L'UTILIZZO DELL'HANDWRITTEN TEXT RECOGNITION NEI SISTEMI DI CONSULTAZIONE DEGLI ARCHIVI: IL CASO DELL'ARCHIVIO STORICO MEDIOBANCA "VINCENZO MARANGHI" E TRANSKRIBUS.

Silvia Carboni
Taddeo Molino Lova

Il Mondo degli archivi
quaderno 1/2025

Introduzione

Nel panorama degli archivi storici è possibile affermare che l'accesso[1] è la funzione oggi soggetta alle maggiori asimmetrie, che sono il prodotto di molteplici variabili che compongono i sistemi di consultazione di ogni archivio. Il numero delle variabili dipende soprattutto dalle scelte tecnologiche che si compiono e dal numero degli strumenti che si utilizzano: il rischio della proliferazione degli strumenti pone dunque la sfida dell'integrazione ottimale di tutti gli elementi.

Ci sono variabili che sono standard ormai affermati e a buon mercato, delle *commodities*, come la digitalizzazione del patrimonio documentario e il ricorso all'OCR (*Optical Character Recognition*) per fornire all'utente la trascrizione di materiale dattiloscritto.

La novità di questi ultimi anni è invece la maggiore accessibilità degli strumenti che permettono il riconoscimento automatico di testi manoscritti grazie all'HTR (*Handwritten Text Recognition*). Rispetto all'OCR, che si basa sul riconoscimento di un carattere alla volta, l'HTR guarda a un'intera linea di testo e cerca di decodificarne i caratteri grazie all'utilizzo di modelli di Intelligenza Artificiale[2]. Una piattaforma che permette di trascrivere documenti manoscritti sfruttando questa tecnologia è Transkribus.

Attualmente, sebbene la letteratura su Transkribus sia in crescita, il quadro degli studi sulle sue possibili applicazioni e integrazioni in ambito archivistico è ancora frammentario[3]. Questo contributo ha quindi l'obiettivo di fornire un quadro del possibile impiego di Transkribus negli archivi, presentando un caso di studio concreto di utilizzo della piattaforma su documenti manoscritti del XX secolo e dell'integrazione delle trascrizioni in un sistema di consultazione.

1 Inteso secondo la definizione dello standard OAIS.

2 Si tratta di *Artificial Neural Networks, vedi nota 18 e* Felix Dietrich, *OCR vs. HTR or "What is AI, actually?"*, https://readcoop.eu/insights/ocr-vs-htr/ (consultato il 21 sett. 2024).

3 Beatrice Couture, Farah Verret, Maxime Gohier, E Dominique Deslandres, *The Challenges of HTR Model Training: Feedback from the Project Donner Le Gout de l'archive a l'ere Numerique*, «Journal of Data Mining & Digital Humanities, Historical Documents and automatic text recognition», 2022, p. 2; Joe Nockels et al., *Understanding the application of handwritten text recognition technology in heritage contexts: a systematic review of Transkribus in published research*, «Archival Science», n. 22 (2022), p. 368, 376, 378-380, DOI: https://doi.org/10.1007/s10502-022-09397-0.

1. Mediobanca e l'archivio Storico "Vincenzo Maranghi"

Mediobanca è nata nel 1946 come istituto di credito speciale, con l'obiettivo di favorire la ricostruzione, lo sviluppo e l'internazionalizzazione delle industrie italiane nel secondo dopoguerra. Fu fondata per iniziativa di Raffaele Mattioli (al tempo Amministratore Delegato della Banca Commerciale Italiana) e di Enrico Cuccia, che fu Amministratore Delegato di Mediobanca fino al 1982 e poi Presidente Onorario fino alla sua morte nel 2000. Attività principale della banca è stata la concessione di crediti a medio-lungo termine. Da metà anni '50 ha iniziato anche attività proprie di una banca d'affari (emissione di azioni e obbligazioni per aumenti di capitale, quotazioni in borsa, fusioni e acquisizioni, …), che svolge tuttora.

L'istituto ha aperto al pubblico il proprio archivio nel settembre 2019, a seguito di un lavoro archivistico iniziato nel 2015 secondo un processo che prevede censimento, acquisizione[4], ricondizionamento, cartulazione, inventariazione e descrizione e che culmina con la scansione di tutti i documenti "lavorati". La diretta conseguenza è stata la scelta di una modalità di consultazione "smaterializzata", esclusivamente online[5].

Inizialmente la consultazione era possibile solo nella sede di Mediobanca, attraverso un portale esposto sulla rete intranet interna. Nel 2020, il blocco di tutte le attività dovuto alla pandemia ha imposto di ripensare questo modello di consultazione. Il sistema è stato quindi aggiornato e nell'aprile 2022 il nuovo portale di consultazione è stato messo online, rendendo disponibili a chiunque le descrizioni e le scansioni dei documenti. Gli utenti possono liberamente navigare in tutto il sito fino al livello delle descrizioni, mentre la visione dei documenti scansionati è possibile solo previa creazione di un account. Non è richiesta alcuna forma di autorizzazione per vedere i documenti e i dati di registrazione sono stati minimizzati come richiesto dal GDPR.

[4] I documenti di un qualsiasi ufficio, ritenuti di interesse storico, passano formalmente sotto la responsabilità dell'Archivio storico.

[5] https://archiviostorico.mediobanca.com/patrimonio/home.html (consultato il 18 ott. 2024).

Il sistema poggia sulla piattaforma di gestione documentale xDams[6], open-source e che adotta lo standard XML. xDams si basa inoltre sull'osservanza degli standard internazionali di descrizione archivistica ISAD(G) e ISAAR(CFP). Ogni elemento dell'alberatura ha una propria scheda descrittiva e le immagini sono associate alle schede sotto forma di "allegati digitali". Sia le schede descrittive che gli allegati digitali possono avere due livelli di visibilità: pubblica (se anteriori al 31/12/1973[7]) o riservata (se posteriori al '73 o nei casi previsti dalla legge per dati sensibili o sensibilissimi). Ciò che nel back-end ha visibilità pubblica è consultabile anche dagli utenti nel front-end, mentre il rimanente è accessibile solo alla consultazione interna del personale dell'Archivio.

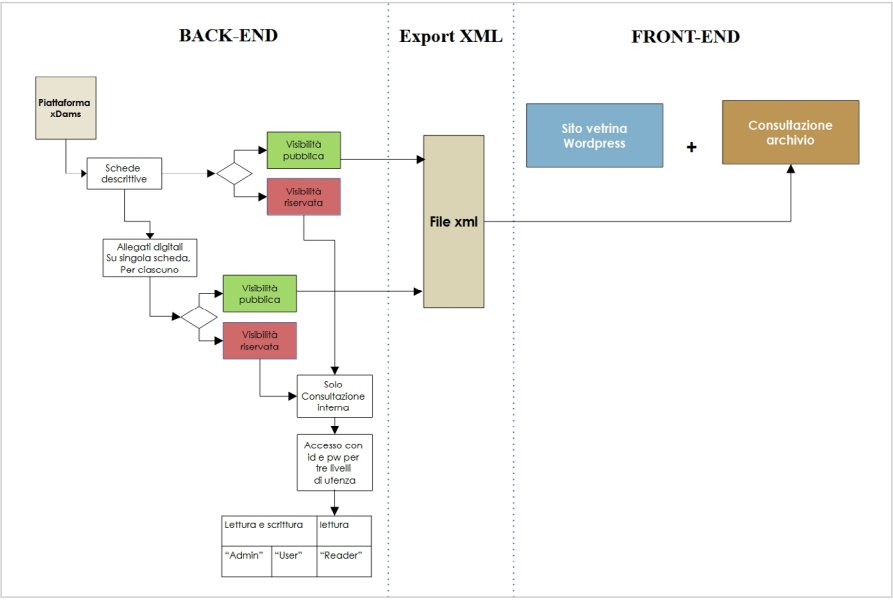

Diagramma 1 - *Funzionamento di xDams*

[6] https://github.com/xdamsorg/xDams-core (consultato il 5 sett. 2024).

[7] Da regolamento, l'Archivio Storico mantiene una finestra di riservatezza di 50 anni. Nel maggio 2023 è avvenuto il secondo grosso rilascio di documenti. Il primo, come detto, risale al 2019.

L'Archivio ha su propri server le immagini dei documenti in diversi formati e risoluzioni: TIFF per la conservazione a lungo termine e JPG per la consultazione. Di quest'ultimo ne esistono tre versioni: una di "media qualità" con apposto un watermark, che è il file effettivamente visualizzato dagli utenti; una versione di "bassa qualità" (utilizzata per la preview delle immagini); e una di "alta qualità", ovvero 300 dpi e senza watermark.

Il front-end, invece, si compone di due ambienti: un "sito-vetrina" realizzato con Wordpress e la sezione dedicata alla consultazione. Senza che l'utente percepisca alcun distacco, si esce dal CMS[8] Wordpress per passare a pagine popolate con le informazioni presenti nelle schede descrittive (solamente quelle pubbliche in xDams). Da qui l'utente, cliccando un bottone, accede alla visione delle immagini. Il *viewer* utilizzato è *Bookreader*[9]: open-source, personalizzabile e progettato da Internet Archive[10].

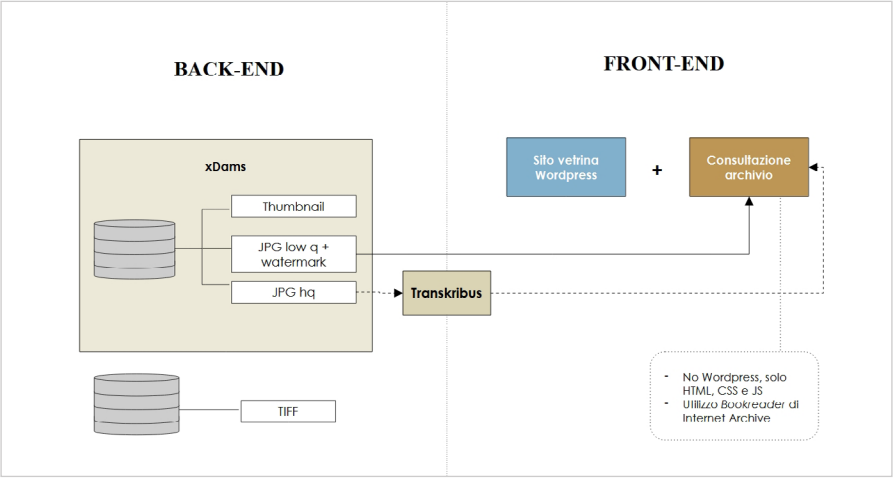

Diagramma 2 - *Gestione delle immagini nel back-end e front-end*

8 Content Management System.

9 https://github.com/internetarchive/bookreader/ (consultato il 4 sett. 2024). Strumento in uso al momento della redazione dell'articolo.

10 https://archive.org/ (consultato il 4 sett. 2024).

La reportistica interna sugli accessi e i feedback diretti ricevuti dagli utenti mostrano come la consultazione da parte degli utenti sia condizionata dalla natura dei documenti e dalla complessità della struttura. I fondi più consultati sono quelli relativi all'Alta direzione, nello specifico le carte della "Segreteria Generale e dell'Amministratore Delegato Enrico Cuccia". Un subfondo che invece risulta essere meno consultato è "Scritture Societarie", che raccoglie i "Verbali delle Assemblee Generali", i "Verbali del Consiglio di Amministrazione" e i "Verbali del Comitato Esecutivo".

Sulla scarsa consultazione hanno inciso diversi fattori. Se tutti gli altri fondi conservati dall'Archivio hanno un livello di descrizione analitico, "Scritture Societarie" presenta invece una descrizione sintetica (presenti e Ordine del Giorno). Questo fattore si incrocia con la numerosità delle sedute, la lunghezza dei verbali, la loro natura manoscritta e l'assenza di strumenti di accesso coevi. Mancano infatti rubriche o indici degli argomenti trattati e gli Ordini del Giorno fino agli anni '70 sono estremamente generici.

I Verbali sono quindi una parte "muta" dell'Archivio: questa considerazione ha dato il via a un percorso per renderli "parlanti" e più accessibili agli utenti, con l'obiettivo di uniformare l'esperienza di consultazione di tutti i fondi oggi consultabili. Si è scelto quindi di trascrivere i Verbali, per permettere agli utenti di leggerli più agevolmente e in modo da indicizzarne il testo nel motore di ricerca del sito dell'Archivio. Per automatizzare, almeno in parte, il processo di trascrizione è stato deciso di utilizzare la piattaforma Transkribus e sfruttare l'aiuto dell'Intelligenza Artificiale.

La possibilità di ottenere buoni risultati con i testi era infatti supportata da una esistente casistica, sia in termini di tipologia di istituzione che di documenti. Guardando agli archivi in ambito bancario, l'Archivio Storico del Banco di Napoli ha infatti avviato nel 2018 un progetto per ottenere con Transkribus la trascrizione delle pandette prodotte dagli istituti bancari napoletani del XVI secolo[11].

[11] Sabrina Iorio, *L'utilizzo della piattaforma Transkribus nell'Archivio Storico del Banco di Napoli: il "Progetto Pandetta"*, «Quaderni dell'Archivio Storico (nuova serie online) della Fondazione Banco di Napoli», a. 1, n. 1 (2019), p. 195-207.

In termini di tipologia documentale, vi sono alcuni casi noti di utilizzo su registri manoscritti e verbali. Per esempio il Gemeente Amsterdam Stadsarchief (Amsterdam City Archive), che ha trascritto i registri notarili dei secoli XVI-XIX da esso conservati[12]. Oppure, parlando più propriamente di verbali, si segnalano i progetti "Records of Low German Urban Diets" del Research Centre for Hanse and Baltic History (FGHO), che vede la trascrizione dei verbali dei consigli cittadini della Lega Anseatica[13]; oppure ancora i verbali del consiglio cittadino di Zurigo[14] o della città di Bautzen in Germania[15].

Da notare però che si tratta prevalentemente di documenti di età moderna e che riguardano enti cittadini, non imprese; e non vi sono casi di utilizzo per documenti di età contemporanea, salvo non dichiarati. Al momento della stesura di questo articolo, si nota inoltre come Transkribus sia poco utilizzato da istituzioni che conservano documenti in lingua italiana. Rispetto ai casi presentati, questa era infatti l'incognita nel caso dell'Archivio Storico Mediobanca, che si inserisce in una sottocategoria poco rappresentata: documenti di età contemporanea e in italiano.

12 https://amsterdam-city-archives.transkribus.eu/ (consultato il 9 sett. 2024).
13 https://rezesse-niederdeutscher-staedtetage.transkribus.eu/ (consultato il 9 sett. 2024).
14 https://ratsmanuale-zuerich.transkribus.eu/ (consultato il 9 sett. 2024).
15 https://transkribus.eu/r/bautzen-ratsprotokolle/#/ (consultato il 9 sett. 2024).

2. Transkribus

2.1 Principi di funzionamento

La piattaforma Transkribus è nata nell'ambito del progetto UE Horizon 2020 "READ"[16] ed è stata successivamente sviluppata grazie al lavoro dalla cooperativa READ-COOP SCE. Il suo punto di forza è la possibilità di utilizzare modelli di Intelligenza Artificiale (AI) per riconoscimento del layout e del testo di documenti manoscritti. È possibile applicare modelli già esistenti, creati e resi disponibili dallo staff o dagli utenti, oppure creare modelli personalizzati per i propri documenti.

Transkribus processa i documenti su propri server, rendendo accessibile l'HTR anche a chi non dispone di macchine con la necessaria potenza di calcolo. Il primo passo necessario nel flusso di lavoro è quindi caricare il proprio materiale sulla piattaforma, in formato JPG o PDF. Precedentemente o contestualmente al lancio del processo di *text recognition*, è necessario eseguire il riconoscimento del layout delle pagine: anche in questo caso si applica un modello di AI per individuare l'area della pagina contenente il testo (*text region*) e, all'interno di questa, le linee di testo (*baselines*)[17]. I modelli di *text recognition*, infatti, analizzano e tentano di riconoscere solo i segni grafici presenti sulle linee.

Il riconoscimento del testo si basa sull'utilizzo di algoritmi che analizzano i pixel di un'immagine alla ricerca di pattern associati a caratteri alfanumerici, confrontandoli con il materiale visto in fase di addestramento[18]. Di conseguenza, i modelli di riconoscimento del testo sono

[16] https://cordis.europa.eu/project/id/674943/it?isPreviewer=1 (consultato il 9 sett. 2024).

[17] I layout models di Transkribus si basano su un approccio in due fasi. Nella prima, un deep neural network analizza i pixel delle immagini e li classifica in baseline, separator e other. La classe separator indica l'inizio o la fine di una linea. Nella seconda fase, la classificazione del neural network viene utilizzata per un processo di clustering che riunisce i pixel che secondo il modello appartengono a una stessa baseline. Vedi Tobias Grüning, Gundram Leifert, Tobias Strauß et al., *A two-stage method for text line detection in historical documents*, International Journal on Document Analysis and Recognition (IJDAR),n. 22 (2019), p. 285–302, DOI: https://doi.org/10.48550/arXiv.1802.03345.

[18] Nel caso dell'HTR, data l'enorme variabilità delle calligrafie, l'associazione tra pattern e caratteri non è predefinita (come per l'OCR) ma viene generata da un *neural network*, che confronta i dati del ground truth per fare una previsione su quale carattere corrisponde a un certo segno grafico. Vedi Felix Dietrich, *OCR vs. HTR or "What is AI, actually?"*, https://readcoop.eu/insights/ocr-vs-htr/ (consultato il 4 sett. 2024).

calibrati sulle specifiche "mani" (calligrafie) su cui sono stati addestrati, quindi è spesso consigliabile creare un modello personalizzato sulle proprie mani o esigenze specifiche di trascrizione[19]. L'uso di un modello "proprietario" può essere necessario anche per il riconoscimento del layout, se le pagine dei documenti hanno strutture complesse o se si vuole essere sicuri che vengano inclusi o esclusi degli specifici elementi (ad esempio numeri di pagina o altri *marginalia*).

Per l'addestramento, è necessario creare un *ground truth*: un set di pagine già trascritte dall'utente o in cui è stato manualmente tracciato un layout perfettamente aderente a quello che si vorrebbe ottenere. Si tratta fondamentalmente di un campione di esempi, da cui il modello impara per poi poter replicare il risultato. Nel caso di modelli di *layout analysis*, è necessario un *ground truth* di almeno 20 pagine; nel caso del *text recognition*, il numero minimo di pagine è di circa 50 per mano, quindi più sono le mani per cui il modello deve essere adatto e più cresce il numero di pagine necessario.

Per valutare la bontà di un modello, Transkribus utilizza alcune pagine di *ground truth* come set di validazione: il modello non le incontra mai durante l'addestramento, poi gli viene chiesto di riconoscerne il testo o layout e il risultato viene confrontato con il *ground truth*. Viene calcolata una percentuale di errore, che può essere stimata sulle parole (*word error rate*, *WER*) o sui singoli caratteri (*character error rate*, *CER*)[20]. Quest'ultima modalità di calcolo è quella più comunemente utilizzata nella piattaforma. Un *CER* nel range 2-8% è considerato un buon risultato[21].

Se l'esito dell'allenamento non è soddisfacente, è possibile "raffinare" il modello aumentando il numero di pagine di *ground truth*. In termini pratici, viene in realtà creato un nuovo modello da zero, indicando però di usare come base (*base model*) il modello creato in precedenza. Utilizzare come base un altro modello è sempre possi-

[19] Il modello è in grado di riconoscere pattern nella trascrizione come lo scioglimento di specifiche abbreviazioni o utilizzi specifici della punteggiatura. Ad esempio, gli si può insegnare a trascrivere tutti gli acronimi puntati in acronimi senza punti (quindi, ad esempio, la sigla "S.p.A." verrebbe trascritta "SpA" e così tutti i casi analoghi).

[20] https://web.archive.org/web/20240222183814/https://readcoop.eu/glossary/character-error-rate-cer/ (consultato il 5 sett. 2024).

[21] Read-Coop Sce, *How to improve the CER of your model*, https://readcoop.eu/how-to-improve-the-cer-of-your-model/ (consultato il 5 sett. 2024).

bile, anche al primo tentativo di addestramento, appoggiandosi per esempio ai modelli creati da altri utenti.

Inoltre, nel processo di addestramento del modello di *text recognition* viene anche generato un *language model*[22], che può risultare utile per un riconoscimento più accurato delle parole.

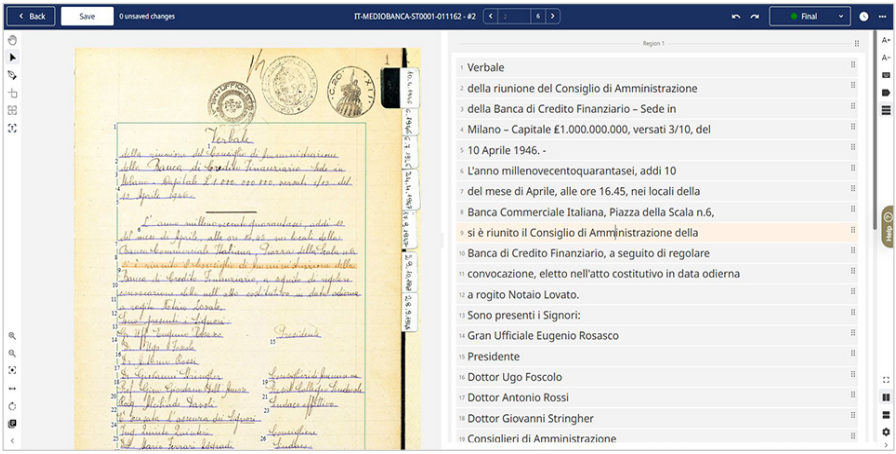

Figura 1 - *Interfaccia della piattaforma*

2.2 Formati

Transkribus permette ai propri utenti di scaricare le trascrizioni in diversi formati: .docx, .txt, PDF, XML TEI, XML PAGE e ALTO XML (questi due ultimi formati sono associati a un file METS contenente i relativi metadati). Inoltre offre un proprio servizio per la pubblicazione di trascrizioni e immagini dei documenti, Transcribus Sites (precedentemente denominato Read&Search). Ognuno di questi formati presenta delle specifiche caratteristiche.

22 Un *language model* è un modello probabilistico di linguaggio naturale, ovvero progettato per comprendere e generare il linguaggio umano. Quindi data una frase o una parola, cerca di prevedere quale potrebbe essere la parola seguente in base al lessico e alle strutture sintattiche del materiale su cui è stato addestrato. Vedi Djoerd Hiemstra, *Language Models*, in *Encyclopedia of Database Systems*, a cura di Ling Liu, M. Tamer Özsu, Boston, Springer, 2009, p. 1591-1594, DOI: https://doi.org/10.1007/978-0-387-39940-9_923.

I file in .txt e .docx contengono il solo testo trascritto. Nel .txt, ogni riga del file corrisponde a una *baseline* del documento in Transkribus; è possibile scaricare sia un file .txt per ogni pagina del documento, sia un file che contiene il testo dell'intero documento, con la suddivisione in pagine segnalata da due righe vuote. Per il formato .docx è possibile scegliere una formattazione del testo simile al .txt (una riga nel file corrisponde a una riga del documento), oppure una soluzione per cui ogni pagina corrisponde a un paragrafo del file .docx, senza rispettare la suddivisione originale del testo in linee.

I file in PDF contengono invece due layer, uno con le immagini del documento e l'altro con la trascrizione; il layer con la trascrizione non è visibile, ma la sua presenza permette di fare ricerche testuali all'interno del file. È però possibile anche generare PDF con delle pagine aggiuntive che contengono solo il testo (rendendolo quindi visibile e leggibile), intervallate alle immagini.

Per quanto riguarda le tre codifiche XML (ALTO, PAGE e TEI), tutte e tre contengono il testo del documento e informazioni sul suo layout (*text regions*, *baselines*, specifici tag strutturali come titolo, paragrafo,…) grazie a un sistema di tag che identifica le diverse informazioni. Sono file strettamente associati a quelli contenenti le immagini del documento trascritto: ogni pixel dell'immagine è identificato da coordinate e i file XML si rifanno a queste coordinate per indicare la posizione di ogni regione o linea di testo[23].

[23] PAGE, ALTO e TEI, pur avendo alcune similitudini, presentano differenze nella struttura, nei tag utilizzati e nella tipologia di informazioni che possono contenere (oltre a testo e layout). Per maggiori informazioni sul formato PAGE, vedi Stefan Pletschacher, Apostolos Antonacopoulos, *The PAGE (Page Analysis and Ground-Truth Elements) Format Framework*, in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR2010) (Istanbul, August 23-26 2010)*, IEEE-CS Press, p. 257-260. Riguardo TEI, vedi https://tei-c.org/. Per il formato ALTO, vedi https://www.loc.gov/standards/alto/about.html (consultati il 18 ott. 2024).

Invece Transkribus Sites, come accennato, è il servizio offerto da Transkribus per permettere agli utenti la condivisione all'esterno delle trascrizioni. Viene creato un sito web che si collega nativamente a una o più "collezioni"[24] di documenti e mostra a chi lo visita testo e immagini affiancati (come nell'interfaccia di trascrizione in Transkribus). C'è anche la possibilità di fare ricerche tramite parole chiave.

I formati TEI e ALTO sono accessibili sottoscrivendo uno dei piani di abbonamento a pagamento di Transkribus, mentre gli altri formati sono inclusi nel piano base (gratuito). Anche per utilizzare Sites è necessario avere un abbonamento dedicato.

[24] In Transkribus, la "collezione" è il contenitore dei documenti, a loro volta composti dalle immagini corrispondenti. Le collezioni sono a un unico livello e non esistono sotto-collezioni che replichino strutture archivistiche complesse (secondo ISAD). All'interno di una collezione è però possibile inserire un tag nei metadati dei documenti per indicarne l'eventuale livello gerarchico. Il tag è identificato come "Hierarchy (Split hierarchy levels with /)" e permette di replicare la struttura gerarchica ISAD solo se si fa corrispondere una collezione a una serie.

3. La trascrizione del fondo "scritture societarie"

Come accennato, "Scritture Societarie" comprende tre serie: "Verbali delle Assemblee Generali", "Verbali del Consiglio di Amministrazione" e "Verbali del Comitato Esecutivo". Si tratta complessivamente di circa 10.000 pagine, per la quasi totalità manoscritte (solo i documenti posteriori al 1980 sono dattiloscritti). Rispetto a questo materiale, il lavoro con Transkribus è iniziato dai Verbali del CdA, serie comprendente circa 3.000 pagine manoscritte.

Come precedentemente illustrato, l'attuale sistema di consultazione mostra all'utente le immagini dei documenti scansionati con un watermark in sovraimpressione, che avrebbe però interferito con il processo di *layout analysis* in Transkribus. Sono stati allora caricati sulla piattaforma file JPG con risoluzione 300 dpi e senza watermark.

Un primo passaggio fondamentale è stata l'analisi del layout dei documenti, per valutare quali elementi fossero fondamentali e quali trascurabili o di intralcio. Un test con il modello "base" di *layout analysis* proposto da Transkribus ha evidenziato alcune criticità: veniva creata una *text region* dedicata per il numero di pagina (ritenuto invece, in questa fase del lavoro, superfluo e da escludere), venivano riconosciute linee "fantasma" nel timbro apposto in cima a ogni pagina ed erano inclusi elementi ritenuti non rilevanti o di difficile trascrizione, come le firme dei presenti alla riunione o la vidimazione del notaio alla fine di ogni registro. Questi elementi sono stati eliminati su circa 50 pagine, che sono state utilizzate per creare un modello di *layout analysis* dedicato (tasso di errore 6.3%). È stato successivamente creato un secondo modello di *layout analysis* (tasso di errore 5.6%) da applicare alle sole pagine con tabelle di bilancio, per ovviare alle difficoltà riscontrate dal primo modello: commetteva diversi errori nel riconoscimento delle righe e nel numerarle secondo l'ordine corretto.

Ottenuti buoni risultati nell'analisi del *layout*, il secondo passaggio è stato verificare la numerosità delle "mani" presenti nel corpus documentale, identificandone 16. I verbali erano redatti dalle segretarie della banca, che avevano studiato calligrafia: scrivevano tutte in modo tendenzialmente posato e con una forma simile seppur con differenze di *ductus*, per cui la maggior parte delle mani presenta una propria unicità ma con forti analogie alle altre. Per il *ground truth* su cui addestrare il modello di *text recognition* sono state allora selezio-

nate circa 25 pagine per ognuna delle 6 mani più numericamente prevalenti, per un totale di 140 pagine.

Si era scelto un *ground truth* relativamente ridotto rispetto all'indicazione "standard" di 50 pagine per mano, con l'idea di fare un primo test ed eventualmente procedere ad aumentare gradualmente il *ground truth*. I risultati sono però stati fin da subito molto positivi: Transkribus ha calcolato un *CER* del 2.9%.

Il modello è stato poi testato su un campione di pagine per ogni mano, in modo da calcolare il tasso di errore specifico per ognuna[25]. Per trascrivere i documenti, il modello è sempre stato applicato spuntando l'opzione "*language model*": i verbali hanno un lessico molto specifico, legato alla natura delle operazioni svolte dalla banca, e si è ritenuto che l'applicazione del *language model* associato al modello di riconoscimento dei caratteri potesse rendere la predizione delle parole più accurata. Forse grazie alle similitudini tra alcune calligrafie e alla scelta di usare il *language model*, questo test ha mostrato risultati positivi anche per le mani non presenti nel *training set*, con CER nel range del 2-8%.

Risultava potenzialmente problematica la mano n. 5: è la calligrafia più diversa da tutte le altre, è presente in molti verbali e il tasso di errore calcolato sul campione di pagine è superiore al 5% pur trattandosi di una mano presente nel training set (al contrario delle altre mani su cui il modello è stato addestrato). Ciò significa che, nonostante il modello "conosca" questa mano grazie all'addestramento, fatica però a riconoscerla. Perciò, in un secondo momento, è stato creato un modello specifico per essa, utilizzando come *ground truth* esclusivamente pagine "mano 5" (*CER* 4%). Si è provato anche a creare un modello con pagine "mano 5" come *ground truth* e il modello "1.0" come *base model*: dà però risultati peggiori rispetto al modello addestrato solo su materiale "mano 5", supportando ulteriormente la teoria che si tratti di una mano troppo distinta dalle altre.

[25] Vedi *Tabella 1*. La possibilità di confrontare ground truth e trascrizione automatica per calcolare la percentuale di errore non è, al momento della scrittura di questo articolo, presente nella versione web di Transkribus, ma solo in quella desktop (Transkribus Expert). Il campione selezionato comprendeva 10 pagine per ogni mano. I modelli di text recognition sono stati applicati insieme al relativo language model.

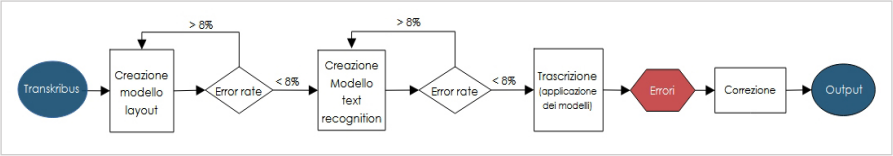

Diagramma 3 - *Flusso di lavoro con Transkribus.*

Dati i buoni risultati dell'addestramento, è quindi iniziato il processo di trascrizione dei Verbali delle riunioni del Consiglio di Amministrazione. La scelta dell'Archivio è di avere la trascrizione esatta dei documenti, quindi anche se gli errori commessi dai modelli nella trascrizione e, in misura minore, nell'analisi del *layout* sono pochi, è necessaria un'operazione di revisione e correzione manuale dei testi.

A settembre 2024, un anno dopo l'addestramento del modello "1.0" e avendo ormai trascritto un sufficiente numero di pagine per ogni mano, si è deciso di provare ad addestrare un nuovo modello, con un *ground truth* di 50 pagine per ognuna delle mani presenti nel subfondo Scritture Societarie (salvo le n. 2 e 13, che contano meno di 10 pagine a testa, e la n. 5). Oltre a questo modello "2.0", è stato addestrato anche il modello "2.0 WBM"[26], che ha cioè come *base model* il modello "1.0". Anche questi modelli sono stati testati su tutte le calligrafie, dando risultati migliori rispetto alla versione precedente[27].

[26] WBM è acronimo di «*with base model*».

[27] Il modello "2.0 WMB" è stato reso pubblico sulla piattaforma Transkribus e può quindi essere utilizzato da altri utenti che volessero trascrivere verbali manoscritti del XX secolo in italiano. Vedi https://www.transkribus.org/model/italian-20th-century-minutes-mediobanca (consultato il 28 ott. 2024).

MANO	1.0		2.0		2.0 WBM			Mano 5		Mano 5 WBM	
	CER (%)	*WER (%)*	*CER (%)*	*WER (%)*	*CER (%)*	*WER (%)*		*CER (%)*	*WER (%)*	*CER (%)*	*WER (%)*
1	2.82	11.29	2.69	10.83	2.39	10.58		-	-	-	-
2	-	-	-	-	-	-		-	-	-	-
3	4.79	14.02	2.99	7.54	3.04	7.37		-	-	-	-
4	6.53	18.05	0.66	2.44	0.31	1.31	(1)	-	-	-	-
5	5.68	20.52	4.10	13.11	3.71	12.70		3.93	14.55	4.22	15.19
6	4.57	17.89	2.76	10.90	2.83	10.90		-	-	-	-
7	6.39	23.47	0.72	2.82	0.16	0.66	(1)	-	-	-	-
8	6.09	19.67	0.75	2.53	0.35	1.06	(1)	-	-	-	-
9	6.56	20.71	2.10	8.39	2.09	8.39		-	-	-	-
10	5.03	18.55	1.62	5.45	1.90	8.07		-	-	-	-
11	5.21	19.31	1.31	5.53	1.11	5.05		-	-	-	-
12	8.95	24.47	2.01	6.02	2.00	6.46		-	-	-	-
13	-	-	-	-	-	-		-	-	-	-
14	2.26	7.01	0.91	3.17	1.93	3.48		-	-	-	-
15	3.73	13.65	1.60	6.00	1.22	4.47		-	-	-	-
16	6.30	19.37	0.57	2.11	0.52	2.11	(1)	-	-	-	-

Tabella 1 - *CER e WER dei modelli addestrati, testati su un campione di 10 pagine per ogni mano. Il test non è stato eseguito per le mani n. 2 e 13 perché presenti in meno di 10 pagine. (1): si noti che per le mani n. 4, 7, 8 e 16 le percentuali di errore per i modelli "2.0" e "2.0 WBM" sono drasticamente diminuite rispetto a "1.0" perché esse sono presenti in un solo verbale, che è stato utilizzato sia per l'addestramento dei modelli "2.0" che per il test.*

4. Considerazioni sull'implementazione delle trascrizioni

4.1 Condizioni

L'esperimento di addestramento dei modelli aveva avuto successo e si era quindi impostato un *workflow* efficace per la trascrizione. Ma subito erano emersi nuovi interrogativi: come si potrebbero integrare i testi prodotti da Transkribus nell'attuale sistema di consultazione? Oppure, come potrebbe quest'ultimo evolversi per accoglierli?

Quando l'Archivio Storico Mediobanca ha iniziato questo progetto, non sono stati trovati studi che potessero guidare un archivio nell'integrare le trascrizioni prodotte da Transkribus nel proprio sistema di consultazione. Consci che si trattava soprattutto di una questione informatica, è stato necessario condurre uno studio sistematico delle possibilità offerte da ogni formato in cui la piattaforma permette di scaricare i testi.

Dal confronto con la direzione dell'Archivio, è emerso come l'obiettivo primario del lavoro fosse rendere i Verbali "parlanti" grazie alle trascrizioni e migliorarne l'accesso. Si è stabilito che per raggiungere questo obiettivo fosse indispensabile rispettare due condizioni:

C_1 Includere il testo nei risultati della ricerca libera nel sistema di consultazione.

C_2 Rendere sempre visibile e leggibile il testo per l'utente.

Oltre a queste due condizioni necessarie, è stata posta una terza condizione C_3 non strettamente essenziale ma gradita:

C_3 Avere una modalità di visualizzazione che unisce testo e immagine del documento, affiancati (come nell'interfaccia di Tranksribus).

Il raggiungimento delle condizioni avrebbe portato a un miglioramento nell'esperienza di consultazione per l'utente e avrebbe sanato l'asimmetria nell'accesso alle Scritture Societarie rispetto agli altri fondi conservati dall'Archivio.

Per ogni formato di output sono state studiate le attività e gli strumenti necessari per integrarlo nel sistema (attuale o con le necessarie modifiche) e per arrivare a una o più delle condizioni fissate. Inoltre, nel valutare quanto ogni formato fosse adatto alle esigenze di questo

caso specifico, si è tenuto conto dell'impatto in termini di *usability/ user experience* per l'utente e di *performance* a livello informatico[28].

Per alcuni formati, però, sono emerse delle limitazioni: alcune legate al funzionamento di Transkribus e altre dipendenti da scelte nell'addestramento dei modelli. Sebbene la loro presenza non vada ad inficiare il raggiungimento delle condizioni poste, risolverle porta a un ulteriore miglioramento della loro fruizione. Si è quindi stimato l'eventuale margine di miglioramento per questi formati e quanto tempo e lavoro aggiuntivo ciò avrebbe richiesto.

4.2 Limitazioni

Come detto sopra, alcune limitazioni erano e sono legate al funzionamento di Transkribus e altre dipendono dall'addestramento dei modelli. Inoltre, una parte è risultata essere comune a più formati.

4.2.1 A capo

Un problema è la gestione delle parole che vanno a capo. Come accennato, i modelli di *text recognition* di Transkribus leggono e interpretano solo i segni grafici presenti sulle *baselines*: per loro, quando termina una linea finisce anche l'ultima parola su quella linea. Di conseguenza, le due metà delle parole "spezzate" dall'andare a capo non vengono riunite, salvo che nei formati .docx, PDF e XML ALTO (gli unici che identificano i segni «-» a fine linea come segnale di a capo). E quindi, se l'utente dovesse ricercare proprio una di queste parole nel sistema, non otterrebbe tra i risultati il caso in cui è spezzata dal a capo.

Nella fase di addestramento, è stato insegnato al modello di *text recognition* a riconoscere e trascrivere il carattere «-». Ma il modello non è in grado di capire che se l'ultimo carattere di una *baseline* è

[28] Le categorie di *usability*, intesa come interazione tra utenti e sistema, e di *performance*, intesa come interazione tra sistema e contenuto, derivano dal modello di interazione tra utenti e ambienti digitali *Interaction Trypthic Model.* Vedi Giannis Tsakonas, Christos Papatheodorou, *Exploring usefulness and usability in the evaluation of open access digital libraries*, «Information Processing & Management», a. 44, n. 3 (2008), p. 1234-1250. Cfr. anche Pierluigi Feliciati, *L'usabilità degli ambienti bibliotecari e archivistici digitali come requisito di qualità: contesto, modelli e strumenti di valutazione*, «Italian Journal of Library, Archives and Information Science», vol. 7, n. 1 (2016), p. 113-130.

«-», ciò indica un andare a capo. E, vista l'estrema varietà di parole coinvolte in questa casistica, che sono sempre diverse, non è nemmeno possibile risolvere la questione provando a insegnargli a completare le parole "spezzate" a fine riga: si baserebbe su un ventaglio molto ristretto di esempi forniti.

È stato calcolato che, sul totale delle parole di un Verbale, quelle che si trovano a fine riga e quindi potenzialmente spezzate sono in media il 15%. Ciò significa che la limitazione riguarda al massimo il 15% delle parole e che l'utente può cercare con successo nell'85% del testo estratto. Date queste percentuali, si era scelto di non correggere manualmente e di fornire una trascrizione aderente all'originale, in attesa di capire come sarebbero state integrate le trascrizioni nel sistema: se sarebbe stata possibile una correzione automatizzata o se si sarebbe adottato un formato che nativamente aggira questa limitazione.

4.2.2 Tabelle

Un'altra limitazione riguarda invece le tabelle. Molti verbali hanno infatti tabelle, contenenti dati legati alle diverse voci di bilancio, inframmezzate a paragrafi di testo. Transkribus permette di creare delle tabelle[29], ma quando queste non hanno una struttura "standardizzata" e sono inframmezzati da testo, il processo risulta complesso. Con Transkribus infatti si possono creare *table models* per i layout tabellari, ma solo se le tabelle hanno una struttura regolare e se le pagine contengono solo tabelle (per esempio, in un registro contabile); altrimenti è necessario creare a mano la *table region* e associare, sempre a mano, ogni riga alla cella corrispondente.

In fase di creazione dei *layout model* per i Verbali, si era deciso di fare in modo che il modello creasse una riga per ogni cella delle tabelle, senza utilizzare una *table region* e rimandando a una fase successiva l'eventuale sistemazione della struttura per renderla più leggibile o aderente a quella originaria del verbale.

[29] https://help.transkribus.org/tables (consultato il 9 sett. 2024).

4.2.3 Numeri di pagina

Nella creazione dei *layout model* si era anche deciso di escludere dalla *text region* il numero apposto in cima a ogni pagina dei registri dei verbali. Successivamente, ci si è resi conto che la mancanza del numero di pagina nel testo non era rilevante in caso di formati che consentono una visualizzazione "testo + immagine", ma sarebbe stato utile con i formati di testo semplice (.docx, .txt) per una migliore leggibilità.

La creazione di un nuovo modello di *layout* non è stata considerata come possibilità: avrebbe richiesto un alto costo in termini di tempo, soprattutto a fronte delle soluzioni automatiche che, come si vedrà, erano state ipotizzate; e sarebbe stato necessario ri-applicarlo a quanto già trascritto, annullando di fatto tutto il lavoro già fatto.

5. Studio dei formati

Come accennato, per ognuno dei formati in cui è possibile ottenere le trascrizioni, è stata pensata una possibile modalità di utilizzo e implementazione, che è stata valutata alla luce delle condizioni poste, delle limitazioni presenti e dell'impatto sia sull'attuale sistema informatico sia in termini di *user experience*. Per ogni formato, si è cercato di stimare, a grandi linee, i costi e i benefici associati, sia per una implementazione "di base" (che portasse al raggiungimento delle condizioni necessarie) che "migliorativa" (per risolvere gli aspetti ritenuti potenzialmente limitanti ma non tali da impedire il raggiungimento dell'obiettivo primario). In particolare, per valutare l'opportunità o meno di risolvere le eventuali limitazioni, ci si è posti la domanda «è accettabile?».

Il risultato di questo studio è stato tradotto in diagrammi di flusso, per evidenziare il ventaglio di scelte possibili e guidare l'Archivio nel selezionare la soluzione migliore per le proprie esigenze.

5.1 Txt

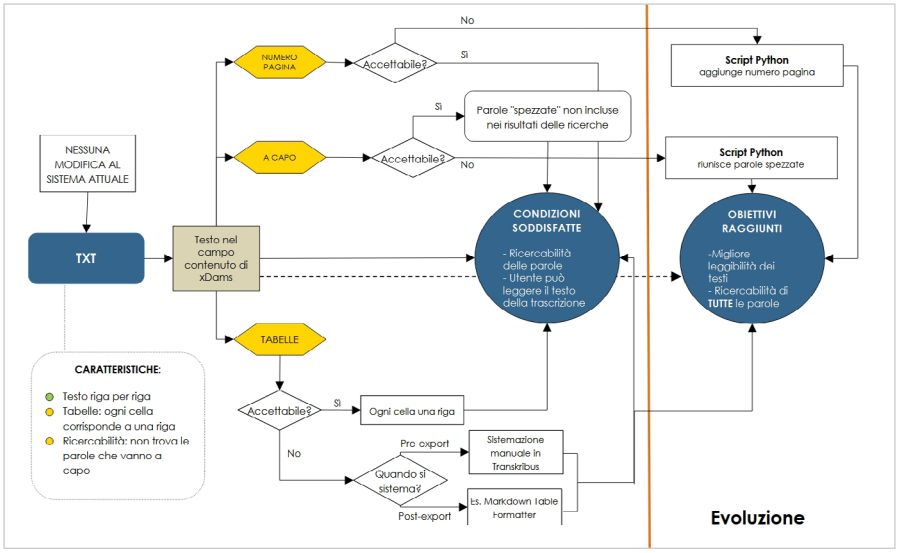

Diagramma 4 - *Flusso decisionale per il formato .txt*

Il testo in .txt prodotto da Transkribus per i verbali del CdA ha le seguenti caratteristiche:

- ogni riga di testo nel file coincide con una riga della pagina;
- le parole spezzate dal a capo non sono rimesse assieme;
- per le tabelle, ogni cella corrisponde a una riga.

Il punto di forza di questo formato, come si vedrà poi anche per .docx, sta nella sua "semplicità": se è vero che mette a disposizione "solo" il testo, ciò permette anche di manipolarne il contenuto molto facilmente. Per poterlo utilizzare è stata allora ipotizzata non tanto l'integrazione del file in sé in xDams, ma piuttosto di estrarne il testo e inserirlo nel campo "contenuto"[30] della scheda descrittiva, che è già indicizzato nel motore di ricerca del sito. Ciò lo rende visibile e leggibile all'utente nel front-end[31] e "agganciato" nelle ricerche, arrivando a soddisfare le condizioni C_1 e C_2 senza modifiche al sistema attuale. Va tuttavia considerata la sostenibilità e scalabilità in termini di prestazioni del sistema: con un alto numero di testi o con testi molto lunghi, si potrebbero appesantire le schede rallentandone quindi il caricamento e la navigazione.

Il formato presenta le seguenti limitazioni, in relazione ai Verbali:

- le parole che vanno a capo sono spezzate e non vengono riunite;
- manca il numero di pagina;
- il contenuto delle tabelle non si presenta come una tabella;

Da un lato, non si trattava di limitazioni problematiche, "*system-breaking*": ritenerle accettabili avrebbe permesso di raggiungere comunque le condizioni necessarie all'obiettivo primario, non avrebbe richiesto alcuna azione ulteriore rispetto al lavoro di base "obbligatorio" in Transkribus e si sarebbero ottenuti alti volumi di documenti trascritti in breve tempo.

Dall'altro lato, la possibile soluzione alle prime due limitazioni risultava essere poco costosa in termini di tempo e lavoro richiesto, a fronte di maggiore leggibilità e una ricercabilità del testo pari al 100% (contro l'85% di un testo non corretto). Si è pensato infatti di utilizzare uno

30 ISAD(G) 3.3.1 (Scope and Content).
31 Vedi *Diagramma 1*.

script in Python, creato e testato in una giornata di lavoro, che modifica il testo in pochi secondi e non richiede costi di licenza.

Nell'addestramento dei modelli e nella loro applicazione, si è prestata attenzione che l'andare a capo fosse segnalato dal simbolo «-»; mentre altri trattini a fine riga non segno di a capo sono stati preceduti da uno spazio, per distinguere i due casi. Lo script procede in questo modo: ricerca ogni carattere «-» alla fine di una riga di testo, valuta se prima del «-» vi è un carattere alfanumerico (quindi non uno spazio) e in caso positivo identifica i caratteri della riga successiva fino al primo spazio, li sposta alla riga precedente eliminando il carattere «-» e rimette insieme così la parola.

Per quanto riguarda invece il numero di pagina, nel file .txt il passaggio da una pagina del documento alla successiva è segnalato con due righe vuote: lo script dovrebbe cercarle nel testo e, ogni volta che le trova, aggiungere una riga con scritto «[Pagina X]», dove X è un numero progressivo a partire da una cifra indicata dall'esecutore (i registri contengono più verbali, quindi i documenti tendenzialmente non iniziano da pagina 1). Questo facilita all'utente dell'Archivio la consultazione: se, dopo aver letto la trascrizione, volesse consultare le immagini del documento, saprebbe facilmente quali sono le pagine con la porzione di testo di suo interesse.

Rispetto invece alle tabelle dei bilanci, si è detto che con il formato .txt a ogni cella corrisponde una riga di testo. Non è l'output peggiore: le tabelle risultano comunque interpretabili, seppur con minore immediatezza. Sono state vagliate due strade percorribili per la riformattazione: intervenire prima o dopo l'export del documento da Transkribus.

La prima soluzione comporta una correzione manuale: per ogni pagina con una tabella è necessario ridimensionare la *text region* esistente, inserire una *table region* apposita, disegnando a mano righe e colonne, e associare a ogni cella la linea di testo corrispondente (tramite una modalità *drag&drop* che si presta a frequenti errori). Per una correzione post-export, si è ipotizzato l'utilizzo delle convenzioni previste dal linguaggio di marcatura Markdown[32]: è necessario anche qui rimaneggiare il testo manualmente, con poi la possibilità di

[32] https://www.markdownguide.org/extended-syntax/#tables (ultima consultazione in data 6 sett. 2024).

renderlo più ordinato usando uno strumento automatico come Markdown Table Formatter[33] (che richiede però l'utilizzo di un font *monospaced*[34]).

Entrambe le soluzioni sono state valutate come molto costose in termini di tempo, in quanto gran parte del lavoro non sarebbe automatizzabile, ottenendo però tabelle di grande chiarezza e leggibilità.

5.2 Docx

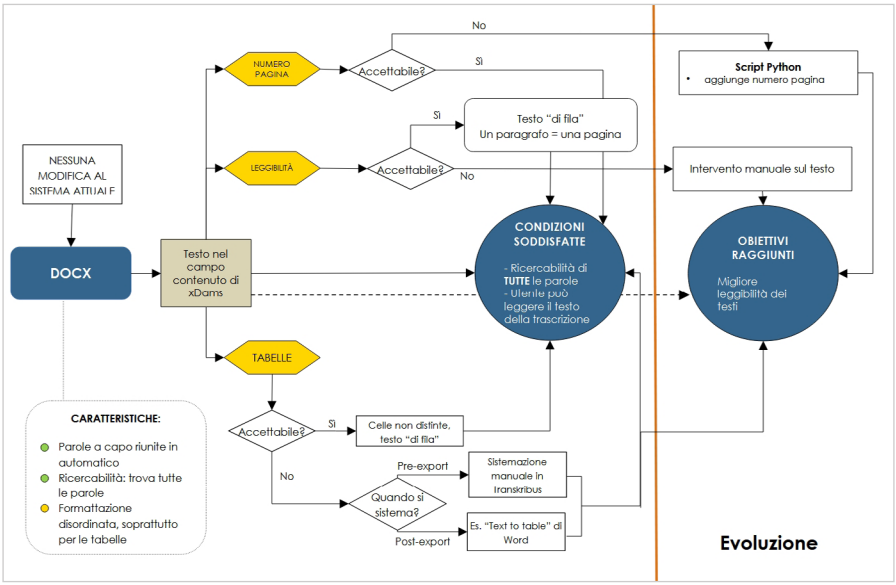

Diagramma 5 - *Flusso decisionale per il formato .docx*

33 http://markdowntable.com/ (ultima consultazione in data 6 sett. 2024).

34 Nei font monospaced i caratteri hanno tutti la stessa larghezza: Markdown Table Formatter si basa su questo presupposto per rendere le tabelle più visivamente ordinate e allineate.

Il testo in .docx prodotto da Transkribus per i verbali del CdA ha le seguenti caratteristiche:

- le parole spezzate dall'andare a capo vengono riunite in automatico;
- di conseguenza, tutte le parole rientrano nei risultati delle ricerche;
- le linee di testo nel file possono non corrispondere alle linee nelle pagine del documento[35].

Anche per questo formato, si è ipotizzato di estrarne il testo e inserirlo nel sistema di consultazione attuale senza apportarvi modifiche, sfruttando il campo "contenuto" della scheda di xDams. Si è ritenuto che questa soluzione soddisfi le condizioni C_1 e C_2, con un possibile valore aggiunto: rispetto a formati che non riuniscono le parole spezzate dall'andare a capo e che hanno quindi, nativamente, una minima percentuale di parole non "pescate" in una ricerca, qui invece il 100% delle parole è ricercabile.

Il formato ha però delle limitazioni, legate alla formattazione del testo:

- il contenuto delle tabelle non si presenta come una tabella;
- mancano i numeri di pagina;
- la formattazione del testo non consente un buon grado di leggibilità.

Nel determinare l'accettabilità di queste limitazioni, è stato seguito lo stesso ragionamento che per il formato .txt: da un lato, si tratta di aspetti che non inficiano il raggiungimento dell'obiettivo primario di dare voce ai Verbali; dall'altro, una limitazione su tre può essere aggirata facilmente.

[35] Come illustrato successivamente e nei diagrammi 5 e 6, il testo in .docx può presentarsi senza rispettare l'originale suddivisione in linee, con una pagina che corrisponde ad un paragrafo nel file, oppure può avere una suddivisione in linee aderente alla struttura del documento.

Rispetto al numero di pagina mancante, infatti, si è valutato l'intervento con uno script in Python. Nei file .docx prodotti da Transkribus, ogni pagina è trattata come un paragrafo e il passaggio da una pagina all'altra è segnalato da una spaziatura (*paragraph break*), quindi lo script può seguire lo stesso principio illustrato per i file .txt: ricerca delle spaziature e, ogni volta che una viene trovata, aggiunta all'inizio del paragrafo successivo della frase [Pagina X], dove X è un numero progressivo a partire da una cifra indicata dall'esecutore.

Per quanto riguarda invece la formattazione del testo, l'output in .docx, di default, non mantiene la suddivisione in righe del testo sulle pagine del documento: il testo è quindi presentato in una sorta di "flusso continuo" e questo comporta un limite in termini di leggibilità, sia in generale che per le tabelle. Per poter ricondurre il testo a una formattazione più simile all'originale, con la corretta divisione in paragrafi e linee di testo, l'unica soluzione che è stata individuata è un intervento manuale sul file.

Per le tabelle, si è pensato sia ad una soluzione pre-export che post. Nel primo caso, come per il formato .txt, per ogni pagina con una tabella bisogna ridimensionare la *text region* esistente, inserire una *table region* apposita, disegnando a mano righe e colonne, e associare a ogni cella la linea di testo corrispondente. Per una correzione a posteriori, si potrebbe usare l'opzione "Text to table" (nativamente presente in Word), che richiede comunque una pre-lavorazione manuale del testo. Ognuna delle soluzioni proposte porta a ottenere una migliore leggibilità dei testi, ma si è stimato un costo di implementazione molto elevato, soprattutto in termini di tempo necessario per svolgere le azioni necessarie su un alto volume di trascrizioni.

Nell'interfaccia di export di Transkribus, però, esiste un'opzione relativa al formato .docx che è stata ritenuta una possibile soluzione alternativa alle limitazioni di formattazione: "*preserve line break*", che mantiene le interruzioni di linea e dà al file .docx le stesse caratteristiche e limitazioni del file .txt:

- ogni riga di testo nel file coincide con una riga della pagina;
- le parole che vanno a capo sono spezzate e non vengono riunite;
- il contenuto delle tabelle non si presenta come una tabella;
- manca il numero di pagina.

Di conseguenza si è pensato a un secondo flusso di lavoro per il formato .docx, a tutti gli effetti identico a quello per il formato .txt.

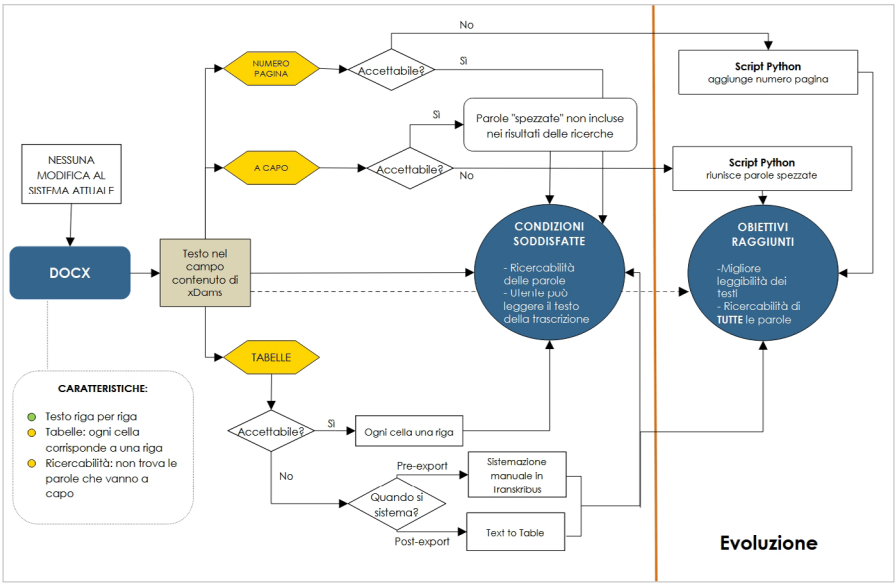

Diagramma 6 - *Flusso decisionale alternativo per il formato .docx*

L'eventuale scelta tra i due diversi file .docx sta nelle rispettive limitazioni e la loro accettabilità. È meglio avere un .docx che nativamente non presenta problemi con le parole spezzate dal a capo ma con una formattazione "limitante", oppure un .docx con leggibilità migliore ma dove alcune parole non sono ricercabili perché spezzate?

5.3 PDF

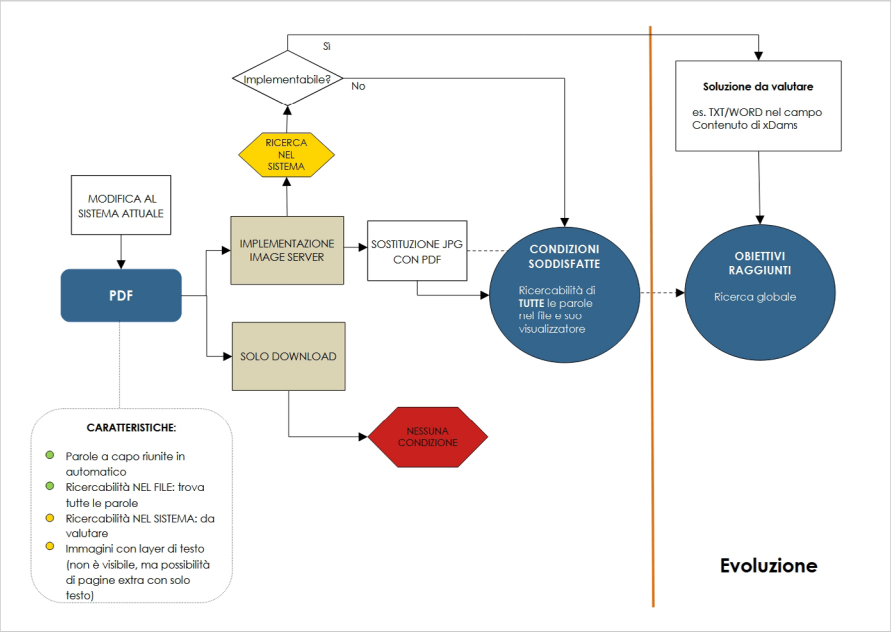

Diagramma 7 - *Flusso decisionale per il formato PDF*

Queste sono le caratteristiche del formato PDF:

- le parole spezzate dall'andare a capo sono automaticamente riunite;
- di conseguenza, tutte le parole sono ricercabili all'interno del file;

il file si compone di un layer con l'immagine e di un layer con il testo, che non è visibile dall'utente. È possibile però generare un PDF con pagine aggiuntive che mostrano solo il testo, senza immagine.

Per valersi di questo formato, sono state studiate due strade percorribili. La prima prevede di fornire il PDF solamente in download all'utente[36] e continuare a mostrare solo le immagini dei documenti, senza trascrizione. Valutando questa soluzione è però emerso come, sebbene sia sicuramente di più facile implementazione, non permette di raggiungere nessuna delle condizioni poste e quindi non porta a un miglioramento dell'accesso ai Verbali.

La seconda via percorribile prevede invece di sostituire i file JPG del Verbale con un file PDF (nella versione con pagine aggiuntive per il testo) e mostrarlo con *Bookreader* in una visualizzazione a due pagine, per avere testo e immagine affiancata. Si è ritenuto che così siano soddisfatte le condizioni C_2 e C_3, ma a patto di una modifica al sistema dal lato informatico: *Bookreader* gestisce formati per immagini come JPG e non supporta nativamente il PDF, quindi serve una componente lato server (un *image server*) che gestisca i PDF e li converta in un formato compatibile con *Bookreader*. Si tratta, chiaramente, di una modifica che comporta dei costi di implementazione maggiori.

Anche rispetto alla condizione C_1, quindi la ricercabilità del testo, si è giunti alla conclusione che per raggiungerla con questa soluzione è necessario un intervento che tenga conto dell'attuale meccanismo del motore di ricerca. Nel sistema di consultazione online dell'Archivio, la ricerca libera funziona tramite la piattaforma Apache Solr[37] e bisogna quindi trovare la giusta modalità per indicizzare il testo in Solr in modo che emerga nei risultati della ricerca, ad esempio copiando il testo dai file .txt o .docx in un campo indicizzabile della scheda xDams (visibile o anche nascosto agli utenti), sistemando le parole spezzate grazie allo script in Python per renderle ricercabili al 100%.

[36] Soluzione adottata, per esempio, nel precedente sito dell'Archivio Centrale dello Stato per fornire i PDF degli inventari. Il sito è ancora consultabile all'indirizzo https://search.acs.beniculturali.it/OpacACS/inventario/home (consultato il 9 sett. 2024).

[37] Solr viene alimentato con i documenti su cui si vuole fare una ricerca, questi vengono analizzati alla ricerca di parole chiave e poi viene creato un indice che associa ad ogni documento le sue parole chiave (indicizzazione). La ricerca ed estrazione di parole chiave avviene secondo uno schema predefinito dall'implementatore di Solr. Nel caso dell'Archivio, per esempio, ogni parola nel campo titolo, integrazione al titolo, contenuto e data della scheda di un documento è indicizzato. https://solr.apache.org/guide/7_2/a-quick-overview.html (consultato il 5 sett. 2024).

5.4 XML TEI

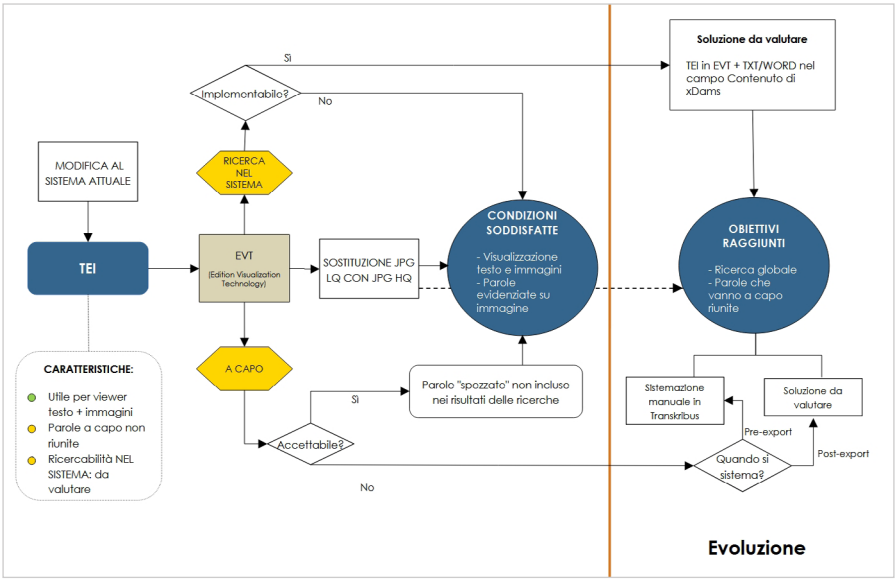

Diagramma 8 - *Flusso decisionale per il formato XML TEI utilizzando EVT*

Delle diverse codifiche XML presenti tra i formati di export di Transkribus (TEI, ALTO, PAGE), si è deciso di concentrare lo studio su TEI e ALTO.

Per quanto riguarda TEI, il suo utilizzo è molto diffuso nella creazione di edizioni digitali e nasce per iniziativa di un consorzio internazionale, di conseguenza ha alle spalle una comunità attiva e diversi strumenti open-source per la sua implementazione. Come accennato, sia TEI che ALTO, seppur con delle differenze, presentano queste caratteristiche: una sezione che, tramite tag appositi, identifica le linee di testo e le loro coordinate (in pixel) sul file JPG associato; una sezione che, sempre tramite l'utilizzo di tag, associa alle linee il testo corrispondente.

Per utilizzare il formato TEI, si è pensato all'utilizzo di un visualizzatore apposito, come EVT (Edition Visualization Technology)[38]. Si tratta di uno strumento open-source, sviluppato dall'Università di Pisa, che permette di creare edizioni digitali con file TEI e JPG. Tra le funzionalità implementabili vi sono la visualizzazione affiancata di testo e immagine, la ricerca all'interno del documento e l'*image-text linking* (cliccando una linea di testo nella trascrizione si evidenzia la linea corrispondente sull'immagine e viceversa). Sono aspetti che ricalcano la modalità di visualizzazione offerta da Transkribus Sites e che permettono di raggiungere le condizioni C_2 e C_3. Per quanto riguarda invece la condizione C_1, come per il formato PDF, è necessario trovare la giusta modalità per indicizzare il testo in Solr.

Nella valutazione di questa soluzione, è stata fatta una stima sommaria del relativo costo di realizzazione. In primis, l'effettivo inserimento di EVT nell'attuale sistema, da cui derivano una serie di ulteriori modifiche. Per esempio, la sostituzione o l'affiancamento degli attuali file JPG con watermark con quelli caricati in Transkribus e utilizzati per la trascrizione (300 dpi, no watermark): dato che nel file TEI le coordinate delle linee di testo hanno un sistema di riferimento vincolato al JPG corrispondente, si rende necessario questo scambio. Oppure ancora, come emerge dalla documentazione di EVT[39], può essere necessario anche: personalizzare il CSS per adeguarsi alla grafica del sito dell'Archivio; modificare i file di configurazione per attivare le funzionalità di ricerca, visualizzazione testo + immagine e *image-text linking*; inserire i file JPG in una struttura di *directories* come quella richiesta da EVT e, di conseguenza, verificare che i file TEI indichino il percorso corretto dei file immagine a cui sono associati.

Ci si è anche interrogati sull'efficacia di far coesistere nel proprio sistema due visualizzatori distinti, EVT per i documenti con trascrizione e *Bookreader* per il restante: è una soluzione efficiente dal punto di vista informatico? In termini di *user experience*, potrebbe creare confusione per l'utente?

[38] http://evt.labcd.unipi.it/ (consultato il 9 sett. 2024).

[39] https://github.com/evt-project/evt-viewer/blob/master/USER_README_EN.md (consultato il 9 sett. 2024).

Inoltre, anche l'export in TEI delle trascrizioni presenta la limitazione legata alle parole spezzate dall'andare a capo. Se l'indicizzazione del testo per la ricerca avviene non utilizzando il file TEI ma, per esempio, copiando il testo dal .txt alla scheda di xDams, la limitazione passa in secondo piano. È stato però comunque valutato come si potrebbe intervenire per una modifica, individuando due opzioni: correggere manualmente il testo pre-export, in Transkribus, o modificare il file TEI post-export. La prima opzione richiede un certo investimento di tempo per modificare il lavoro di trascrizione già fatto (da rivedere completamente), ma più ammortizzabile per il lavoro futuro, visto che si correggerebbe il testo una volta sola e la sistemazione degli errori di trascrizione è un passaggio obbligato per le esigenze dell'Archivio. Una modifica post-export è da considerare invece solo se automatizzabile, per esempio creando un apposito script in Python, visto che altrimenti risulta più immediata la correzione in Transkribus.

Nel caso del formato TEI, non è stata ritenuta invece una limitazione l'aspetto delle tabelle. Anche qui, come per i file .txt, ad ogni cella corrisponde una riga di testo. Con però una visualizzazione che affianca testo e immagine, la possibilità di vedere la struttura originaria della tabella rende nulle le potenziali difficoltà di lettura e interpretazione.

5.5 ALTO XML

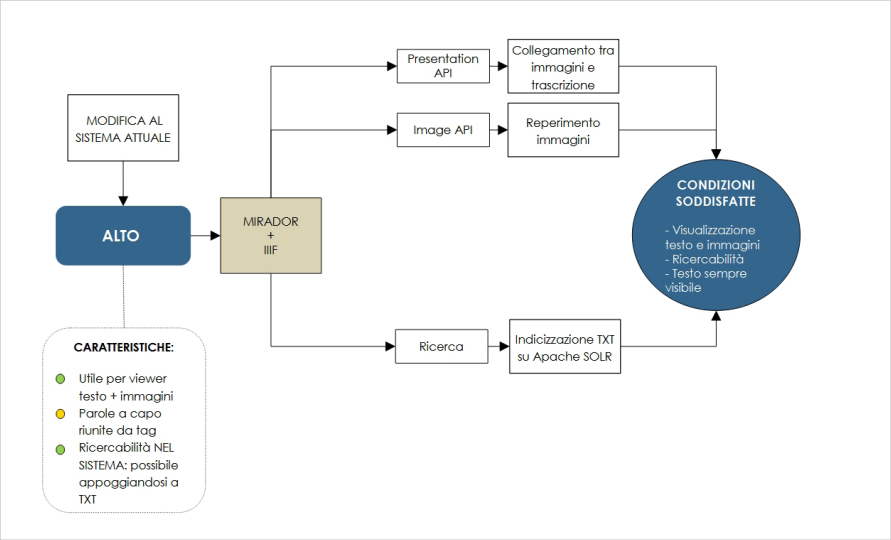

Diagramma 9 - *Flusso decisionale per il formato ALTO*

Considerando invece il formato ALTO, questo presenta le stesse caratteristiche del formato TEI ma con due notevoli differenze: i file ALTO generati da Transkribus possono contenere non solo le informazioni (coordinate e contenuto) riguardanti una linea di testo, ma anche quelle delle singole parole; e contengono nativamente un tag che segnala le parole spezzate dall'andare a capo e ne unisce le due metà[40].

Per poter utilizzare ALTO, si è ipotizzato di cambiare il *viewer* delle immagini, da *Bookreader* a *Mirador*[41], per appoggiarsi allo standard IIIF e le sue API Image e Presentation. IIIF (International Image Intero-

[40] Questo è un esempio sintassi, con un tag che contiene le informazioni riguardanti una parola all'interno di una linea di testo: <String HEIGHT="133" WIDTH="407" VPOS="933" HPOS="1923" CONTENT="Filodram" SUBS_TYPE="HypPart1" SUBS_CONTENT="Filodrammatici"/>.

[41] https://projectmirador.org/ (consultato il 9 sett. 2024). Mirador è un visualizzatore di immagini che supporta lo standard IIIF e le sue API.

perability Framework) è uno standard tecnologico che offre un insieme di specifiche e protocolli per consentire l'accesso e la condivisione di risorse digitali, in particolare le immagini, in modo interoperabile su diverse piattaforme e applicazioni[42]. Image API è il web service per il reperimento delle immagini, utilizzato per recuperare le immagini dal server e mostrarle in *Mirador* (come per TEI, bisognerebbe sostituirle con quelle a 300dpi senza watermark). Presentation API invece fornisce ai visualizzatori come *Mirador* le informazioni sulla struttura e il layout delle immagini, contenute nei relativi file Manifest[43], per consentirne la corretta visualizzazione. Grazie ad essa, inserita la trascrizione in ALTO nel Manifest di un Verbale, la si può rendere visibile in *Mirador* accanto alle immagini (fissa o attivabile/disattivabile con un pulsante)[44].

Esiste anche la Content Search API, il servizio creato dall'IIIF Consortium per la ricerca all'interno delle trascrizioni. Questa API è quindi pensata per la ricerca in una risorsa e non per trovare la risorsa stessa[45]. Per integrare i testi con la ricerca libera sul sito dell'archivio, e raggiungere la condizione C_1, si è pensato a una soluzione analoga a quella proposta per i formati TEI e PDF: indicizzazione del testo in Solr appoggiandosi al formato TXT.

Perciò, passando a un visualizzatore compatibile con IIIF, utilizzandone le API e sostituendo i file JPG attuali con quelli ad alta risoluzione e senza watermark, si crea un sistema con le seguenti caratteristiche:

- le parole sono indicizzate nel sistema di ricerca del sito;
- è possibile visualizzare testo e immagini affiancate;
- di conseguenza, il testo trascritto può essere sempre visibile e leggibile dall'utente.

42 IIIF consente agli utenti di visualizzare, ingrandire, annotare e confrontare immagini provenienti da diverse collezioni digitali. https://iiif.io/get-started/how-iiif-works/ (consultato il 9 sett. 2024).

43 Manifest è l'unità principale di IIIF, che elenca tutte le informazioni che compongono un oggetto IIIF. Comunica come visualizzare gli oggetti digitali e quali informazioni visualizzare su di essi, compresa la struttura, con vari gradi di complessità determinati dall'implementatore del sistema. Vedi https://iiif.io/guides/using_iiif_resources/ (consultato il 9 sett. 2024).

44 Vedi https://iiif.io/api/presentation/3.0/ e https://iiif.io/api/image/3.0/ (consultato il 22 ott. 2024).

45 Vedi https://iiif.io/api/search/2.0/ (consultato il 22 ott. 2024).

Sono quindi rispettate le condizioni C_1, C_2 e C_3.

Anche in questo caso, ci si è interrogati sul costo di implementazione e sulle ricadute in termini di *performance* e *user experience*. Cambiare il visualizzatore delle immagini e utilizzare lo standard IIIF sono modifiche sostanziali al sistema, sia in termini economici che per il lavoro necessario. D'altra parte, questa soluzione consente di adottare uno standard diffuso, ben supportato e che porta con sé una buona esperienza per l'utente: ad esempio un ampio controllo sulla visualizzazione dell'immagine (ruotarla, modificare luminosità e saturazione, visualizzarla in scala di grigi, …) o la possibilità di confrontare nello stesso visualizzatore documenti conservati da enti diversi (a patto che entrambi utilizzino IIIF). Inoltre, rispetto a un visualizzatore come EVT, che nasce per visualizzare immagini con file TEI associati, Mirador può essere utilizzato come visualizzatore per tutto il materiale dell'Archivio, con trascrizione o senza: ciò rende più fluida l'esperienza utente.

5.6 Transkribus Sites

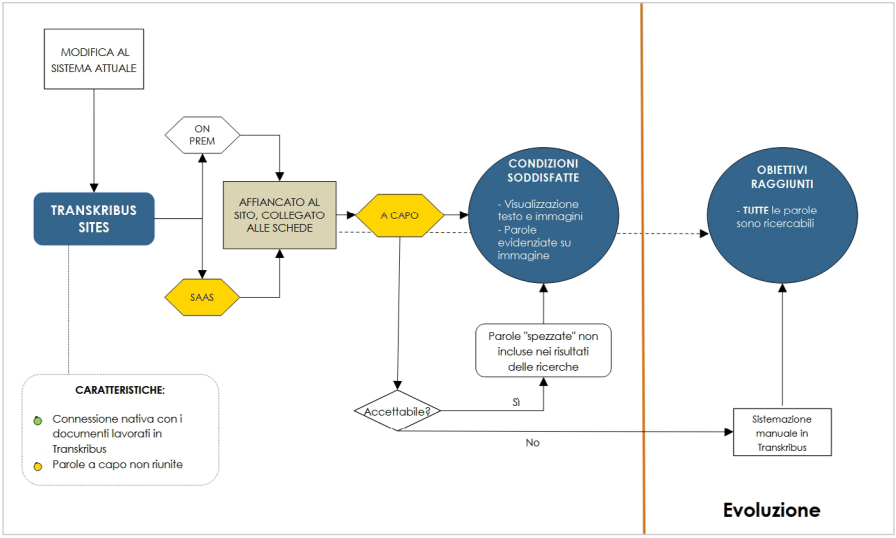

Diagramma 10 - *Flusso decisionale per Transkribus Sites*

Come accennato, Transkribus offre ai propri utenti una soluzione per la visualizzazione dei documenti integrata con la piattaforma. *Transkribus Sites*[46] ha le seguenti caratteristiche:

- visualizzazione di testo e immagini affiancati (e quindi il testo trascritto è sempre visibile all'utente);
- cliccando su una linea di testo nell'immagine viene evidenziata la linea corrispondente nella trascrizione e viceversa;
- è prevista una funzionalità di ricerca, sia all'interno del singolo documento che a livello globale (in tutte le collezioni messe a disposizione sul sito);
- si ha una connessione nativa con i documenti lavorati in Transkribus, quindi eventuali modifiche vengono aggiornate in Sites in automatico.

Nello studio di questo "formato", è emerso come le sue caratteristiche permettono di soddisfare le condizioni C_2 e C_3. Per quanto riguarda la condizione C_1, quindi l'integrazione con la ricerca libera dell'Archivio, essendo *Sites* una piattaforma distinta dal sito dell'Archivio non si può che pensare a una soluzione analoga a quella presentata per PDF e formati XML: indicizzare il testo di .txt o .docx in Solr.

Transkribus Sites è, come Transkribus, in primis una soluzione SAAS (Software As A Service)[47]. Quindi i siti creati con *Sites* hanno dominio *transkribus.eu* e sono ospitati sui server di READ-COOP. Ciò significa, per l'Archivio Storico Mediobanca, avere una "costola" separata dal resto del sito di consultazione. Si potrebbe certamente inserire nella scheda xDams del singolo Verbale un link al documento in *Sites*, per collegare le due piattaforme, ma come nel caso del formato TEI e di EVT, ci si è chiesti se possa essere una buona soluzione per l'utente o se non crei confusione in termini di accesso. Per l'Archivio, un sito "a parte" per le trascrizioni non è una soluzione ottimale, si vorrebbe avere un solo e unico ecosistema ben integrato (come accade già ora per il sito-vetrina e il portale di consultazione).

[46] Vedi https://www.transkribus.org/sites (consultato il 18 ott. 2024).

[47] La piattaforma è accessibile via web, su server remoti, ed è prevista la sottoscrizione di un piano di abbonamento.

Parte della problematica non si pone invece qualora si decida di passare alla versione On-Premises[48] di Transkribus: è infatti possibile avere un'installazione di Transkribus su macchine o server propri, sottoscrivendo una apposita licenza, e possibilmente anche *Sites* in questa configurazione. Quindi *Sites* potrebbe essere erogato dallo stesso server del sito di consultazione dell'archivio. Rimane però valida una considerazione: è efficace, in termini di esperienza utente, avere una modalità di visualizzazione delle trascrizioni diversa dal resto del materiale conservato dall'Archivio?

In ogni caso, la limitazione per i testi in *Sites* riguarda le parole spezzate dal a capo: anche qui vengono trattate come due parole distinte. Non è un problema se il testo utilizzato per l'indicizzazione in Solr proviene dai file .docx o .txt (opportunamente corretti), ma è problematico invece per la ricerca interna a Sites. L'unica soluzione, vista la connessione tra Transkribus e Sites, è la correzione manuale del testo: un processo inizialmente costoso in termini di tempo, dovendo ritornare su quanto già trascritto, mentre il costo futuro è quasi pari a zero, integrandosi nel processo di correzione già previsto per ogni documento.

[48] Vedi https://readcoop.eu/transkribus/on-prem/ (consultato il 18 ott. 2024).

6. Valutazioni

Alla luce dei punti di forza e debolezza di ogni formato che sono emersi da questo studio, il passaggio successivo del processo è stato il confronto tra i diversi output, per capire quale di questi offrisse il giusto equilibrio tra *desiderata*, costi di realizzazione e impatto in termini di *user experience*, integrazione a livello informatico e prestazioni del sistema.

Ne è risultato che, se l'Archivio si fosse trovato nella condizione di dover pubblicare le trascrizioni sul sito in tempi ristretti, il formato che avrebbe permesso di raggiungere le condizioni necessarie C_1 e C_2 con il minimo investimento di tempo e risorse era sicuramente il formato .docx, che rispetto a .txt ha anche il vantaggio di una nativa indicizzazione e ricercabilità del 100% delle parole nel testo.

Se invece si fosse scelto uno dei formati di puro testo per difficoltà nell'implementare una delle soluzioni più tecnicamente complesse, ma ci fossero stati il tempo e le risorse per un intervento migliorativo sul testo, si poteva utilizzare il formato .txt: si avrebbe avuto infatti una trascrizione con una formattazione più *user friendly*. In questi primi due scenari, però, ci sarebbe stato il rischio di dover scendere a compromessi in termini di prestazioni informatiche, se l'inserimento di numerose e lunghe trascrizioni nelle schede xDams avesse rallentato il sistema.

Si è allora concluso che nel caso ottimale, ovvero avendo a disposizione tempo e risorse, la soluzione migliore in termini di condizioni rispettate, esperienza utente e prestazioni del sistema sarebbe stato l'utilizzo del formato ALTO supportato da IIIF.

Quest'ultima via è quella che è stata poi effettivamente intrapresa dall'Archivio, grazie anche a una già avviata riflessione sull'adozione dello standard IIIF per rendere i propri asset digitali interoperabili. È quindi in corso una modifica della struttura del backend. *Bookreader* sarà sostituito con *Mirador* e, quando gli utenti vorranno visualizzare un documento, *Mirador* effettuerà richieste di dati a componenti del

backend: grazie alla Presentation API chiederà alla *Digital Library*[49] di erogare il Manifest della risorsa; interpretando il Manifest otterrà la trascrizione come elemento IIIF Annotation; mentre grazie all'Image API, chiederà all'*Image Server* di fornire le immagini nel formato e nelle dimensioni corrette; e tutto ciò permetterà una visualizzazione simultanea di testo e immagine.

In questa infrastruttura, l'*Image Server* recupera l'immagine da erogare dal server dell'Archivio e la *Digital Library* crea il Manifest a partire da una "lettura" delle "cartelle" contenenti testi e immagini sullo stesso server. Si è scelto di tenere trascrizioni e immagini in due strutture di "cartelle" separate: essendo i documenti trascritti solo una piccola parte del totale, la *Digital Library* può inserirne le trascrizioni nei rispettivi Manifest senza dover "leggere" le "cartelle" di tutto il materiale dell'Archivio. Ciò rende il sistema più veloce e più facilmente aggiornabile man mano che tutti i documenti di Scritture Societarie saranno trascritti. Il testo viene inserito nel Manifest a partire sia dal formato ALTO (da cui vengono estratte linee di testo e loro coordinate) che dal formato .txt (da cui si ottiene il testo dell'intero documento), precedentemente "migliorato" con lo script in Python. I file .txt vengono utilizzati anche per l'indicizzazione in Solr ai fini della ricerca libera nel sito dell'Archivio. Non sono inseriti nella scheda xDams, per evitare che sul lungo periodo il loro peso vada a rallentare il sistema: l'indicizzazione avviene direttamente a partire dai .txt sul server dell'Archivio.

[49] La *Digital Library* è lo strumento creata da Regesta, sviluppatore di xDams, per la gestione di asset digitali. È pienamente conforme allo standard IIIF e offre servizi di caricamento, condivisione, metadatazione, ricerca e conservazione degli asset digitali.

Conclusione

Gli Archivi Storici dovranno, d'ora in avanti, dare sempre maggiore attenzione al miglioramento dell'esperienza utente. In questi termini, Transkribus ha dimostrato di essere un valido aiuto, rendendo disponibili le trascrizioni integrali di fonti di difficile lettura.

Come si è visto, adottare questo strumento non è un "miracolo automatico". Anzi, complica il quadro generale, perché comporta la proliferazione degli strumenti. Si rende quindi necessario studiare la soluzione migliore, rispetto alla propria situazione, per integrarne i frutti.

Mediobanca ha scelto di puntare a una completa integrazione tra le trascrizioni e il sistema già esistente, modificando l'infrastruttura per renderla scalabile e sostenibile sul lungo periodo. Agli utenti l'ardua sentenza.