# Alfabeto arabo e OCR open source: un'analisi grafica e linguistica dei risultati di elaborazione per il recupero di dati catalografici

«DigItalia» 2-2025
DOI: 10.36181/digitalia-00150

**Riccardo Amerigo Vigliermo**
*Università di Modena e Reggio Emilia – FSCIRE*

*Il riconoscimento ottico dei caratteri (*Optical Character Recognition, OCR*) è uno strumento centrale per la conservazione di dati digitalizzati, ma mostra limiti significativi con alfabeti non latini, specie di fronte a stampe desuete o manoscritti, dove dati "rumorosi" e variabili contestuali ostacolano il riconoscimento ottimale. L'OCR è centrale nello sviluppo di applicazioni che prevedono l'impiego di altre tecniche di elaborazione del linguaggio naturale (*Natural Language Processing, NLP*). L'analisi degli errori, come fase di post-elaborazione (*post-processing*), può migliorare l'accuratezza soprattutto se combinata con un'analisi contestuale. Lo studio qui presentato ha l'obiettivo di delineare i tratti comuni degli errori commessi da tali OCR testati nel progetto Digital Maktaba.*

## Introduzione

Gli OCR mostrano ancora limiti nell'elaborazione di testi non nativi digitali, specialmente in ambito storico-religioso, a causa della natura eterogenea e "rumorosa" dei dati e delle difficoltà linguistiche, in particolare se applicati all'alfabeto arabo[1]. Spesso i risultati generati dai sistemi OCR si rivelano infatti inesatti, inficiando potenzialmente la validità di applicazioni di tecniche di *Natural Language Processing* (NLP), che utilizzano come dati di partenza i risultati degli OCR per compiti anche molto diversi tra loro[2]. In questo senso, studi recenti hanno indicato aree di ricerca promettenti concentrandosi in particolare sulle fasi di preparazione dei dati (*pre-processing*) e della loro correzione e rifinitura (*post-processing*). Ricerche recenti hanno indicato nei processi di *pre-* e *post-processing* una via per migliorare l'accuratezza, tramite correzioni contestuali, modelli probabilistici, soluzioni basate su regole (*rule-based*)[3]. A queste strategie di ottimizzazione si sono recentemente aggiunte le possibilità offerte dai nuovi sviluppi nel campo dei *Large Language Models* (LLM) e *Large Multimodal Models* (LMM) - le cosiddette Intelligenze artificiali - i cui studi hanno individuato nell'integrazione fra OCR e LLM

[1] Ahmed — Abidi 2019.
[2] Chiron et al. 2017.
[3] Nguyen et al. 2021, p. 9; Bassil — Alwani 2012; Afli — Loïc —Schwenk 2016; Khosrobeygi et al. 2020.

un punto di partenza di grande potenziale per il miglioramento dei processi di recupero delle informazioni[4].

Lo studio qui presentato vuole porre il focus sull'importanza dell'analisi dei risultati in contesti testuali multilingua[5] e ha l'obiettivo di delineare i tratti comuni degli errori commessi da alcuni sistemi OCR open-source testati nel corso del progetto Digital Maktaba (da qui in avanti DM), nato dalla collaborazione tra informatici, storici, bibliotecari, ingegneri e linguisti riuniti dalla start-up mim.fscire, dall'Università di Modena e Reggio Emilia (UniMoRe) e dalla Fondazione per le scienze religiose (FSCIRE)[6].

Il set di dati alla base di questo progetto proviene dalla vasta collezione di documenti in formato PDF messa a disposizione dalla Biblioteca "Giorgio La Pira" di Palermo, specializzata in Storia e dottrine dell'Islam e facente parte della FSCIRE. Sono stati condotti diversi test con soluzioni OCR open-source, per poter stabilire quale potesse fornire i migliori risultati su un sottoinsieme di frontespizi e copertine multilingua in alfabeto arabo (una selezione di 100 esemplari tra frontespizi e copertine, con vari *layout* e con diversi stili calligrafici scansionati in formato PDF). I frontespizi sono stati trattati con l'impiego di GoogleDocs[7], Tesseract[8] e EasyOCR[9], generando i risultati di seguito illustrati[10].

## Brevi note sulle caratteristiche generali della scrittura araba

Alcune caratteristiche generali della scrittura araba possono essere così riassunte:

– L'alfabeto arabo è esclusivamente corsivo con andamento da destra a sinistra[11].

– L'alfabeto è esclusivamente consonantico e omografico (si veda sotto). Le vocali (fatḥa "a", ḍamma "u" e kasra "i") e i punti diacritici sono espressi sopra o sotto il rigo. Le vocali lunghe (ā, ī, ū) sono sistematicamente rappresentate mediante i grafemi ا (alif), ي (yā', y) e و (wāw, w)[12].

– I grafemi sono dipendenti dalla posizione in cui compaiono nella parola e modificano la loro forma se sono isolati, in posizione iniziale, mediana o finale (es. ع عـ ـعـ ـع ).

– I grafemi ا و ر ز د ذ anche ژ in persiano) non legano a sinistra, creando sequenze grafiche non connesse da legature (ad esempio ضرب , أضرار ).

[4] Boros et al. 2024; Pakhale 2023.

[5] Smith — Cordell 2018.

[6] Da novembre 2022 fa parte del progetto ITSERR finanziato dal Ministero della ricerca italiano con i fondi del programma NextGenerationEU. L'obiettivo principale di DM è quello di sviluppare un sistema per l'estrapolazione automatica della conoscenza e la classificazione di documenti in alfabeti non latini (in particolare in arabo) in contesti di vasti archivi di documenti digitali (in formato PDF).

[7] https://christophersrose.com/2020/05/05/how-to-use-google-docs-to-ocr Arabic-text/.

[8] https://tesseract-ocr.github.io/tessdoc/.

[9] https://github.com/JaidedAI/EasyOCR/blob/master/README.md.

[10] I sistemi OCR open source hanno limitazioni di addestramento sui font, causando inesattezze con caratteri nuovi o misti (Zoizou — Zarghili — Chaker 2020, p. 576). Per questo motivo "errore" sarà utilizzato tra virgolette.

[11] Se si considera la resa delle cifre la scrittura può considerarsi anche bidirezionale, in quanto i numeri seguono un orientamento inverso. La compresenza di grafemi arabi e numeri può generare conflitti di lettura bidirezionale, aspetto assai rilevante per l'estrazione di metadati catalografici dai frontespizi. Ulteriori complicazioni possono derivare dalla coesistenza di numeri latini ( سنة 1345), arabi ( سنة ١٣٤٥ ) e persiani ( سال ١٣٤٥ ), con set grafici parzialmente diversi e codici Unicode distinti: numeri arabi (0660-0669) e numeri persiani (06F0-06F9), nonostante non tutte le cifre siano graficamente differenti.

[12] Garbini — Durand 1994.

– Alcune legature tipiche di alcuni stili calligrafici comportano modifiche nelle posizioni dei punti diacritici come anche del rigo, come accade ad esempio per lo stile *nastaʿliq*[13].

Di seguito si fornisce una tavola per la traslitterazione doppia basata rispettivamente sullo standard ISO 233-2:1993 per l'arabo e ISO 233-3:2023 per il persiano. Eventuali discostamenti da questi standard adottati nel presente studio sono indicati in nota.

| Arabo | traslitterazione | Persiano | traslitterazione |
|---|---|---|---|
| ا | v.note | ا | v.note |
| ء | ʿ (v.note) | ب | b |
| ب | b | پ | p |
| ت | t | ت | t |
| ث | ṯ | ث | ṯ |
| ج | ǧ | ج | ǧ |
| ح | ḥ | چ | č |
| خ | ḫ | ح | ḥ |
| د | d | خ | ḫ |
| ذ | ḏ | د | d |
| ر | r | ذ | ẕ |
| ز | z | ر | r |
| س | s | ز | z |
| ش | š | ژ | ž |
| ص | ṣ | س | s |
| ض | ḍ | ش | š |
| ط | ṭ | ص | ṣ |
| ظ | ẓ | ض | ẓ (v.note) |
| ع | ʿ | ط | ṭ |
| غ | ġ | ظ | ẓ |
| ف | f | ع | ʿ |
| ق | q | غ | ġ |
| ك | k | ف | f |
| ل | l | ق | q |
| م | m | ک | k |
| ن | n | گ | g |
| ه | h | ل | l |
| ة | v. note | م | m |
| و | w/ū | ن | n |
| ي | y/ī | و | v |
| ى | v. note | ه | h |
| | | ى | y |

[13] Il riconoscimento OCR dello stile *nastaʿliq* presenta criticità specifiche dovute alla diagonalità e alle legature che si sovrappongono sia orizzontalmente che verticalmente. La segmentazione richiede quindi un'analisi bidimensionale che tenga in considerazione sovrapposizioni di caratteri adiacenti e legature (Javed — Hussain 2009, p. 2), nonché righi multipli orizzontali e diagonali, fattori che aumentando la complessità del riconoscimento.

1. La *alif* ا se preceduta da vocale breve <a> segnala l'allungamento di quest'ultima e sarà traslitterata come <ā>.
2. L'articolo determinativo *al-*, quando presente, se preceduto dalle preposizioni *bi*, *li*, *ʿalā*, *ʻilā*, *fī* e le congiunzioni *wa* e *fa*, verrà riportato come segue: es. *wa-l-*, *fī-l-*, *bi-l-* ecc.
3. Nel caso di particelle come *bi*, *li*, *ʿalā*, *ʻilā*, *fī*, se seguite da un pronome enclitico, non saranno separate da quest'ultimo mediante un trattino (es. *fiha*, *ʿalayna*, *ʻilayhim* ecc.).
4. La *tāʾ marbūṭa* verrà traslitterata con *-a* (finale); in caso di stato costrutto (*iḍāfa*) o dopo una *alif* di prolungamento *ā* verrà traslitterata con *t* (es. *ṣalāt*).
5. La *alif madda* آ e la *alif maqṣūra* ى verranno traslitterate con *ā*.
6. Nel caso in cui il nome inizi in *alif-hamza* <ʾ>, quest'ultima sarà omessa e verrà resa maiuscola la vocale adiacente, come ad esempio in: **أبو** *Abū*. Nel caso in cui il nome inizi in *ʿayn* <ʿ> verrà resa maiuscola la vocale adiacente successiva.
7. Per quanto riguarda la traslitterazione del grafema ض con <ẓ> in ISO 233-3: 2023 (persiano), nel presente studio si è mantenuta la <ż> della versione precedente ISO 233-3:1999.

## Similarità e omografia

Da un punto di vista terminologico è necessario precisare due concetti per una migliore comprensione delle relazioni grafiche tra i vari grafemi. Si tratta di relazioni scontate ad un occhio "umano", ma da non sottovalutare nel contesto del riconoscimento "automatico".

Il primo concetto è quello di similarità dei grafemi. In questo caso due grafemi possono condividere parti del tratto calligrafico[14], sopra o sotto al rigo, che se non perfettamente riprodotte possono portare a confondere un grafema per l'altro, (per esempio: <*d*> د e <*r*> ر). Il secondo riguarda invece l'omografia[15], caratteristica tipica dell'alfabeto arabo che vede due o più grafemi, generalmente legati in una stringa (o parola), condividere uno o più glifi. I grafemi vengono distinti solamente mediante l'utilizzo di uno, due o tre punti diacritici, come ad esempio: <*ṭ*> ط e <*ẓ*> ظ. In questo caso, i punti diacritici sono l'unico fattore discriminante tra caratteri, di conseguenza il loro mancato riconoscimento può generare forme omografe valide a livello linguistico ma non esatte dal punto di vista del risultato, come anche forme non esistenti.

[14] Per *glifo* si intende qui l'unità grafica minima, distinta dal *grafema*, che ha invece valore linguistico; la precisazione è necessaria poiché non tutti i glifi sono grafemi. Con il termine "carattere" invece si indicherà l'insieme di glifi e grafemi in un particolare stile calligrafico e tipografico (al pari del termine inglese *font*).

[15] L'omografia era già nota ai grammatici arabi con il nome di ištirāk lafẓī o ištirāk lafẓī ḫaṭṭī, si veda a tal proposito (Munaǧǧid 1999, p. 30). Gli studi distinguono generalmente l'omografia in eterografa e omofona: la prima occorre quando vengono a mancare i segni diacritici di vocalizzazione (es. **كتب** kataba, kattaba, kutiba ecc.), la seconda invece quando due parole identiche e con la stessa vocalizzazione recano due significati diversi (**بيت** bayt = "casa" o "verso di poesia") (Taouka — Coltheart 2004, p. 32).

## Metodologia

Dal punto di vista operativo, i frontespizi sono stati automaticamente estratti e convertiti in formato PNG utilizzando il modulo Fitz della libreria PymuPDF[16], preservando la risoluzione originale. Tale conversione si è resa necessaria poiché i sistemi OCR open-source analizzati non supportano input PDF, ad eccezione di GoogleDocs.
La scelta dell'impiego dei frontespizi e delle copertine per formare il set di campioni è stata operata seguendo i seguenti criteri:

– presenza/assenza di frontespizio o copertina;
– presenza di sfondi di vario colore;
– varietà nella qualità delle immagini pdf;
– varietà nello stato di deterioramento dei testi;
– varietà di "rumore" presente sul testo;
– rappresentazione proporzionale delle lingue dell'archivio (arabo, persiano, turco azero)[17];
– rappresentazione della varietà di stili calligrafici arabi utilizzati in diversi volumi, in diverse pagine e anche nella medesima pagina;
– presenza di testo vocalizzato/non vocalizzato;

Dal momento che la natura dei risultati dell'elaborazione di un sistema OCR è soggetta a numerose variabili, gli esempi proposti e la loro categorizzazione hanno valore puramente dimostrativo, senza pretendere di definire esaustivamente tutti i possibili esiti. Data la fluidità della calligrafia araba, mappare ogni errore sarebbe poco produttivo; si analizzano invece gli errori più distinguibili e frequenti tramite una categorizzazione *ad hoc*. Ad ogni categoria corrisponde una tabella dove la colonna di destra rappresenterà il carattere di input (Cinp) mentre la colonna di sinistra riporta l'output (Cout). L'analisi considera due livelli di variabili: "interne" (relative a riconoscimento di singolo grafema e sistema OCR) ed "esterne" (estranee ma ugualmente influenti sul riconoscimento).

È inoltre necessario tenere in considerazione una serie di variabili grafematiche, come:

– differenze tra vari alfabeti a base arabo (ad esempio persiano);
– carattere non nativo digitale / carattere nativo digitale (es. Tasmeem Emiri)[18];
– i caratteri seguono gli stili calligrafici arabo-persiano (*nasḫ*, *nastaʿlīq* - *taʿlīq*, *kūfī*, *ruqʿa*, *diwānī*, *ṯuluṯ*, *muḥaqqaq*, *tawqīʿ*)[19];
– i caratteri "artistici" sono quelli ideati appositamente per un particolare frontespizio (ogni carattere ha peculiarità grafiche proprie che comportano numerose sfide a livello grafico);

[16] https://pymupdf.readthedocs.io/en/latest/recipes-images.html.
[17] La presente analisi si concentrerà principalmente sull'arabo in quanto alfabeto d'origine e in comune con persiano e turco azero iraniano. Sporadici esempi in persiano si troveranno nelle tabelle.
[18] Font del progetto Tasmeem costruito con le specifiche ACE (Arabic Calligraphic Engine) di DecoType. Emiri è una ricostruzione dello stile tipografico dell'edizione del Corano del Cairo del 1923 (Milo 2006).
[19] Per un'analisi complessiva degli stili calligrafici arabi vedano gli studi di al-Ǧubūrī (1998, p. 203-222) e Schimmel — Rivolta 1992 (p. 32–33).

– presenza di segni diacritici di vocalizzazione (sopra o sotto il rigo);
– presenza di decorazioni estranee al carattere che generano "rumore" confondendosi talvolta con vocalizzazione o con gli stessi punti diacritici.

Variabili legate ai sistemi OCR open source disponibili sono invece:
– qualità di riconoscimento complessiva;
– qualità di riconoscimento di metadati sintattici (ad esempio la posizione del testo nel *layout* della pagina);
– grado di addestramento del sistema sull'alfabeto arabo;
– robustezza alle variazioni di dimensione e di font.

Per quanto riguarda le variabili "esterne", le più rilevanti sono:
– qualità dell'immagine complessiva;
– risoluzione del carattere;
– variazioni del carattere in stile, grandezza e forma;
– colori dello sfondo e dei dati presenti sulla copertina (qualora non sia presente il frontespizio);
– presenza di alterazioni sulla pagina del testo (macchie, timbri ecc.)

Emergono poi ulteriori variabili "strutturali" non collegate al riconoscimento OCR ma legate a questioni più specificatamente biblioteconomiche, di struttura del libro, che risultano rilevanti in un progetto come DM, quali: struttura del frontespizio, scelte artistiche nella resa degli elementi (titolo, autore, editore) e posizione del testo nel *layout*. Si tratta di variabili che riguardano l'analisi dell'impaginazione, il riconoscimento delle porzioni testuali e dei metadati catalografici.
Le sezioni successive introducono schemi di traslitterazione, concetti di similarità e omografia, studi sulla valutazione OCR e analisi degli "errori" per categorie, inclusi esempi di scrittura coranica.

## Studi sulla valutazione di sistemi OCR: caratteristiche principali e diversità con la presente analisi

La valutazione OCR può essere condotta in modalità *blackbox* o *whitebox*: la prima considera il sistema come un tutt'uno, confrontando *output* e *ground truth* (cioè fra risultato del riconoscimento e dati veri e verificati[20]), la seconda invece analizza le prestazioni dei singoli moduli. Secondo diversi studi, le metriche possono variare per livello di dettaglio: accuratezza di grafemi e parole, precisione, esattezza su stringhe e classi di grafemi[21]. Per l'arabo, sono state proposte classificazioni basate su posizione, punti diacritici, grafemi con hamza (o "hamzati" a cui si aggiunge il grafema

[20] Kanungo — Marton — Bulbul 1999.
[21] Saber — Ahmed — Hadhoud 2014; Batawi — Abulnaja 2012; Rice — Kanai — Nartker 1993.

ك <*k*>)[22], rapporto col rigo e presenza di tratti chiusi come nel caso della classe *loop*, la quale raccoglie tutti i grafemi che presentano un tratto chiuso circolare o simile (es. ط و ع ﻫـ )[23].

Le metriche più comuni e generalmente diffuse sono Character Error Rate (CER) e Word Error Rate (WER): il primo misura i caratteri errati rispetto al totale, il secondo prende in considerazione le parole. Entrambi si basano sulla distanza di Levenshtein, cioè il numero minimo di operazioni base (sostituzione, cancellazione e inserimento) necessarie per correggere l'OCR, e di conseguenza allineare il testo estratto alla *ground truth*[24].

I sistemi presi in considerazione nella presente analisi sono stati valutati sia con le metriche note (CER e WER), sia con metriche *ad hoc* (qdiff, qscore) introdotte per valutare la qualità dei risultati in relazione a input e *ground truth* del set di dati preso in considerazione.

Dal momento che i parametri valutativi generalmente utilizzati per i sistemi OCR (tra cui l'accuratezza) poco si confacevano con le variabili presenti nei campioni a disposizione, sono state ideate due metriche distinte dipendenti dalla qualità del documento e basate principalmente sul riscontro linguistico:

– qdiff (range [-2,2]), relazione di qualità dei risultati rispetto alla *ground truth*;
– qscore (range [1,5]) qualità del risultato di riconoscimento rispetto a qualità di input.

Con le seguenti formule:

– qscore = $(5 - ((2 - oq) * (iq + 1))$ se $oq \neq 0{,}1$ altrimenti,
– qdiff = $oq - iq$

La metrica qdiff penalizza i risultati OCR quando la qualità di *input* supera quella di *output*. I test evidenziano che Tesseract ed EasyOCR estraggono testi di qualità media fornendo metadati di posizione, mentre Google Docs offre una qualità superiore e identificazione linguistica ma senza metadati sulla posizione del testo[25].

I test hanno evidenziato diversi punti di forza (Tesseract, EasyOCR: estrazione media e metadati di layout; Google Docs: riconoscimento linguistico e qualità più alta) e alcuni limiti (assenza di metadati di posizione). Gli esperimenti condotti tramite le metriche sopra descritte hanno fornito indicazioni sull'implementazione delle librerie OCR *off-the-shelf* (cioè pronte all'uso) testate per l'impiego nel contesto catalografico, senza però approfondire l'aspetto grafico e linguistico. Per questa ragione il presente studio si soffermerà proprio sull'analisi grafica e sulle implicazioni linguistiche derivanti dal riconoscimento del testo. L'esito del processo è infatti da considerarsi, in questo contesto, come una parte di un più ampio sforzo finalizzato allo sviluppo di uno strumento di catalogazione semi-automatica per una biblioteca digitale.

[22] Saber et al. 2016, p. 453.
[23] Nel presente studio tale tratto verrà chiamato: "occhiello".
[24] Alghamdi — Alkhazi — Teahan 2016, p. 2.
[25] Per i risultati numerici e statistici dei test condotti in DM si rimanda a Bergamaschi et al. 2022, p. 12–13.

## Analisi degli errori e loro categorizzazione. Scambio di forme con alto grado di similarità

| Cout per Cinp | Cout | Cinp |
|---|---|---|
| *d* per *r* | د | ر |
| *ḏ* per *z* | ذ | ز |
| *r* per *w/ū* | ر | و |
| *alif* per *l* (più raramente) | ا | ل |

Questo errore nell'individuare le forme dei caratteri non include l'elisione di punti diacritici, a meno che non intervengano ulteriori decorazioni esterne alla parola o deterioramenti dell'immagine. Il riconoscimento erroneo delle forme ha effetto sulla composizione del grafema; come anticipato, un grafema come د <*d*> rappresentato con la sua base sotto il rigo potrebbe condurre al riconoscimento di un ر <*r*> e, viceversa, <r> in cui la parte finale si alzi sopra al rigo potrebbe generare una confusione grafica per un sistema OCR. Lo stesso può avvenire tra <*ḏ*> e <*z*>. L'ultimo esempio riportato nella tabella (*alif* per *lām*) è assai meno frequente e spesso condizionato da variabili "esterne" che possono essere di diversa natura. Nel caso particolare riportato in Fig. 1 la mancanza di inchiostro nella legatura di *lām* sulla matrice di stampa comporta il riconoscimento del grafema *lām* come *alif*.

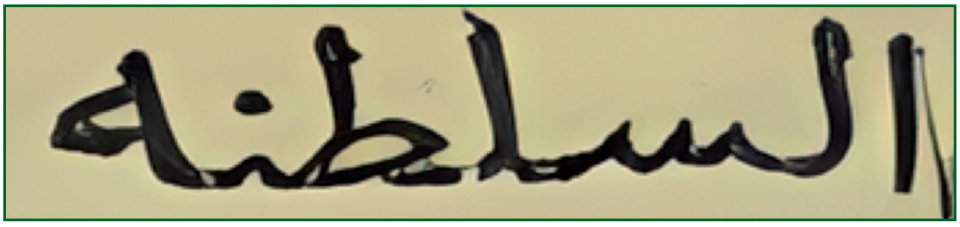

Figura 1. *Esempio di distanziamento tra i caratteri tra lām e carattere successivo nella parola "al-sulṭana"*

## Scambio di forme non simili (con e senza diacritici)

| Cout per Cinp | Cout | Cinp |
|---|---|---|
| *h* per *m* (in posizione iniziale) | هـ | مـ |
| *m* per *h* (in posizione iniziale) | مـ | هـ |
| *k* per *h* (solo in alcuni casi, si veda sotto) | كـ | هـ |
| *ṯ* per *š* (non in posizione finale, più raramente in posizione iniziale) | ث | ش |

Se si prendono come esempio i grafemi ث e ش, può accadere che assumano forme simili quando i denti tipici di <*š*> sono appiattiti sul rigo (in stili come *ruqʿa, taʿlīq e nastaʿlīq*). L'errore di scambio di forma è soggetto quindi alle variabili definite qui come "interne" e proprie della calligrafia araba. L'esempio di <*h*> per <*k*> sembra pertanto graficamente il più indicativo per la suddivisione qui proposta. Lo scambio di <h> per <k> si può riscontrare soprattutto quando la ك <*k*> espressa in stili quali *ruqʿa* e *nastaʿlīq* nella posizione iniziale e mediana vede la sua parte più prossima al rigo arrotondarsi a formare un occhiello. Quando rappresentata in questa forma può essere erroneamente riconosciuta come هـ <*h*> (si veda Fig. 2).
Le variabili "interne" allo stile calligrafico sono anche responsabili della fusione dei punti diacritici; due punti diacritici allineati possono infatti essere orientati in modo diverso e anche fusi insieme in un unico tratto parallelo al rigo. In casi particolari i punti diacritici possono essere disposti anche in senso verticale, aumentando la possibilità di errore. Caratteri calligrafici come il *ruqʿa* esprimono i due punti, sopra o sotto al rigo, come un trattino orizzontale e i tre punti come una sorta di accento circonflesso.

Figura 2. *Esempio dello stile ruqʿa con punti diacritici uniti in trattino nella parola "al-muškilāt"*

## Rimozione totale o parziale dei punti diacritici

| **Cout per Cinp** | **Cout** | **Cinp** |
|---|---|---|
| *n* per *t* (non in posizione finale) | ن | ت |
| *t* per *ṯ* | ت | ث |
| *ḥ* per *ḫ* | ح | خ |
| *f* per *q* | ف | ق |

La rimozione dei punti diacritici può essere totale o parziale, causando convergenza verso un grafema diverso (خ <*ḫ*> in ح <*ḥ*>) o forme non grammaticali. Si prenda ad esempio in considerazione il termine *taḫayyur* **تحير** («scegliere, optare») dalla radice خ-ي-ر. In questa forma la parola, priva di vocalizzazione e contesto, può rappresentare sei parole distinte (non omofone), di cui cinque verbi (tre con diatesi attiva e due con diatesi passiva) e un sostantivo. Al contempo, la parola può appartenere a classi grammaticali diverse e avere significati diversi pur rimanendo graficamente invariata. Ponendo ora che il riconoscimento OCR abbia erroneamente eliso il punto diacritico del grafema خ generando ciò che segue: **تحير** ("confusione, l'essere confuso"), è dunque evidente

che un errore di riconoscimento può avere conseguenze non limitate al contesto grafico ma anche sul piano semantico, sebbene riconosciuta da una risorsa linguistica come parola valida in un'ontologia o un modello linguistico. L'esempio vuole rimarcare le dimensioni del problema, che rappresenta un concreto ostacolo all'applicazione a posteriori di strumenti di NLP (come ad esempio uno *stemmer*[26]), comportando il rischio di propagare l'errore nelle fasi successive. Dunque l'errore inizialmente grafico genera conseguenze semantiche di non poco conto.

Nel contesto dell'ideazione e progettazione di un flusso di lavoro catalografico, errori sui diacritici dei nomi d'autore possono generare forme ambigue, aumentando i tempi della fase di verifica e controllo catalografico. La precisione nell'estrazione dati dai frontespizi è cruciale per evitare tali problematiche.

A tal proposito un esempio è proposto da Sublet[27] nel contesto degli studi onomastici arabi in riferimento all'opera di al-Ḏahabī dal titolo *al*-Muštabih fī al-riǧāl[28], nel quale sono riportati i possibili esiti di lettura di elementi onomastici omografi[29] con variazioni di punti e segni diacritici: الجُندي، الجَندي or الحيدي . Al-Ḏahabī specifica anche nel dettaglio lo scambio di forme simili riportando di nuovo gli esiti di lettura possibili, come ad esempio: الحيري الجَنْزي، الخَبْري، الخَبَري، الجبْري، الحُتْري، الجَبْري،.

Gli esempi riguardanti ه/ة e ي/ى, non riportati nella tabella per brevità, sono ben noti nella linguistica computazionale e negli studi di ANLP (Arabic Natural Language Processing), in particolare nell'ambito dello sviluppo di sistemi di segmentazione e di NLP per la lingua araba. Proponendo un segmentatore per l'arabo classico, Mohamed e Sayyed[30] categorizzano questi due gruppi come "t/h confusion set" e "y confusion set" da considerarsi a fianco del trattamento della *alif-hamza*, "the hamza confusion set", nel contesto dell'ortografia araba standard o sub-standard[31].

## Aggiunta di punti diacritici

In maniera inversa alla rimozione dei punti diacritici, alcune variabili "esterne" (decorazioni sul testo, deterioramento dell'immagine o della stampa), come "interne" ("rumore" generato da compresenza di vocalizzazione e punti diacritici), possono concorrere, nell'output, all'aggiunta di punti diacritici ove non dovrebbero essere presenti. L'"errore" in questione è diametralmente opposto al precedente ma ne condivide numerose caratteristiche con conseguenze simili a quelle descritte nel punto precedente. Per questa ragione non verranno trattate nello specifico.

---

[26] Gli algoritmi di normalizzazione post-processing rimuovono diacritici, articoli, prefissi e suffissi. Tra i principali *stemmer*: Buckwalter (2002), Tashaphyne Light Stemmer, CAMeL lab (Obeid et al. 2020)

[27] Sublet 1999, p. 125.

[28] Ḏahabī 1962.

[29] In questo caso la *nisba*, un elemento onomastico costituito dall'aggiunta di un suffisso derivativo *-yā' (o -yā' nisbiyya* ـيّ) a un sostantivo. In alcuni casi, la *-yā*› può essere preceduta (per ragioni fonetiche) dal nesso *alif-nūn* ( انـي = ي + ان ) o succeduto da *tā' marbūṭa* ( ـيّة ) qualora femminile.

[30] Mohamed — Sayyed 2019, p. 4-5.

[31] Data il numero consistente di rappresentazioni dei grafemi arabi e persiani, anche in ambito computazionale sono numerosi gli studi sull'ortografia. Si veda: (Habash 2010, p. 31-34; Habash — Diab — Rambow 2012, p. 711-718).

## Scivolamento dei punti diacritici

| Cout | Cinp |
|---|---|
| نقتيش | تقتيش |
| محيور | مجبور |
| ثحميس | تخميس |
| ثناقد | تناقد |

Uno dei più comuni "errori" è quello generato dallo spostamento dei diacritici, collegato alle variabili "interne" dello stile calligrafico. Questo tipo di "errore" è soggetto anche al grado di addestramento e taratura del sistema OCR sulla distinzione dei caratteri e dei collegamenti fra di essi. La sovrapposizione dei caratteri attraverso specifiche legature sposta i punti diacritici in posizioni che possono risultare ambigue (Fig. 3) per un sistema non sufficientemente addestrato. La posizione e la forma assunta dai grafemi dovuti al tipo di legatura può inoltre prevedere un posizionamento non necessariamente parallelo al rigo.


Figura 3. *Esempio di legatura per la parola "naǧm": a destra con legatura distesa, a sinistra contratta*

Se si prende in considerazione lo stile calligrafico *nastaʿlīq* nella sua varietà *šekasteh*[32] (pers.: "rotto, spezzato"), l'orientamento del rigo, inclinato a 45 gradi, in aggiunta a un set di legature specifiche, rende il riconoscimento arduo per sistemi OCR non sufficientemente sviluppati. Un esempio concreto è fornito in Fig. 4, in corrispondenza del nesso consonantico <*n-s-t*> ( نستـــ ), dove il punto diacritico relativo a ن <*n*> è slittato verso il centro della stringa, i tre "denti" di س <*s*> sono appiattiti sul rigo mentre i due punti di ت <*t*> sono spostati verso l'inizio della stringa. Dal punto di vista delle conseguenze sul riconoscimento si può notare che lo spostamento dei diacritici può comportare l'aggiunta o la sottrazione dei segni diacritici, con esiti del tutto assimilabili a quelli già descritti.

[32] Nella variante *nastaʿlīq-šekasteh*, ogni elemento influenza il successivo trovando un'unità grafica nel suo risultato finale e sviluppandosi in modo inclinato sul rigo. Grafemi, parole e legature sono continue e formano l'una la base per la successiva, seguendo una forma ellittica (Maʿnavī Rād 2013, p. 23).

Figura 4. *Cerchiato in rosso si trova il nesso consonantico n-s-t rappresentato con lo stile šekasteh per la parola "nastaʿlīq"*

## Interferenza rumorosa tra segni e punti diacritici

| **Cout per Cinp** | **Cout** | **Cinp** |
|---|---|---|
| *ġ* per *ʿu* | غنير | عُنير |
| *t* per *nu* | تور | نُور |
| *z* per *ra* | ضَنزائب | ائبضَنَر |
| [33]*ġ* per *ʿ (sukūn)* | فغل | فعْل |

Nel corso dei test si sono potute rilevare interferenze tra vocali e punti diacritici, risultanti in un riconoscimento erroneo di punti diacritici. Tuttavia, se è vero che le vocali possono essere rese erroneamente come punti diacritici, non è altrettanto vero il contrario. Infatti, come già rilevato, generalmente i punti diacritici posso subire trattamenti scorretti, come già indicato in precedenza, possono essere rimossi, aggiunti, dimenticati o traslati orizzontalmente, o in alcuni casi verticalmente. Quando presenti, le vocali possono interferire con i punti diacritici, causandone spesso l'aggiunta dove non sono previsti. Questa interferenza ha come conseguenza lo scambio delle consonanti, generando forme non grammaticali o lemmi di significato diverso da quello della parola in input.

## Riconoscimento dovuto a dimensioni dei caratteri

Talvolta la variazione della dimensione del carattere può essere decisiva nel determinare la buona riuscita di un riconoscimento, concorrendo con altre variabili sia "interne" che "esterne" a restituire un esito "errato". Poniamo qui l'esempio di due parole con il medesimo schema morfologico *fiʿāla* (plur. *fiʿālāt*), dal momento che le vocali lunghe sono due *alif* generalmente estratte correttamente. Ciò che si modifica sono le consonanti che la compongono ma con inaspettate "evoluzioni" grafiche:

[33] Segno diacritico apposto su consonanti per indicare l'assenza di vocale in lettura, *sukūn* "quiete, assenza di movimento, tranquillità"

| Cout per Cinp | Cout | Cinp |
|---|---|---|
| | صراعات | دراسات |
| ṣ per *d* | صـ | د |
| ʿ per *s* | عـ | سـ |

Il primo termine a destra, *dirāsāt* ("studi"), rappresenta *l'input* mentre il termine a sinistra, *ṣirāʿāt* ("lotte, conflitti"), la stringa effettivamente generata dall'OCR. Se si analizzano attentamente i caratteri "errati", ad esempio nella riga ع <ʿ> - س <s>, l'"errore" appare poco comprensibile. I due grafemi hanno due forme sopra il rigo con un basso grado di similarità. Quando l'*occhiello* concavo con apertura a destra tipico della <ʿ> (عـ) si trova sopra al rigo, differisce molto dal punto di vista grafico rispetto ai tre *denti* della <s> (سـ). Nonostante le palesi differenze formali, però, i due caratteri vengono confusi. Questo esito suggerisce alcune interessanti osservazioni: da un lato si può notare come la dimensione influenzi il riconoscimento anche in casi macroscopici, qualora il sistema OCR non sia in grado di garantire una adeguata robustezza (variabile "interna"); dall'altra la dimensione ridotta comporta una compressione dei grafemi e un aumento della probabilità che si verifichi uno scambio di forma soprattutto in prossimità del rigo[34].

## Risultati derivanti da trattamento di *alif* e *hamza*

| Cout per Cinp | Cout | Cinp |
|---|---|---|
| *alif* per *alif-hamza* | اقول | أقول |
| | سال | سأل |
| *alif maqṣūra* per *alif-hamza* (in contesto di vocale *i)* | قرى | قرئ |
| punto diacritco per *hamza* (e viceversa, sotto) | سذال | سؤال |

Il trattamento del nesso *alif-hamza* rappresenta una delle difficoltà che la grafia araba pone, non solo per il riconoscimento ottico dei caratteri ma anche per le tecniche di analisi testuale (analizzatori morfologici, *stemmer*, *tagger* ecc.) prodotte nel campo dell'ANLP[35]. A tal proposito si prenda in considerazione il quarto esempio سـأل/سال , dove la lettura della *hamza* è fondamentale per distinguere il verbo *saʾalᵃ* "chiedere, domandare" (media *hamza*) e il verbo *sālᵃ* "scorrere, fluire, divenire/essere liquido". In questo caso la ء distingue due radici afferenti a due campi semantici ben distinti: *s-ʾ-l* e *s-y-l*. Analogo è anche l'esempio del trattamento della ء sostenuto da *alif maqṣūra*

[34] I test condotti hanno sottolineato che con immagini di buona qualità, anche con caratteri di piccola dimensione, i grafemi sopra il rigo (alif ا) e isolati (come t in -āt ات del plurale femminile) vengono riconosciuti correttamente.

[35] Il trattamento della *hamza* e delle varianti della *alif* è stato affrontato in ambito ANLP (morfologia, stemming, tagging, tokenizzazione, lemmatizzazione) e, negli studi OCR, dal punto di vista della segmentazione grafica dei caratteri. Si veda ad esempio (Qaroush et al. 2020, p. 2-5).

(ى) qui riportato nella I forma del verbo passivo III p.s. della radice <*q-r-ʾ*> (“leggere, recitare, salmodiare, studiare”). Nuovamente, la forma passiva قرئ *quriʾa* si distingue dal plurale fratto قرى *qurā* (“villaggi”) solamente per la presenza della *hamza*. Per questo motivo non si può prescindere, dal punto di vista dell’analisi della semantica, da risorse linguistiche capaci di disambiguare le parole.

## Mancata lettura degli spazi

Il rilevamento degli spazi è uno dei temi più studiati nel campo dell’ANLP, in particolare in relazione alla normalizzazione e alla segmentazione del testo. In quest’analisi, tuttavia, il testo è considerato come unità da riportare semplicemente nell’output in maniera più fedele possibile a ciò che si trova sul foglio oggetto dell’analisi. Qui di seguito si riportano due esempi:

| Cout | Cinp |
|---|---|
| أحمد بنعلي | أحمد بن علي |
| وَرَبَّه | وَرَبُّه |

La mancata lettura degli spazi può generare forme inesistenti o semplicemente diverse dall’input in maniera del tutto simile alle ambiguità generate con sottrazione, addizione e scivolamento di diacritici. Nel secondo esempio, la *wa* di congiunzione in arabo è prefissa alla parola che segue e nel primo nesso a destra si ha *wa-rabbᵘhᵘ* (“e il suo signore, padrone”) mentre nel nesso più a sinistra si può considerare *wa* come parte della radice *w-r-b* “essere marcio, marcire” alla II forma *warrabᵃ-yuwarribᵘ* coniugata alla III p.s.m passato con suffisso *-hu* (“egli, lui”), quindi *warraba-hu*: letteralmente “lo espresse in maniera contraddittoria”. Si tratta di un esempio limite che esemplifica chiaramente il modo in cui aspetti grafici si intersecano con quelli semantici e morfologico-grammaticali.

## Risultati con “errori” multipli (o composti)

Un’ulteriore categoria è rappresentata dagli errori multipli (o composti), ovvero di diversa tipologia occorrenti nel risultato di riconoscimento della medesima stringa di grafemi o di testo. Qui di seguito alcuni esempi esplicativi di questa categoria:

| Cout | Cinp |
|---|---|
| مشود نزويگ | ميشود نزديک |
| تژس ولدذ | ترس ولرز |
| النبوي صحعحلفمهر | النبوي القصص صحيح |

Anche in questo caso, bisogna sempre tenere in considerazione la presenza di una commistione di variabili “interne” ed “esterne” che concorrono nel generare errori. Sono possibili combinazioni quali: scambio di forme, rimozione e addizione di diacritici, mancata lettura di spazi e interferenze tra vocalizzazione e punti diacritici. Estendendo le stringhe testuali e passando da unità minime (glifi e grafemi) a unità testuali più com-

plesse (parole, frasi, paragrafi ecc.) diviene più facile trovare una quantità maggiore di errori e, in caso di OCR non sufficientemente addestrati, errori multipli di crescente complessità fino a casi di stringhe completamente irriconoscibili.

## Segni di lettura e diacritici nel Corano

Per quanto concerne il Corano, sono presenti online numerosi strumenti per l'analisi del testo, come anche versioni già digitalizzate e navigabili[36]. Tuttavia, nel caso di altri testi non digitalizzati in cui si trovano riferimenti diretti ai versetti coranici (*tafsīr*, *fiqh*, raccolte di *ḥadīṯ* ecc.) le stringhe di testo possono presentare una serie supplementare di segni diacritici sopra i grafemi, per segnalarne la corretta recitazione. In questo caso nel testo non vocalizzato sono inseriti i versetti coranici a cui si fa riferimento completi di vocalizzazione e segni diacritici come nella Fig. 5.

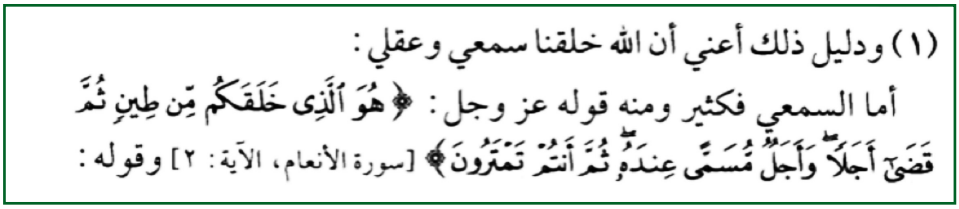
(١) ودليل ذلك أعني أن الله خلقنا سمعي وعقلي:
أما السمعي فكثير ومنه قوله عز وجل: ﴿هُوَ ٱلَّذِى خَلَقَكُم مِّن طِينٍ ثُمَّ
قَضَىٰٓ أَجَلًا ۖ وَأَجَلٌ مُّسَمًّى عِندَهُۥ ثُمَّ أَنتُمْ تَمْتَرُونَ﴾ [سورة الأنعام، الآية: ٢] وقوله:

Figura 5. *Frammento di testo da al-ʿUṯaymīn, M. 2004. Šarḥ ṯalāṯ al-ʾuṣūl, Riyāḍ, dār al-ṯurayyā li-l-našr, p. 29*

Questi particolari segni diacritici sono l'espressione grafica delle minuziose regole di dizione e lettura coranica che costituiscono la scienza islamica del *taǧwīd*[37]. I segni diacritici relativi alle vocali, definiti nel *taǧwīd* come segni di "precisazione" (*ʿalāmāt al-ẓabṭ*)[38] possono eventualmente aumentare la quantità di "rumore" sul testo. A titolo di esempio si vogliono ricordare: la piccola *m* sovrascritta in contesto di *n-b* o *m-b* (*iqlāb*); il simbolo che indica la presenza di *imāla*[39] o di *išmām*[40], che consiste in un circoletto o rombo pieno posizionato sotto ◌ o sopra il rigo ◌ (*dāʾira maṭmūsa*) rispettivamente, es. Cor. XI:41 مَجْرٖىٰهَا (*maǧrā-hā*) e in Cor. XII:11 تَأْمَنَّا (*taʾmannā*); o ancora i segni di allungamento (*madd*) — come quello riscontrabile sulla *alif-madda* anche al di fuori del testo coranico. La *alif* sovrascritta (detta anche *alif ḫanǧariyya*) stante alla presenza di una *alif* di prolungamento non graficamente espressa nella sua forma consueta, come in Q. CXIII:4 ( نفٰثٰت ).

[36] Si veda ad esempio il progetto open source The Quranic Arabic Corpus: <https://corpus.quran.com/> e il Tanzil Project: <https://tanzil.net/>.

[37] *taǧwīd* significa letteralmente: "articolare i fonemi nel loro punto appropriato rispettando tutte le caratteristiche" (Zalaṭ bin Raf'at 2006, p. 5).

[38] Šukrī 2014, p. 1496.

[39] *'imāla* (letteralmente "l'inclinazione, il tendere verso qc."), in fonetica araba è usato ad indicare l'innalzamento della vocale a verso la vocale *i* causata dall'influenza della *kasra* sulla *alif* stante per vocale a. Tale fenomeno era già stato osservata dal famoso Sībawayhi (1988, p. 259-261).

[40] Con *išmām* si intende il posizionamento delle labbra per produrre una vocale *u, ḍamma* (o vocale i, *kasra*) ma senza emissione di suono effettivo (Naṣr 2009, p. 615).

## Conclusioni e prospettive di ricerca

Questo studio vuole mettere in evidenza alcuni dei comportamenti comuni dei sistemi OCR open-source testati su frontespizi arabi, analizzando le relazioni tra questi sistemi e le peculiarità dell'alfabeto arabo nel contesto di in un possibile flusso di lavoro di catalogazione automatizzata. Il riconoscimento ottico dell'arabo rimane un campo in parte inesplorato, sia dal punto di vista teorico che pratico. Gli "errori" identificati possono essere causati da fattori intrinseci all'alfabeto o "esterni", che si riducono spesso alla rimozione o aggiunta di punti diacritici. I risultati teorici trovano riscontro in ricerche analoghe, come l'analisi di Kiessling su KrakenOCR[41], che evidenzia problemi simili: lettura degli spazi, difficoltà con diacritici, trattamento del nesso *alif-hamza* e riconoscimento di consonanti specifiche[42].

Uno dei problemi principali dell'AOCR (OCR di caratteri arabi) è la scarsità di set di dati disponibili rispetto ad altri alfabeti[43], in particolare per quanto riguarda i testi definiti nel contesto OCR come "historical". Nonostante la crescita di database annotati, mancano ancora *corpora* storici di alta qualità per sistemi multilingua. Nel caso specifico del progetto Digital Maktaba si registra invece la necessità, a cui si sta lavorando attualmente, di creare un set di dati specifico per frontespizi a stampa.

Un altro importante tema affrontato nel corso di questo studio riguarda la codifica Unicode e, in generale, la mappatura dei caratteri. La combinazione OCR-Unicode è fondamentale per risultati ottimali. Lo standard Unicode presenta ambiguità di codificazione e limitatezze per gli stili calligrafici meno rappresentati. Ad esempio, il carattere ى ha due codici (U0649 arabo, U06CC persiano) per la stessa rappresentazione grafica. Queste ambiguità, dovute all'omografia araba, hanno ripercussioni sull'individuazione di informazioni ai fini catalografici. I numerali presentano set differenti: arabo (0660-0669) e persiano/altre lingue (06F0-06F9), tema finora poco trattato ma rilevante per il recupero dati catalografici.

Interessanti studi come quello di Milo e Martínez[44] hanno introdotto il concetto di *arcigrafema*, evidenziando la necessità di un addestramento linguistico per OCR dinamici con intelligenze artificiali. Allo stesso modo considerazioni sull'ambiguità della scrittura araba come un "meccanismo di tolleranza", evidente nei manoscritti coranici con doppi marcatori diacritici[45] sono fondamentali per avanzare il dibattito anche nel campo OCR all'interno del contesto condiviso delle *Digital Humanities*.

Al contempo non si devono sottovalutare le possibilità tecniche derivanti dalle possibilità offerte dai *Transformers* (che elaborano sequenze di dati tramite un meccanismo che valuta l'importanza relativa di ogni elemento rispetto agli altri) e dai recenti sviluppi nel campo dei modelli linguistici. Lo sviluppo di algoritmi *Deep Learning* (reti neurali artificiali multistrato) ha rinnovato l'interesse per le lingue araba, persiana, curda e

[41] Romanov et al. 2017.
[42] Kiessling et al. 2021, p. 10-11.
[43] Wagaa — Kallel — Mellouli 2022, p. 1.
[44] Milo — Martínez 2019.
[45] Fedeli 2020.

urdu. I modelli di distribuzione basati su *Vector Semantics* (semantica distribuzionale[46]) potrebbero migliorare l'analisi contestuale tramite disambiguazione semantica[47].
Modelli come BERT, AraBERT e LLMs (come Mistral, LaMa, o GPT) offrono prospettive per il NLP arabo attraverso *word embeddings* (rappresentazione distribuita delle parole) e previsione contestuale nella fase di post-elaborazione. L'integrazione fra OCR e LLM rimane però poco esplorata per la scrittura araba, nonostante applicazioni promettenti nell'estrazione di informazioni[48].
Infine, risultano di un certo interesse per future prospettive di ricerca le analogie riscontrate con alcuni studi cognitivi sulla lettura: i sistemi OCR potrebbero infatti essere paragonati al processo di apprendimento della lettura infantile, sostituendo conoscenze pregresse con risorse linguistiche. Gli studi di Azzam e Abu-Rabia per l'arabo[49] e Baluch per il persiano[50] potrebbero fornire spunti preziosi per l'analisi degli errori OCR confrontandoli con quelli dei giovani lettori.

*In digitisation in the context of document preservation in large archives, OCR systems have become essential tools as a compromise between good character recognition and low cost. However, they are still lacking in training on historical and religious texts and in particular with non-Latin alphabets. Large amounts of unstructured data characterised by scatter and noise expose the limitations of text mining techniques on several levels. These limitations add up to contextual variables hindering OCR systems from achieving optimal character recognition. This represents a real problem when considering such systems as central to the development of applications involving the downstream use of other NLP techniques. From this point of view, an error analysis is part of a post-processing phase that can have corrective effects on the output in order to improve the recognition result especially when combined with a context analysis. Greater attention to post-processing on both glyphs and graphemes could bring about a considerable improvement in OCR effectiveness by significantly advancing the current state of the art. The brief study presented here aims to outline common traits of errors committed by such OCRs tested in the Digital Maktaba project.*

[46] Sadrzadeh — Muskens 2018.
[47] Gavin 2018, p. 660-663.
[48] Tang et al. 2024; Xu et al. 2023.
[49] Azzam 1989, Abu-Rabia 1998.
[50] Baluch 2005.

L'ultima consultazione dei siti web è avvenuta nel mese di dicembre 2025.

# Riferimenti bibliografici


Abu-Rabia 1998 — Salim Abu-Rabia. *Reading Arabic Texts: Effects of Text Type, Reader Type and Vowelization*. «Reading and Writing», 10 (1998), n. 2, p. 105–119. <https://doi.org/10.1023/A:1007906222227>.

Afli — Loïc —Schwenk 2016 — Haithem Afli — Barrault Loïc — Holger Schwenk. *OCR Error Correction Using Statistical Machine Translation*. «International Journal of Computational Linguistics and Applications», 7 (2016), n. 1, p. 175–191.

Ahmed — Abidi 2019 — Muna Ahmed — Ali Abidi. *Review on Optical Character Recognition*. «International Research Journal of Engineering and Technology (IRJET)», 6 (2019), n. 6, p. 3666–3669.

Alaasam — Barakat — Kassis — El-Sana 2017 — Reem Alaasam — Berat Barakat — Majeed Kassis — Jihad El-Sana. *Experiment study on utilizing convolutional neural networks to recognize historical Arabic handwritten text.* In: *1st International Workshop on Arabic Script Analysis and Recognition* (ASAR), 2017, p. 124–128. <https://doi.org/10.1109/ASAR.2017.8067773>.

Alghamdi — Alkhazi — Teahan 2016 — Mansoor A. Alghamdi — Ibrahim S. Alkhazi — William J. Teahan. *Arabic OCR evaluation tool.* In: *2016 7th International Conference on Computer Science and Information Technology* (CSIT), 2016, p. 1–6. <https://doi.org/10.1109/CSIT.2016.7549460>.

Azzam 1989 — Rima Azzam. *Orthography and Reading of the Arabic Language.* In: *Reading and Writing Disorders in Different Orthographic Systems*, ed. by P. G. Aaron, R. Malatesha Joshi (NATO ASI Series; 52). Dordrecht: Springer Netherlands, 1989, p. 203–218. <https://doi.org/10.1007/978-94-009-1041-6_12>.

Baluch 2005 — Bahman Baluch. *Persian Orthography and Its Relation to Literacy.* In: *Handbook of Orthography and Literacy*, ed. by R. Malatesha Joshi, P. G. Aaron. London: Lawrence Erlbaum Associates, 2005, p. 365-376. <https://eprints.mdx.ac.uk/7649/>.

Bassil — Alwani 2012 — Youssef Bassil — Mohammad Alwani. *OCR Post-Processing Error Correction Algorithm using Google Online Spelling Suggestion*. 2012. <https://arxiv.org/abs/1204.0191>

Batawi — Abulnaja 2012 — Y. Batawi — Osama Abulnaja. *Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique: Experimental Study*. «International Journal of Electrical & Computer Sciences IJECS-IJENS», 12 (febbraio 2012), p. 29–33.

Bergamaschiet et al. 2022 — Sonia Bergamaschi — Stefania De Nardis — Riccardo Martoglia — Federico Ruozzi — Luca Sala — Matteo Vanzini — Riccardo Amerigo Vigliermo. *Novel Perspectives for the Management of Multilingual and Multialphabetic Heritages through Automatic Knowledge Extraction: The DigitalMaktaba Approach*. «Sensors», 22 (2022), n. 11, 3995. <https://doi.org/10.3390/s22113995>.

Boros et al. 2024 — Emanuela Boros — Maud Ehrmann —Matteo Romanello — Sven Najem-Meyer — Frédéric Kaplan. *Post-correction of Historical Text Transcripts with Large Language Models: An Exploratory Study*. In: *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*. St Julian (Malta): The Association for Computational Linguistics, 2024, p. 133-159.

Buckwalter 2002 — Tim Buckwalter. *Buckwalter Arabic Morphological Analyzer Version 1.0*. Linguistic Data Consortium, 2002. <https://doi.org/10.35111/7VZM-MB15>.

Buziyād 2011 — Sāmiya Buziyād. *mağallat al-dirāsāt al-luġawiyya*. «ẓāhirat al-rawm wa-l-'išmām bayna al-nuḥḥāt wa-l-qurrā': dirāsa ṣawṭiyya», 13 (2011) p. 65–84.

Chiron eet al. 2017 — Guillaume Chiron — Antoine Doucet — Mickael Coustaty — Muriel Visani — Jean-Philippe Moreux. *Impact of OCR Errors on the Use of Digital Libraries: Towards a Better Access to Information*. In: *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2017, p. 1–4. <https://doi.org/10.1109/JCDL.2017.7991582>.

Ḏahabī 1962 — Šams al-dīn al-Ḏahabī. *al-Muštabih fī al-riǧāl: asmā'i-him wa ansābi-him*, a cura di 'Alī Muḥammad al-Bağāwī. Bayrūt: Dār iḥiyā' al-kutub al-'arabiyya, 1962.

Drobac — Lindén 2020 — Senka Drobac — Krister Lindén. *Optical character recognition with neural networks and post-correction with finite state methods*. «International Journal on Document Analysis and Recognition (IJDAR)», 23 (2020), p. 279–295. <https://doi.org/10.1007/s10032-020-00359-9>.

Fedeli 2020 — Alba Fedeli. *The Qur'ānic Text from Manuscript to Digital Form: Metalinguistic Markup of Scribes and Editors*. In: *From Scrolls to Scrolling, Sacred Texts, Materiality, and Dynamic Media Cultures*., ed. by B. A. Anderson. De Gruyter, 2020, p. 213–246. <https://doi.org/10.1515/9783110634440-010>.

Garbini — Durand 1994 — Giovanni Garbini — Olivier Durand. *Introduzione alle lingue semitiche*. (Studi sul vicino Oriente antico; 2). Brescia: Paideia, 1994.

Gavin 2018 — Michael Gavin. *Vector Semantics, William Empson, and the Study of Ambiguity*. «Critical Inquiry», 44 (2018), n. 4, p. 641–673. <https://doi.org/10.1086/698174>.

Ğubūrī 1998 — Muhammad Šukr al-Ğubūrī. *al-ḫaṭṭ al-ʿarabī wa-l-zaḫrafa al-ʻislāmiyya*. Irbid, al-'Urdun: Dār al-'amal li-l-našr wa-l-tawzīʿ, 1998.

Habash 2010 — Nizar Habash. I*ntroduction to Arabic Natural Language Processin*g. Cham: Springer, 2010. <https://link.springer.com/book/10.1007/978-3-031-02139-8>.

Habash — Diab — Rambow 2012 — Nizar Habash — Mona Diab — Owen Rambow. *Conventional Orthography for Dialectal Arabic*. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012, Istanbul*). Paris: European Language Resources Association, 2012, p. 711-718.

Hajiali 2023 — Mahdi Hajiali. *OCR Post-Processing Using Large Language Models*. (Dissertation). Las Vegas: UNLV, 2023. <https://digitalscholarship.unlv.edu/thesesdissertations/4811>.

Ibn-Durustawaih et al. 1998 — Ibn-Durustawaih — Abdallah Ibn-Gafar — Muhammad Badawi al-Mahtun — Ramadan Abd-at-Tauwab. *Tashih al-Fasih wa-sarhihi*. al-Qahira: Gumhuriyat Misr al-`Arabiya, Wizarat al-Auqaf, al-Maglis al-A`la li-s-Suun al-Islamiya, 1998.

ISIRI 6219 2002 — Institute of Standards and Industrial Research of Iran. *ISIRI 6219 - FarsiWeb*. 2002. <https://persian-computing.org/archives/Sharif-FarsiWeb-Inc/ISIRI_6219.html>.

Javed — Hussain 2009 — Sobia Tariq Javed — Sarmad Hussain. *Improving Nastalique specific pre-recognition process for Urdu OCR*. In: *IEEE 13th International Multitopic Conference (Islamabad, 2009)*. Piscataway (NJ): IEEE, 2009, p. 1–6. <https://doi.org/10.1109/INMIC.2009.5383111>.

Kanungo — Marton — Bulbul 1999 — Tapas Kanungo — Gregory A. Marton — Osama Bulbul. *Performance evaluation of two Arabic OCR products*. In: *27th AIPR Workshop: Advances in Computer-Assisted Recognition (29 January 1999)*. SPIE, 1999, p. 76-83. <https://doi.org/10.1117/12.339809>.

Kew 2005 — Jhonathan Kew. *Notes on Some Unicode Arabic Characters: Recommendations for Usage*. Draft 2. 21 aprile 2005. <https://it.scribd.com/doc/235215269/Arabic-Letter-Usage-Notes>.

Khosrobeygi et al. 2020 — Zohreh Khosrobeygi — Hadi Veisi — Hamid Reza Ahmadi — Hanieh Shabanian. *A rule-based post-processing approach to improve Persian OCR performance*. «Scientia Iranica», 27 (2020), n. 6, p. 3019–3033. <https://doi.org/10.24200/sci.2020.53435.3267>.

Kiessling et al. 2021 — Benjamin Kiessling — Gennady Kurin — Matthew Thomas Miller — Kader Smail. *Advances and Limitations in Open Source Arabic-Script OCR: A Case Study*. «Digital Studies / Le Champ Numérique», 11 (2021), n. 1. <https://doi.org/10.16995/dscn.8094>.

Lībī 2017 — A. al-Mālikī al-Lībī. *al-tuḥfa al-mālikiyya fī talḫīṣ ’uṣūl riwāya ḥafṣ ʿan ʿāṣim min ṭarīq al-šāṭibiyya*. 1a ed. silsila mulaḫḫaṣāt ’uṣūl al-qirā’āt 2., 2017.

Maʿnavī Rād 2013 — Mītrā Maʿnavī Rād. *Negareh: faṣlnāmeh ʿelmī – pajūheši-ye negareh. In: taʿāmol-e sāḫtār va sabok dar šekasteh nevīsī: šafīʿān, darvīš va golestāneh*. 27 (2013), p. 21–33.

Milo 2006 — Thomas Milo. *The original Tasmeem Manual*. 2006. <https://www.academia.edu/3517400/The_original_Tasmeem_Manual_2006_>.

Milo — Martínez 2019 — Thomas Milo — Alicia González Martínez. *A New Strategy for Arabic OCR: Archigraphemes, Letter Blocks, Script Grammar, and shape synthesis*. In: *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (DATeCH2019)*. New York: Association for Computing Machinery, 2019, p. 93–96. <https://doi.org/10.1145/3322905.3322928>.

Mohamed — Sayyed 2019 — Emad Mohamed — Zeeshan Ali Sayyed. 2019. *Arabic-SOS: Segmentation, Stemming, and Orthography Standardization for Classical and pre-Modern Standard Arabic*. In: *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (DATeCH2019)*. New York: Association for Computing Machinery, 2019, p. 27–32. <https://doi.org/10.1145/3322905.3322927>.

Munağğid 1999 — Muḥammad Nūr al-dīn al-Munağğid. *al-ištirāk lafẓī fi-l-qur’ān al-karīm: bayna al-naẓariyya wa-l-taṭbīq. Dimašq: Dār al-fikr*, 1999. <https://ebook.univeyes.com/95981/pdf-الاشتراك اللفظي في القرآن الكريم بين النظرية والتطبيق>.

Naṣr 2009 — Ismāʿīl bin Ḥamād al-Ǧawharī abū Naṣr. *al-ṣaḥḥāḥ*. al-Qāhira: Dār al-ḥadīṯ, 2009.

Naṣrāwī 2015 — ʿādil ʿabbās al-Naṣrāwī. *dawāt / mağallah faṣliyyah tuʿnà bi-l-buḥūṯ wa-l-dirasāt al-luġawiyyah*. «al-manhağ al-riyāḍī fī al-dars al-muʿğamīʿind al-Farāhīdī: taqwīm wa taǧdīd», 1 (2015), n. 3, p. 75–101.

Nguyen et al. 2021 — Thi Tuyet Hai Nguyen — Adam Jatowt — Mickael Coustaty — Antoine Doucet. *Survey of Post-OCR Processing Approaches*. «ACM Computing Surveys», 54 (2021), n. 6, p. 1–37. <https://doi.org/10.1145/3453476>.

Obeid et al. 2020 — Ossama Obeid — Nasser Zalmout — Salam Khalifa — Dima Taji — Mai Oudah — Bashar Alhafni — Go Inoue — Fadhl Eryani — Alexander Erdmann — Nizar Habash. *CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing*. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille: European Language Resources Association, 2020, p. 7022–7032 <https://aclanthology.org/2020.lrec-1.868>.

Pakhale 2023 — Kalyani Pakhale. *Large Language Models and Information Retrieval*. «SSRN Scholarly Paper», 18 dicembre 2023. <https://doi.org/10.2139/ssrn.4636121>.

Qaroush et al. 2020 — Aziz Qaroush — Abdalkarim Awad — Mohammad Modallal — Malik Ziq. *Segmentation-based, omnifont printed Arabic character recognition without font identification*. «Journal of King Saud University - Computer and Information Sciences», 34 (2020) n. 6, p. 3025-3039. <https://doi.org/10.1016/j.jksuci.2020.10.001>.

Ramaḍān 2014 — Ibrāhīm ʿAbd al-karīm Mūsà Ramaḍān. *ʿalāmāt al-waqf fi-l-maṣāḥif al-maṭbūʿa. In: nadwa ṭibāʿa al-qur'ān al-karīm wa našrihi. al-Madīna al-munawwara*, 2014, p. 1603–1668 <https://ebook.univeyes.com/86194/pdf- علامات الوقف في المصاحف المطبوعة >.

Rice — Kanai — Nartker 1993 — Stephen Rice — J. Kanai — Thomas Nartker. *An Evaluation of OCR Accuracy*. In: *Annual Research Report*. Las Vegas: University of Nevada, Information Science Research Institute, 1993, p. 9-34.

Romanov et al. 2017 — Maxim Romanov — Matthew Thomas Miller — Sarah Bowen Savant — Benjamin Kiessling. *Important New Developments in Arabographic Optical Character Recognition (OCR)*. arXiv, 28 Mar 2017. <https://doi.org/10.48550/arXiv.1703.09550>.

Saber et al. 2016 — Shimaa Saber — Ali Ahmed — Ashraf Elsisi — Mohiy M. Hadhoud. *Performance Evaluation of Arabic Optical Character Recognition Engines for Noisy Inputs*. In: *The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015, Beni Suef, Egypt)*. Cham: Springer, 2016, p. 449–459. <https://doi.org/10.1007/978-3-319-26690-9_40>.

Saber — Ahmed — Hadhoud 2014 — Shimaa Saber — Ali Ahmed — Mohy Hadhoud. *Robust metrics for evaluating arabic OCR systems. In: International Image Processing, Applications and Systems Conference (IPAS)*. Piscataway (NJ): IEEE, 2014, p. 1–6. <https://doi.org/10.1109/IPAS.2014.7043272>.

Sadrzadeh — Muskens 2018 — Mehrnoosh Sadrzadeh — Reinhard Muskens. *Static and Dynamic Vector Semantics for Lambda Calculus Models of Natural Language*. «Journal of Language Modelling», 6 (2018), n. 2, p. 319–351.

Schimmel — Rivolta 1992 — Annemarie Schimmel — Barbara Rivolta. *Islamic Calligraphy*. Leiden: Brill Archive, 1992.

Sībawayhi 1989 — ʻAbū bašar ʿAmrū bin ʿŪṯmān Sībawayhi. Kitāb Sībawayhi. al-Qāhira: Maktabat al-ẖānğī, 1988.

Smith — Cordell 2018 — David Smith — Ryan Cordell. *A Research Agenda for Historical and Multilingual Optical Character Recognition*. DRS, 2018. <https://repository.library.northeastern.edu/files/neu:f1881m035>.

Sublet 1999 — Jacqueline Sublet. *Ḥiṣn al-ism : qirā'āt fī al-asmā' al-'arabiyya*, trad. di S. M. Barakat. Dimašq: Institute François de Damas, 1999.

Šukrī 2014 — ʻAḥmad Ḫālid Yūsif Šukrī. *alāmāt al-ẓabṭ fi-l-maṣāḥif: bayn al-wāqiʿ wa-l-ma'mūl. In: nadwa ṭibāʿa al-qur'ān al-karīm wa našrihi. al-Madīna al-munawwara*, 2014, p. 1483–1552. <https://ebook.univeyes.com/85064/pdf-الضبط في المصاحف بين الواقع والمأمول علامات>.

Taji et al. 2018 — Dima Taji — Salam Khalifa — Ossama Obeid — Fadhl Eryani — Nizar Habash. *An Arabic Morphological Analyzer and Generator with Copious Features*. In: *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Brussels: Association for Computational Linguistics, 2018, p. 140–150. <https://doi.org/10.18653/v1/W18-5816>.

Tang et al. 2024 — Jingqun Tang — Chunhui Lin — Zhen Zhao — Shu Wei — Binghong Wu — Qi Liu — Hao Feng et al. 2024. *TextSquare: Scaling up Text-Centric Visual Instruction Tuning*. arXiv, 9 Apr 2024. <https://doi.org/10.48550/arXiv.2404.12803>.

Taouka — Coltheart 2004 — Miriam Taouka — Max Coltheart. *The Cognitive Processes Involved in Learning to Read in Arabic*. «Reading and Writing», 17 (2004), n. 1, p. 27–57. <https://doi.org/10.1023/B:READ.0000013831.91795.ec>.

ISO 233-2 1993 — Technical Committee ISO/TC 46. *International Standard ISO 233-2: Information and Documentation — Transliteration of Arabic Characters into Latin Characters. Part 3: Arabic Language — Simplified Transliteration*. Geneva: ISO, 1993. <https://cdn.standards.iteh.ai/samples/78514/749948ae77474f7fa057a6b278281dbb/ISO-233-3-2023.pdf>.

ISO 233-3 2023 — Technical Committee ISO/TC 46. *International Standard ISO 233-3 : Information and Documentation — Transliteration of Arabic Characters into Latin Characters. Part 3: Persian Language — Transliteration*. Geneva: ISO, 2023. <https://cdn.standards.iteh.ai/samples/78514/749948ae77474f7fa057a6b278281dbb/ISO-233-3-2023.pdf>.

Wagaa — Kallel — Mellouli 2022 — Nesrine Wagaa — Hichem Kallel — Nédra Mellouli. *Improved Arabic Alphabet Characters Classification Using Convolutional Neural Networks (CNN)*. «Computational Intelligence and Neuroscience», 11 January 2022. <https://doi.org/10.1155/2022/9965426>.

Xu et al. 2023 — Derong Xu — Wei Chen — Wenjun Peng — Chao Zhang — Tong Xu — Xiangyu Zhao — Xian Wu — Yefeng Zheng — Enhong Chen. *Large Language Models for Generative Information Extraction: A Survey*. arXiv, 9 dicembre 2023. <https://doi.org/10.48550/arXiv.2312.17617>.

Zalaṭ bin Raf'at 2006 — Muḥammad bin Zalaṭ bin Raf'at. *'aḥkām al-tağwīd wa-l-tilāwa*. al-Qāhira: Mu'assasa qurṭuba, 2006. <https://www.noor-book.com/كتاب 0015 كتاب احكام التجويد والتلاوه محمد رافت زلط-pdf>.

Zoizou — Zarghili — Chaker 2020 — Abdelhay Zoizou — Arsalane Zarghili — Ilham Chaker. *A New Hybrid Method for Arabic Multi-Font Text Segmentation, and a Reference Corpus Construction*. «Journal of King Saud University - Computer and Information Sciences», 32 (2020), n. 5, p. 576–582. <https://doi.org/10.1016/j.jksuci.2018.07.003>.