anche della natura di determinati marcatori. Quindi, un marcatore come <head>, usato per i titoli, ai fini dell'individuazione di strutture viene trattato come una frase, anche in assenza di interpunzione finale. In questo modo, il sistema può utilizzare un modulo di suddivisione delle frasi che sfrutta le informazioni provenienti sia dal testo sia dai suoi marcatori XML, ove presenti. Lo stesso modulo può essere riutilizzato ogni volta che sia necessario individuare i confini di frase, sia per documenti di testo che XML, indipendentemente dal loro dialetto.

10. *structure value filters*: filtri applicati ai valori delle strutture testuali.
11. *text retriever*: componenti utilizzati per recuperare il testo completo di ciascun documento dalla sua fonte, che potrebbe essere il database dell'indice stesso, o una risorsa esterna (*file system*, *cloud storage*, ecc.). La possibilità di disporre del testo completo nel suo formato originario consente di utilizzare il sistema non solo come strumento di ricerca, ma anche come ambiente di consultazione e lettura dei testi.
12. *text mapper*: componenti che costruiscono una mappa di testo navigabile e gerarchicamente ordinata a partire dall'analisi di un documento. Una mappa di testo è un'astrazione modellata come un albero, in cui ogni nodo punta a una parte specifica del documento. Tali mappe vengono utilizzate per consultare i documenti, e per selezionare porzioni di testo dai nodi della mappa. Quest'ultima funzione viene utilizzata quando si presenta il risultato di una ricerca nel suo contesto, per garantire che questo risulti significativo e conforme alla struttura del documento, piuttosto che meccanicamente tagliato in base a criteri quantitativi.
13. *text renderer*: componenti che renderizzano un testo per la sua presentazione all'utente finale. Ad esempio, in Atti Chiari viene usato un *renderer* basato su XSLT, che trasforma TEI in HTML.

La flessibilità di questa architettura sta proprio nelle innumerevoli possibilità di ricombinare i suoi componenti, o di introdurne di nuovi. Un esempio relativo alla costruzione del *corpus* Atti Chiari è costituito proprio dall'integrazione di un servizio UDPipe per la lemmatizzazione e la marcatura morfologica dei testi. Dato che il sistema deve mantenere la propria generalità, occorre progettare dei componenti adatti a un contesto che rimanga totalmente agnostico e personalizzabile rispetto al formato digitale del testo trattato (sia esso TEI o qualsiasi altro dialetto XML, *plain text*, ecc.) e ai processi di indicizzazione. Ad esempio, l'algoritmo di tokenizzazione potrebbe ben essere diverso da quello utilizzato da UDPipe.

In questo caso dunque sono stati creati due diversi componenti destinati a lavorare in concerto: un filtro UDP per il testo e un secondo filtro per i *token*. Il primo appartiene ai componenti del punto 3 citato sopra, e rappresenta quindi un filtro applicato al testo del documento nel suo complesso. In tal caso, questo filtro non tocca il testo, ma approfitta della sua posizione

privilegiata quasi all'inizio della *pipeline*[33] per sottoporre il testo completo all'esame di UDPipe prima che esso venga variamente (e imprevedibilmente, dato che i componenti nella *pipeline* possono variare) alterato. Più oltre nella *pipeline* poi, una volta che il testo è stato diviso in *token*, interviene il secondo filtro UDP, il cui compito è semplicemente mappare il *token* generato dalla *pipeline* con quello prodotto da UDPipe e memorizzato dal filtro precedente (laddove i *token* possano essere confrontabili). In tal modo, quale che sia il formato di *input* o il tipo di tokenizzazione, diviene possibile nella maggior parte dei casi riconciliare i prodotti dell'analisi di UDPipe con quelli della specifica *pipeline*, iniettando così nuovi metadati a livello di ciascun *token*. Il risultato è che diviene possibile cercare parole in base alla loro forma lemmatizzata o alla loro classificazione morfologica.

Benché questo esempio riguardi aspetti di dettaglio del processo di indicizzazione, si può osservare che i componenti non si limitano a quest'area, ma comprendono un vero e proprio sottosistema di lettura e consultazione del testo, completamente integrato e progettato assieme al sistema di ricerca (punti da 11 a 13). Infatti, tipicamente *Pythia* viene utilizzato nell'ambito di progetti più ampi, dove i *corpora* raccolti hanno potenzialmente molti usi diversi e rappresentano un valore anche in sé. In questo modo, diviene possibile fornire un ambiente di lavoro in cui gli utenti possono passare dalla ricerca alla lettura senza soluzione di continuità, in una sorta di biblioteca digitale.

In effetti, anche una ricerca banale di una singola parola, come "abbandonato" nell'interfaccia utente mostrata in Figura 15, fornisce un ambiente di lettura completo: la ricerca, che implica trovare tutti gli oggetti il cui valore di testo dei metadati è uguale ad "abbandonato", fornisce come *output* una tradizionale presentazione KWIC. Cliccando sulla parola cercata si apre una porzione del documento tagliata in modo coerente, con la parola evidenziata al suo interno. A sinistra si trova poi la mappa completa del documento, che consente agli utenti di navigare liberamente nel documento spostandosi in qualsiasi altra sua parte, seguendo la sua intrinseca struttura.

[33] In effetti anche questo filtro è preceduto da altri, come quello che neutralizza il testo relativo ai tag XML. In caso contrario, anche questi sarebbero passati come testo al servizio UDPipe.

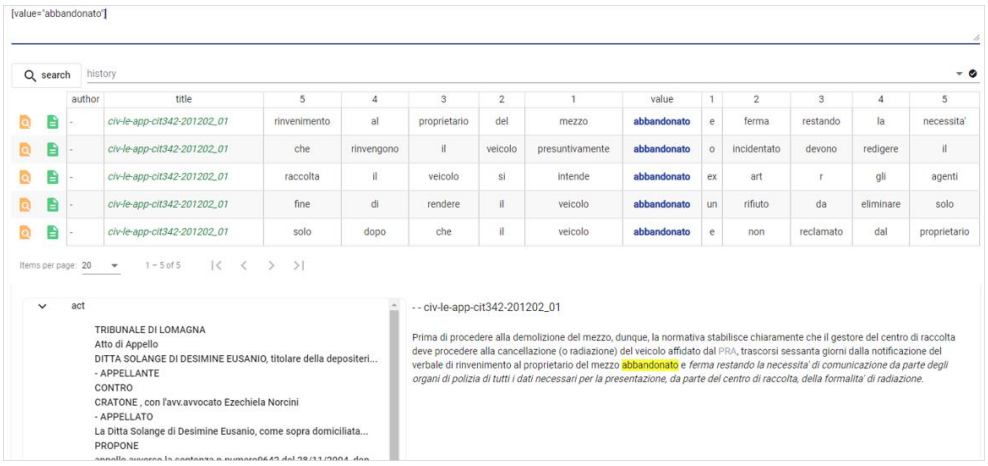


Figura 15: L'interfaccia utente di ricerca di Pythia nella web app client.[34]

Si può inoltre osservare che lo stesso testo del documento viene reso con un insieme di stili e impaginazioni che è frutto della mescolanza fra lo stile del documento originale (per la sua parte estratta dal formato DOCX, ad es. qui il corsivo) e la resa delle marcature aggiunte dai processi di anonimizzazione e analisi (ad es. qui il diverso colore dell'abbreviazione PRA).

12. Architettura aperta

La combinazione del sistema di pseudonimizzazione e del motore di ricerca fornisce dunque un sistema completamente configurabile e modulare, con un livello di astrazione più elevato che consente una serie di caratteristiche peculiari, assai utili nel contesto di questo e di altri progetti. In tal modo si completa il flusso che inizia con una serie di documenti Word e termina con un archivio digitale strutturato, disponibile sul web con funzionalità di ricerca e lettura.

L'intero sistema è peraltro progettato per l'apertura e l'integrazione, a partire dalla sua architettura stratificata e distribuita su più livelli. Alla base si trova un database relazionale, non direttamente accessibile ai livelli superiori; piuttosto, uno strato ulteriore (*data layer*) si frappone come astrazione intermedia tra esso e gli strati superiori. Al di sopra giace un *business layer*, con la logica centrale del sistema, a sua volta sormontato da uno strato che espone le funzioni del sistema ai *client* web tramite API (*Application Programming Interface*) e, infine, da uno strato superficiale che fornisce un'applicazione web per l'utente finale, sfruttando funzioni e dati offerti dallo strato API. Si tratta naturalmente di un'architettura tipica, che consente di sostituire uno qualsiasi di questi livelli con un altro, e di integrarlo facilmente in un sistema di terza parte.

[34] L'espressione di interrogazione produce una serie di risultati, mostrati nella tipica griglia KWIC. In fondo, puoi sfogliare e leggere il documento eventualmente partendo da uno qualsiasi dei risultati (come qui, evidenziato in giallo). La mappa del documento a sinistra viene generata automaticamente in base al contenuto del documento. La parte di testo visualizzata corrisponde al nodo selezionato in quella mappa; se si seleziona il nodo radice, verrà mostrato l'intero documento.

In aggiunta, i dati dell'indice in sé non sono basati su un formato di archiviazione proprietario, ma su un RDBMS standard, che può essere facilmente integrato e utilizzato, anche bypassando il motore fornito. Il suo linguaggio di interrogazione è definito da una grammatica ANTLR, che funge da mediatore fra gli utenti e SQL; e l'intero processo che porta dal documento originale nel suo formato al modello utilizzato per la ricerca e la lettura è configurato in una *pipeline* modulare, definita in un file esterno. Il sistema può essere arricchito semplicemente introducendo nuovi moduli e concatenandoli nella *pipeline*, che può quindi essere utilizzata per aggiungere nuove fonti di dati da diverse analisi, fondendole tutte in una superficie uniforme. In tal modo, la peculiare natura dei documenti trattati da questo progetto e le sue esigenze, sia in termini linguistici che giuridici, possono ben trovare riscontro nel livello di astrazione e nell'elevata modularità della soluzione proposta.

13. Eventuali sviluppi futuri

Dato che i modelli e gli strumenti alla base di questo progetto sono stati disegnati proprio per costituire dei paradigmi autonomi, le prospettive per future espansioni di Atti Chiari si possono immaginare di pari passo con l'evoluzione del progetto stesso.

Ad esempio, sarà possibile avvalersi delle procedure già ampiamente sperimentate per ampliare il *corpus* degli atti, specie in vista di un maggiore bilanciamento rispetto alle loro tipologie (un'evoluzione in questo senso è già in corso).

Sul versante più tecnico, poi, è possibile accompagnare a questa espansione una nuova generazione di strumenti di pseudonimizzazione basati sul medesimo approccio, che potrà essere potenziato ulteriormente: ad esempio si potrebbe integrare un sistema NER (*Named-Entity Recognition*) per assistere la fase di marcatura dei documenti; e, dato che questa deve avvenire direttamente sui file ricevuti per poi sottoporli a pseudonimizzazione, l'idea è di inserire il nucleo del sistema esistente all'interno di una estensione di MS Word, accoppiando funzionalità esposte da un servizio API con una interfaccia a base HTML e JS (*JavaScript*) in un pannello di quell'applicativo. In tal modo gli utenti potranno accelerare in modo notevole il processo più lungo e delicato per l'adattamento degli atti al *corpus*, operando direttamente all'interno dell'applicativo di videoscrittura.

Ancora, sul piano della ricerca si potrebbero allargare le fonti e le tipologie dei metadati incorporati nel testo, cui questo tipo di architettura non pone alcun limite. Ad esempio, particolare interesse è emerso per tutti gli aspetti paragrafematici dei documenti anche in funzione delle limitazioni imposte alla redazione degli atti nelle nuove modalità digitali dell'amministrazione della giustizia: in tal caso, un sistema di analisi potrebbe operare direttamente sui documenti di videoscrittura per estrarne ulteriori dettagli stilistici e metrici variamente intersecabili con i metadati già esistenti. Si potrebbe, poi, esplorare l'analisi fonologica e ritmica dei testi laddove si intenda saggiare la possibilità di questo tipo di analisi negli atti più stilisticamente curati; anche qui, un esperimento è stato già condotto avvalendosi dei componenti di analisi fonologica forniti dal citato sistema Chiron alla stessa base del motore qui adottato.

Infine, il *corpus* con il suo patrimonio di metadati potrà fornire la base di una sistematica analisi lessicale della lingua giuridica, sì da costituire un nuovo componente di imprese lessicografiche ben più monumentali, già in corso di realizzazione.

Tutti questi esempi si riferiscono dunque a vie già esplorate e rese percorribili dalla disponibilità di strumenti concepiti per andare oltre il singolo progetto che ha costituito l'occasione del loro sviluppo, e proprio lo stesso motore di ricerca qui adottato, nato in tutt'altro contesto accademico, rappresenta in questo senso un chiaro paradigma pratico e metodologico.

References

[1] Buzzetti, D. 2002. "Digital Representation and the Text Model." *New Literary History* 33 (1), 61–87.

[2] Candrilli, Fernanda. 2021. "Il progetto di archiviazione e anonimizzazione." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 19–29. Viterbo: Sette Città.

[3] Caponi, Remo. 2014. "Il processo civile telematico tra scrittura e oralità." In *Lingua e processo. Le parole del diritto di fronte al giudice*, Atti del Convegno (Firenze, 4 aprile 2014), a cura di Federigo Bambi, 176–86. Firenze: Accademia della Crusca.

[4] Cavallone, Bruno. 2010. "Un idioma coriaceo: l'italiano del processo civile." In *L'italiano giuridico che cambia*, Atti del Convegno (Firenze, 1 ottobre 2010), a cura di Barbara Pozzo e Federigo Bambi, 85–95. Firenze: Accademia della Crusca.

[5] Chiari, I. 2012. "Corpora e risorse linguistiche per l'italiano. Stato dell'arte, problemi e prospettive." *Italienisch* 34 (2), 90–105.

[6] Clemenzi, Laura. 2021. "L'interrogazione della base dati Atti Chiari." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 41–52. Viterbo: Sette Città.

[7] Conte, Giuseppe. 2013. "Il linguaggio della difesa civile." In *Lingua e diritto. Scritto e parlato nelle professioni legali*, a cura di Alarico Mariani Marini e Federigo Bambi, 35–67. Pisa: Pisa University Press.

[8] Cresti, Emanuela, e Alessandro Panunzi. 2013. *Introduzione ai corpora dell'italiano*. Bologna: Il Mulino.

[9] Dalianis, Hercules. 2019. "Pseudonymisation of Swedish Electronic Patient Records Using a Rule-Based Approach." In *Proceedings of the Workshop on NLP and Pseudonymisation*, edited by Lars Ahrenberg and Beáta Megyesi, 16–23. Turku: Linköping Electronic Press. https://aclanthology.org/W19-6503.

[10] Dell'Anna, Maria Vittoria. 2014. "Fra attori e convenuti. Lingua dell'avvocato e lingua del giudice nel processo civile." In *Lingua e processo. Le parole del diritto di fronte al giudice*, Atti del Convegno (Firenze, 4 aprile 2014), a cura di Federigo Bambi, 83–101. Firenze: Accademia della Crusca.

[11] Dell'Anna, Maria Vittoria (a cura di). In stampa. *Lingua e scrittura forense. Storia, temi, prospettive*. Torino: Giappichelli.

[12] Douglass, Margaret, Gari D. Clifford, Andrew Reisner, George B. Moody, and Roger G. Mark. 2004. "Computer-Assisted De-Identification of Free Text in the MIMIC II Database." *Computers in Cardiology* 31: 341–44.

[13] Elger, Bernice S., Jimison Iavindrasana, Luigi Lo Iacono, Henning Müller, Nicolas Roduit, Paul Summers and Jessica Wright. 2010. "Strategies for health data exchange for secondary, cross-institutional clinical research." *Computer Methods and Programs in Biomedicine* 99 (3): 230–251. https://www.sciencedirect.com/science/article/pii/S0169260709003046?via%3Dihub.

[14] Felicetti, Achille, e Francesca Murano. 2021. "La modellazione semantica delle entità testuali. Il modello CRMtex e la descrizione ontologica dei testi antichi." *Umanistica Digitale* 11: 163–75. https://umanisticadigitale.unibo.it/article/view/13674/13774.

[15] Freddi, Maria. 2019. *Linguistica dei corpora*. Roma: Carocci.

[16] Fusco, Francesca. 2021. "Marcatura linguistica e tutela della riservatezza nello studio di un corpus di scritture forensi." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 29–40. Viterbo: Sette Città.

[17] Fusco, Francesca. 2022. "Forestierismi e linguaggio giuridico contemporaneo: il caso degli atti di parte." *Testo e Senso* 24: 189–207. https://testoesenso.it/index.php/testoesenso/article/view/585.

[18] Fusco, Francesca. In stampa. "Salvis iuribus. Il latino negli atti di parte." In *Atti Chiari. Lingua e scrittura forense tra storia, temi, prospettive*, a cura di Maria Vittoria Dell'Anna, Torino: Giappichelli.

[19] Fusi, Daniele. 2020. "Text Searching Beyond the Text: a Case Study." *Rationes Rerum* 15: 199–230.

[20] Fusi, Daniele. 2021. "Digitalizzazione e marcatura XML degli atti." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 59–73. Viterbo: Sette Città.

[21] Gualdo, Riccardo, e Laura Clemenzi, eds. 2021. *Atti Chiari. Chiarezza e concisione nella scrittura forense*. Viterbo: Sette Città, 2021.

[22] Gualdo, Riccardo, e Maria Vittoria Dell'Anna. 2014. "Per prove e per indizi (testuali). La prosa forense dell'avvocato e il linguaggio giuridico." In *La lingua variabile nei testi letterari, artistici e funzionali contemporanei. Analisi, interpretazione, traduzione*, Atti del XIII Congresso SILFI, a cura di Giovanni Ruffino e Marina Castiglione, 623–35. Firenze: Cesati.

[23] Gualdo, Riccardo. 2021. "Chiarezza e concisione negli atti processuali." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 11–18. Viterbo: Sette Città.

[24] Lombardi, Giulia. 2021. "I vantaggi del programma an-tool." In *Atti Chiari. Chiarezza e concisione nella scrittura forense*, a cura di Riccardo Gualdo e Laura Clemenzi, 29–40. Viterbo: Sette Città.

[25] Mortara Garavelli, Bice. 2001. *Le parole e la giustizia. Divagazioni grammaticali e retoriche su testi giuridici italiani*. Torino: Einaudi.

[26] Mortara Garavelli, Bice. 2003a. "L'oratoria forense: tradizione e regole." In *L'avvocato e il processo. Le tecniche della difesa*, a cura di Alarico Mariani Marini e Maurizio Paganelli, 66–91. Milano: Giuffrè.

[27] Mortara Garavelli, Bice. 2003b. "Strutture testuali e stereotipi nel linguaggio forense." In *La lingua, la legge, la professione forense*, a cura di Alarico Mariani Marini, 3–19. Milano: Giuffrè.

[28] Noumeir, Rita, Alain Lemay & Jean-Marc Lina. 2007. "Pseudonymization of Radiology Data for Research Purposes." *Journal of Digital Imaging* 20 (3). 284–295. https://link.springer.com/article/10.1007%2Fs10278-006-1051-4.

[29] Oksanen, Arttu, Minna Tamper, Jouni Tuominen, Aki Hietanen & Eero Hyvönen. 2019. "ANOPPI: A Pseudonymization Service for Finnish Court Documents." In *Legal Knowledge and Information Systems. JURIX 2019: The Thirty-second Annual Conference*, edited by Michal Araszkiewicz and Víctor Rodríguez-Doncel, 251–54. Amsterdam: IOS Press. https://helda.helsinki.fi/handle/10138/315951.

[30] Sabatini, Francesco. 2003. "Dalla lingua comune al linguaggio del legislatore e dell'avvocato." In *L'avvocato e il processo. Le tecniche della difesa*, a cura di Alarico Mariani Marini e Maurizio Paganelli, 3–14. Milano: Giuffrè.

[31] Sprondel, Johanna T. 2014. "Toward a Humanities of the Digital? Reading Search Engines as a Concordance." In Bod R., Maat J., Weststejn T. (edd.). *The Making of the Humanities:* Volume III: *The Modern Humanities*, 479-493. Amsterdam.

[32] Visconti, Jacqueline. 2018. "La chiarezza tra superfluo e necessario". In AA.VV., *Breviario per una buona scrittura*, 15–19. Roma: Ministero della Giustizia. https://www.federnotizie.it/wp-content/uploads/2018/10/BREVIARIO_ATTI_PROCESSUALI.pdf.