Umanistica Digitale – ISSN: 2532-8816 – n. 23, 2026: Special Issue
G. Ferrara – Philo-L1. *L'*emendatio *dei testi latini come problema di* denoising
DOI: http://doi.org/10.60923/issn.2532-8816/23602

Philo-L1
L'*emendatio* dei testi latini come problema di *denoising*

Giuseppe Ferrara

University of Siena, Italy
giuseppe.ferrara@student.unisi.it

Abstract

L'emendazione dei testi letterari antichi rappresenta una delle sfide più complesse della filologia classica. I modelli esistenti per semplificare questo *task* (*Latin BERT* e *Logion*) adottano un approccio di tipo *fill-mask* che presenta alcuni limiti significativi. Questo contributo introduce *Philo-L1*, un LLM di tipo *seq2seq* di circa 297 milioni di parametri basato sull'architettura T5, che tratta l'*emendatio* dei testi letterari latini come un *task* di generazione di testo con *denoising* dell'*input* del modello, e *Ianus AI*, la piattaforma *web* pensata per il suo utilizzo. *Philo-L1*, ottenuto dal *fine-tuning* di *Philo-1-preview* (a sua volta, risultato del *fine-tuning* di *PhilTa*), è stato addestrato su un *dataset* sintetico di circa 5 milioni di coppie di frasi contenenti nove classi di corruttele: errori paleografici, di pronuncia, di *divisio*, di inversione, di eco, *saut du même au même*, errori da integrazione con parola-segnale, aplografie e dittografie. In fase di valutazione, il modello ha raggiunto un'*exact match accuracy* (EMA) del 74.01%, una *perplexity* di 1.17 e un *BLEU score* di 94.51. Il confronto diretto con *Latin BERT* conferma la validità dell'approccio proposto (EMA: 77.96% *vs* 0.50%). In futuro, si prevede di ampliare le funzionalità del modello e di integrare tecniche di *chain of thought* ed *Explainable AI*.

Parole chiave: filologia digitale; Large Language Model; Ianus AI; Philo-L1; emendazione

*The emendation of ancient literary texts is one of the most challenging tasks in classical philology. Existing models designed to assist with this task (*Latin BERT *and* Logion*) rely on a fill-mask approach that presents significant limitations. This paper introduces* Philo-L1*, a seq2seq LLM of approximately 297 million parameters based on the T5 architecture, which reframes the* emendatio *of Latin literary texts as a text generation task with input denoising, alongside* Ianus AI*, a web platform developed for its use.* Philo-L1*, obtained by fine-tuning* Philo-1-preview *(itself the result of fine-tuning* PhilTa*), was trained on a synthetic dataset of approximately 5 million sentence pairs covering nine classes of textual corruptions: palaeographic and pronunciation errors, errors of* divisio*, inversion, echo,* saut du même au même*, errors arising from integration with a signal word, haplographies, and dittographies. The model achieved an exact match accuracy (EMA) of 74.01%, a perplexity of 1.17, and a BLEU score of 94.51. A direct comparison with* Latin BERT *confirms the*

validity of the proposed approach (EMA: 77.96% vs *0.50%). Future work will focus on extending the model's scope and incorporating chain of thought and Explainable AI techniques.*

Keywords: digital philology; Large Language Model; Ianus AI; Philo-L1; emendation

1. Il problema dell'*emendatio*. Modelli SOTA vs *Philo-L1*

La trasmissione delle opere antiche per mano dei copisti nel corso dei secoli ne ha alterato la *facies*, introducendovi un numero variabile di corruttele. Recuperare la forma originaria di questi testi rappresenta una delle sfide più complesse della filologia classica, poiché richiede, al tempo stesso, solide competenze linguistiche e paleografiche e un alto livello di familiarità con lo stile dei singoli autori e generi letterari e delle diverse epoche e realtà geografiche. In questo contributo, si presentano *Philo-L1*, un *Large Language Model* (LLM) di tipo *sequence-to-sequence* basato sull'architettura T5 (*Text-to-text Transfer Transformer*) [12], specializzato nell'emendazione dei testi letterari latini, e *Ianus AI*, la piattaforma *web* sviluppata appositamente per interrogare il modello.[1]

Nel campo delle lingue antiche, i *LLMs* sono stati largamente impiegati per compiti caratteristici del *Natural Language Processing* (NLP), quali il *PoS tagging*, il *dependency parsing* e la lemmatizzazione ([4] [13][16][17][18][20]), mentre la loro applicazione alla ricostruzione dei testi antichi è un fenomeno relativamente recente. I modelli SOTA in questo campo si dividono in due categorie: 1) modelli per la ricostruzione delle epigrafi (*Pythia*, *Ithaca*, *Aeneas*) ([1][2][3]); 2) modelli per la ricostruzione dei testi letterari (*Latin BERT*, *Logion*) ([4][6][8]).

Diversamente da quanto si vedrà per *Philo-L1*, i modelli appartenenti al secondo gruppo trattano l'*emendatio*, ovvero la correzione degli errori individuati nel testo tràdito di un'opera antica con l'obiettivo di ricostruirne la versione originale [5], come un *task* di *fill-mask*, coerente con il Masked Language Modeling (MLM) utilizzato durante il loro *pre-training* ([4][6]). In termini semplici, essi predicono la parola più adatta a colmare una lacuna introdotta artificialmente all'interno della frase in corrispondenza di una parola considerata errata. Questo paradigma consente di utilizzare direttamente i modelli ottenuti attraverso il *pre-training* senza ulteriori *fine-tuning*, ma comporta alcuni svantaggi significativi. Da un lato, rende indispensabile lo sviluppo di un sistema che consenta al modello di individuare correttamente, in autonomia, le parole errate presenti nella frase analizzata. Dall'altro, scarta a priori gli indizi della lezione originale ancora presenti nella parola corrotta, che nella pratica filologica guidano la ricostruzione del testo. Da ultimo, l'approccio *fill-mask* risulta strutturalmente inadeguato per la correzione di alcune categorie di errori che non si configurano come semplici sostituzioni di una parola per un'altra, tra cui gli errori di *divisio*, di inversione dell'*ordo verborum* o dovuti a integrazione con parola-segnale.

I risultati di *Latin BERT* e *Logion* nel *task* di *emendatio* confermano i limiti di tale approccio: il primo, infatti, raggiunge una *top-1 accuracy* del 33.1% [4], il secondo del 58.3% [6].

1 Una volta terminata la fase di beta testing, Philo-L1 e Ianus AI verranno distribuiti sotto licenza GNU GPLv3 rispettivamente su https://huggingface.co/giuseppeferrara/philo-l1 (cons. 04/12/2025) e https://github.com/giuseppeferrara/ianus-ai.git (cons. 04/12/2025).

Il paradigma di *Philo-L1* si basa su una diversa modellazione dell'*emendatio* come *task* di generazione di testo (*text-to-text*) con rimozione del rumore presente nell'*input* del modello (*denoising*). Tale formulazione del problema trova il suo fondamento nella teoria dell'informazione [14][15]. Mutuando gli assunti di tale teoria, il testo originale può essere considerato come un segnale X trasmesso attraverso un canale rumoroso (le operazioni di copiatura realizzate dagli scribi) [11]. Il prodotto di tale processo è un segnale Y identificabile con il testo tràdito, tale che

$$Y = f(X, N)$$

dove N sono gli errori introdotti nel testo e f è il processo di corruzione testuale. Le corruttele filologiche, dunque, rappresentano il rumore del segnale Y, l'emendazione il *denoising* del segnale, l'informazione mutua $I(X;Y)$ la misura della quantità di segnale originale ancora presente nel segnale corrotto. Perché l'emendazione sia possibile, deve valere $I(X;Y) > 0$. In altri termini, perché si possa ricostruire il testo originale è indispensabile che tracce grafiche, fonetiche e morfo-sintattiche della lezione originaria persistano nella lezione corrotta. In questa prospettiva, l'obiettivo di *Philo-L1* è apprendere la funzione

$$g: Y \rightarrow \hat{X} \text{ tale che } \hat{X} \approx X$$

che, a partire da uno stadio più o meno rumoroso del testo Y, restituisca la ricostruzione $\hat{X}$, il più vicina possibile alla versione originaria del testo priva di rumore.

In sintesi, lo sviluppo di *Philo-L1* è stato guidato dalla volontà di superare le limitazioni dei modelli esistenti attraverso una piena valorizzazione dell'informazione presente nel testo tràdito e della conoscenza filologica dei meccanismi che hanno portato alla genesi delle diverse categorie di corruttele.

2. Architettura, *dataset* e *fine-tuning* di *Philo-L1*

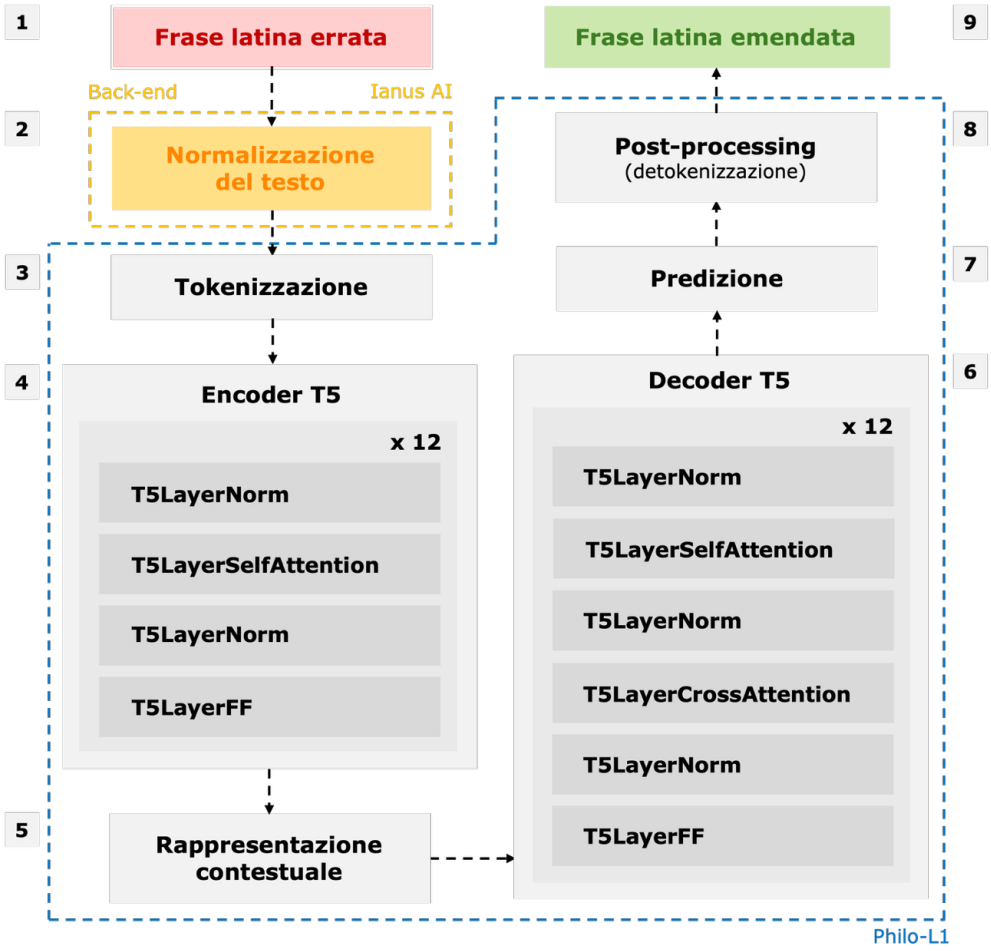

Fig. 1 - Architettura di *Philo-L1* con flusso logico di *Ianus AI*.

Philo-L1 è un modello di ~297 milioni di parametri ottenuto attraverso il *fine-tuning* di *Philo-1-preview* [7]. Quest'ultimo, a sua volta, è un modello multilingue sperimentale da me sviluppato (per la correzione, nei testi greci e latini, degli errori paleografici, di pronuncia e di aplografia e dittografia di singole lettere interne a una stessa parola) attraverso il *fine-tuning* di *PhilTa*, una variante *encoder-decoder* multilingue del modello *T5-Base* pre-addestrata su vari *corpora* di testi greci e latini [13].

L'architettura di *Philo-L1*, integrata con il flusso logico della piattaforma *Ianus AI*, è riportata in Fig. 1. Il testo latino corrotto per la presenza, al suo interno, di uno o più errori viene sottoposto ad alcune operazioni di normalizzazione (rimozione delle lettere ramiste e della punteggiatura, sostituzione delle lettere maiuscole con le rispettive minuscole) per convertirlo nella forma attesa dal modello. Il testo normalizzato viene, quindi, tokenizzato e fornito in *input* all'*encoder* di T5,

che ne produce una rappresentazione contestuale utilizzando il meccanismo di *self-attention* dell'architettura *Transformer*. Il *decoder* di T5, a sua volta, sfrutta una *self-attention* mascherata sui *token* già generati e una *cross-attention* sulla rappresentazione fornita dall'*encoder* per predire la sequenza di *token* della frase emendata, la quale viene poi convertita in una stringa testuale attraverso la detokenizzazione.

L'obiettivo del *fine-tuning* è stato dotare il modello della capacità di svolgere il seguente *task*: prendere in *input* una frase corrotta Y e restituire in *output* la frase $\hat{X}$, il più prossima possibile alla versione originaria del testo priva di rumore X, sfruttando la comprensione del contesto, la conoscenza della lingua latina modellata sul *corpus* utilizzato per il *pre-training* e la conoscenza dei meccanismi di corruzione dei testi letterari latini. In termini matematici, il *fine-tuning* mira a risolvere il problema di ottimizzazione

$$\min_{\theta} \sum_{i=1}^{n} \mathcal{L}(g_\theta(Y_i), X_i)$$

dove θ rappresenta i parametri del modello e $\mathcal{L}$ la funzione di *loss* calcolata sulla sequenza *target*.

Per poter eseguire il *fine-tuning*, è stato necessario sviluppare preliminarmente alcuni *script* Python che replicassero i meccanismi di alterazione alla base delle diverse categorie di corruttele filologiche trattate da *Philo-L1*: errori paleografici (di maiuscola e minuscola) e di pronuncia, di *divisio*, di inversione (alterazione dell'*ordo verborum*, anasillabismo e inversione di liquide), di eco (perseverazione e anticipazione), *saut du même au même*, errori dovuti a integrazione con parola-segnale, aplografie e dittografie (di singole lettere, sillabe e parole monosillabiche) [5]. I testi latini a cui applicare gli *script* sono stati ricavati dal *corpus* di *Perseus Digital Library*[2] attraverso alcune operazioni di *pre-processing*: eliminazione degli elementi paratestuali (*tag XML*, commenti editoriali, note, apparati critici) ed estrazione delle porzioni testuali in lingua latina; normalizzazione *Unicode* NFD; conversione in minuscolo; eliminazione dei segni di punteggiatura. Sono stati, dunque, generati tre *dataset* sintetici (di *training*, *validation* e *test*, con *split* 90/5/5) di coppie di frasi (errata/corretta) per ciascuna tipologia di errori, per un totale di 5.314.139 *record* unici.

2 https://www.perseus.tufts.edu/hopper/.

Categorie di errori (con relativo numero di sottocategorie)	***Training set***	***Validation set***	***Test set***
Paleografici e di pronuncia (166)	1.583.355	90.975	90.975
Divisio (4)	400.003	22.222	22.222
Inversione (6)	600.000	33.336	33.336
Eco (2)	121.692	6.762	6.760
Saut du même au même (2)	213.276	11.847	11.853
Integrazione con parola-segnale (10)	1.000.000	55.560	55.560
Aplografie (5)	358.998	19.942	19.951
Dittografie (5)	500.000	27.775	27.785
TOTALE	**4.777.286**	**268.440**	**268.413**

Tab. 1 - Composizione dei dataset di *training*, *validation* e *test* per categoria di errore utilizzati per il *fine-tuning* di *Philo-L1*.

Compatibilmente con le dimensioni e la varietà del *corpus* utilizzato, si è cercato di bilanciare la distribuzione degli esempi tra le diverse tipologie di errore in ciascun *dataset*, al fine di limitare fenomeni di *class imbalance* che avrebbero potuto compromettere le *performance* del modello sulle classi meno rappresentate [10]. Si è cercato, inoltre, di organizzare ciascuna categoria in modo che contenesse un numero analogo di esempi con parole latine valide e *voces nihili* come lezioni corrotte. Il *target* per ciascuna sottocategoria di ciascuna classe di corruttele è stato di 100.000 esempi per il *training set* e di 5.556 esempi per i *dataset* di *validation e test*. Per gli errori paleografici e di pronuncia, tuttavia, l'elevato numero di regole di trasformazione ha reso necessario limitare gli esempi per regola a 10.000 nel *training set* e a 556 nei *dataset* di *validation* e *test*. La composizione di ciascun *dataset* è riportata in Tab. 1.

Iperparametro	**Valore**
Numero di epoche	4
Learning rate	2.12×10^{-4}
Batch size	16
Weight decay	0.30

Tab. 2 - Iperparametri utilizzati per il *fine-tuning* di *Philo-L1*.

Gli iperparametri utilizzati per il *fine-tuning* di *Philo-L1* (vd. Tab. 2) riprendono la configurazione già validata durante lo sviluppo di *Philo-1-preview* [7]. Il valore del *weight decay* (0.30), sebbene particolarmente elevato, si è rivelato efficace nel contrastare il rischio di *overfitting* e nel favorire l'apprendimento dei meccanismi di correzione, invece della memorizzazione degli esempi del *training set*. L'unica modifica rispetto alla configurazione originaria ha riguardato il numero di epoche, ridotto da 5 a 4: la convergenza, infatti, è risultata più rapida, in quanto il modello di partenza era già stato specializzato su un *task* affine a quello di *Philo-L1*. Il *fine-tuning* del modello, della durata di circa 4 ore, è stato svolto su un *cluster* di 8 GPU NVIDIA H100.

3. Valutazione e validazione di *Philo-L1*

La valutazione delle *performance* di *Philo-L1* è stata condotta utilizzando un ampio spettro di metriche: *exact match accuracy* (EMA), *token accuracy*, *character accuracy*, *Character Error Rate* (CER), *BLEU score*, *chrF score*, *precision*, *recall*, *F1 score*, *perplexity* e *cross-entropy loss*.

Metrica	**Valore**
Exact match accuracy	74.01%
Token accuracy	91.02%
Character accuracy	88.43%
Character Error Rate	4.16%
BLEU score	94.51
chrF score	97.35
Precision	98.72%
Recall	97.75%
F1 score	98.24%
Perplexity	1.17
Cross-entropy loss	0.153

Tab. 3 - Risultati complessivi della valutazione di *Philo-L1* sul *test set*.

I risultati complessivi sul *test set* (vd. Tab. 3) dimostrano che, grazie al *fine-tuning*, *Philo-L1* ha raggiunto *performance* elevate nel *task* di interesse. In particolare, l'EMA del 74.01% indica che, quando si richiede al modello di effettuare una predizione, questa coincide esattamente con la frase emendata attesa in più di 7 casi su 10. Inoltre, la *perplexity* di 1.17, molto vicina a 1, dimostra che il modello è particolarmente sicuro quando formula una congettura. Il *BLEU score*

di 94.51, infine, segna un incremento di 27.01 punti rispetto al valore registrato dal modello sperimentale *Philo-1-preview* (67.50) [7]. Questo miglioramento può essere ricondotto sia alla maggiore dimensione e varietà del *training set* sia alla specializzazione monolingue del modello nel dominio del latino.

Categoria di errore	**EMA**	**CER**
Paleografici e di pronuncia	74.44%	3.60%
Divisio	72.29%	4.31%
Inversione	72.69%	3.17%
Eco	76.47%	3.28%
Saut du même au même	70.10%	4.65%
Integrazione con parola-segnale	73.13%	4.19%
Aplografie	76.62%	3.33%
Dittografie	73.91%	4.90%

Tabella 4 - Analisi stratificata delle *performance* di *Philo-L1* per categoria di errore in termini di EMA e CER.

Sebbene nell'analisi stratificata per categoria di errore (vd. Tabella 4) le *performance* del modello appaiano abbastanza omogenee su tutte le tipologie di corruttele su cui è stato condotto il *fine-tuning*, si possono effettuare alcuni raggruppamenti in base al valore di EMA. I risultati migliori (≃76%) si registrano per gli errori di aplografia (76.62%) e di eco (76.47%). È probabile che i primi risultino più facili da identificare per il modello per due ragioni: 1) perché il loro esito coincide spesso con una *vox nihili* che può essere agevolmente individuata e mappata alla voce corretta corrispondente; 2) perché, nei casi in cui l'aplografia riguarda la rimozione di parole monosillabiche ad alta frequenza (ad esempio, *et*, *in* o *ad*), il modello deve limitarsi a reinserire questi elementi facilmente inferibili dal contesto. Gli errori di eco, invece, devono risultare più semplici da correggere in quanto caratterizzati da *pattern* sufficientemente regolari, limitati alla modifica di prefissi e desinenze per influsso del prefisso o della desinenza di un'altra parola della frase. A seguire, si hanno le categorie con EMA ≃74%: errori di dittografia (73.91%) ed errori paleografici e di pronuncia (74.44%). Almeno in linea teorica, ci si potrebbe attendere che le *performance* del modello sulle dittografie risultassero più prossime a quelle sulle aplografie, visto che le due categorie di corruttele hanno meccanismi di generazione speculari. Tuttavia, la maggiore convergenza dei risultati verso gli errori paleografici e di pronuncia, che beneficiano della regolarità dei *pattern* di trasformazione a livello delle singole parole, si può ricondurre alla presenza, tra le aplografie, della sottocategoria relativa all'omissione di parole monosillabiche ad alta frequenza di cui si è parlato, che alza la media di questa classe di errori e non ha un corrispettivo speculare nelle dittografie. Con un'EMA inferiore (≃72-73%) si hanno gli errori dovuti a integrazione con parola-segnale (73.13%), quelli di inversione (72.69%) e quelli di *divisio* (72.29%). La leggera riduzione di *performance* si spiega, in questo caso, con il fatto che

queste categorie di corruttele comportano una riorganizzazione complessiva della frase, per cui è necessaria una comprensione sintattica del testo più profonda rispetto a quella necessaria per le corruttele a livello di parola. Infine, le *performance* più basse (≃70%) si registrano per la categoria del *saut du même au même* (70.10%), che richiede l'identificazione e la ricostruzione di porzioni di testo omesse di lunghezza variabile e non nota *a priori*.

Accanto alla valutazione quantitativa, è utile illustrare le *performance* del modello attraverso un esempio concreto. Si consideri il verso 470 del terzo libro delle *Georgiche* di Virgilio, trasmesso da P con la lezione errata *greber* (al posto di *creber*), riconducibile a un errore paleografico dovuto allo scambio di lettere di forma simile in onciale [9]: «non tam greber agens hiemem ruit aequore turbo».[3] Interrogato su questo passo, *Philo-L1* restituisce correttamente il testo atteso «non tam creber agens hiemem ruit aequore turbo» come prima congettura. L'esempio è riportato in Fig. 2.

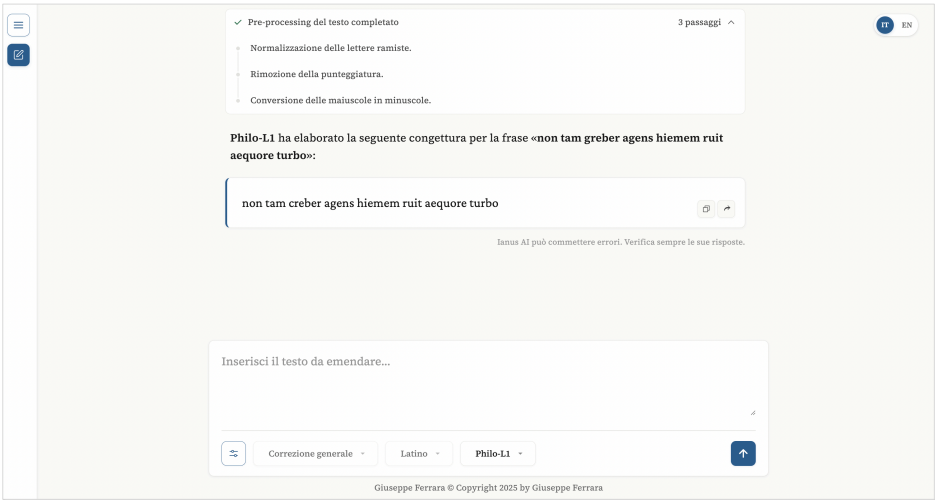

Fig. 2 - Esempio di emendazione di Verg. georg. 3, 470 con *Philo-L1* (con numero di correzioni alternative impostato a 1) su *Ianus AI*.

Per valutare quale approccio sia più efficace tra quello *fill-mask* e quello *text-to-text* combinato con il *denoising*, si è realizzato anche un confronto tra *Latin BERT* e *Philo-L1* sullo stesso *dataset*, circoscritto soltanto alle categorie di corruttele strutturalmente compatibili con il paradigma *fill-mask*. In particolare, si sono selezionati gli errori paleografici e di pronuncia, gli errori di inversione riguardanti elementi interni alla stessa parola, gli errori di eco, gli errori di aplografia e dittografia con omissione o duplicazione di elementi interni alla stessa parola. Il *dataset* così definito si è limitato a 143.620 esempi, in cui, per *Latin BERT*, si sono sostituite le parole corrotte con il *token [MASK]*. Il confronto è stato circoscritto all'EMA: poiché, infatti, l'approccio *fill-mask* interviene su una sola parola della frase, lasciando invariato il contesto, le

3 Verg. georg. 3, 470.

altre metriche avrebbero prodotto una sovrastima artificiosa delle *performance* di *Latin BERT* rispetto a *Philo-L1*, che interviene, invece, sull'intera frase.

Modello	**EMA**
Latin BERT	0.50%
Philo-L1	77.96%

Tab. 5 - Risultati del confronto tra *Latin BERT* e *Philo-L1* in termini di EMA.

I risultati del confronto sono riportati in Tab. 5. *Latin BERT* ha raggiunto un'EMA dello 0.50%, contro il 77.96% di *Philo-L1*. Questi valori dimostrano che il paradigma utilizzato nello sviluppo di *Philo-L1*, basato sull'architettura T5, rappresenta una scelta più efficace per il *task* di *emendatio*, che richiede sia un'adeguata valutazione del contesto in cui si inseriscono le congetture sia l'utilizzo dell'informazione relativa alla lezione corretta ancora rintracciabile nella lezione errata come guida nel processo di emendazione.

4. La piattaforma *Ianus AI*

Per rendere *Philo-L1* facilmente interrogabile, è stata sviluppata la piattaforma *web Ianus AI*, disponibile sia in italiano sia in inglese e dotata di un *design* che minimizza la curva di apprendimento dell'utente per massimizzare l'usabilità dell'applicazione.

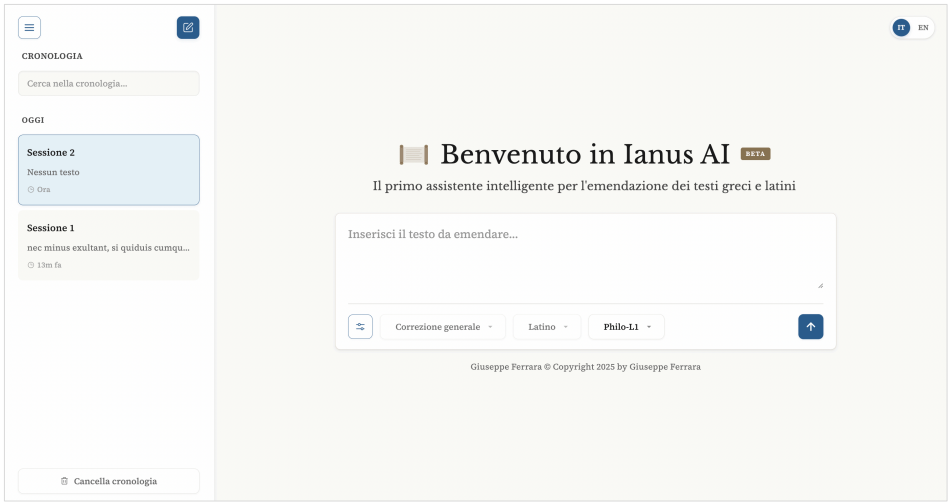

Fig. 3 - *Homepage* di *Ianus AI*, con *sidebar* espansa a sinistra e area di *input* a destra.

L'interfaccia si articola in tre componenti. Il primo (vd. Fig. 3) è la *sidebar* espandibile posizionata sul lato sinistro dello schermo, che permette di gestire la cronologia delle sessioni di correzione svolte con il modello. Di ciascuna sessione sono indicati il nome (modificabile dall'utente), la

data dell'ultima modifica e un'anteprima del testo latino analizzato. All'interno della *sidebar*, sono presenti due bottoni, uno per la creazione di nuove sessioni e uno che permette di cancellare la cronologia.

Il secondo componente è rappresentato dall'area di *input* (vd. Fig. 3), centrata all'interno della pagina, che ospita una *text area* per l'inserimento del testo da emendare e una barra con alcuni menu di configurazione: uno per la selezione della lingua del testo (la piattaforma, infatti, è stata pensata per ospitare modelli di correzione per i testi sia latini sia greci), uno per la scelta del modello che si vuole utilizzare per l'emendazione, uno per la modalità di correzione impiegata (generale o mirata), uno per la regolazione di alcuni parametri avanzati di generazione, quali il numero di correzioni alternative richieste al modello (1-5), l'ampiezza del *beam search* (1-10) e la temperatura del modello (0.1-1.0).

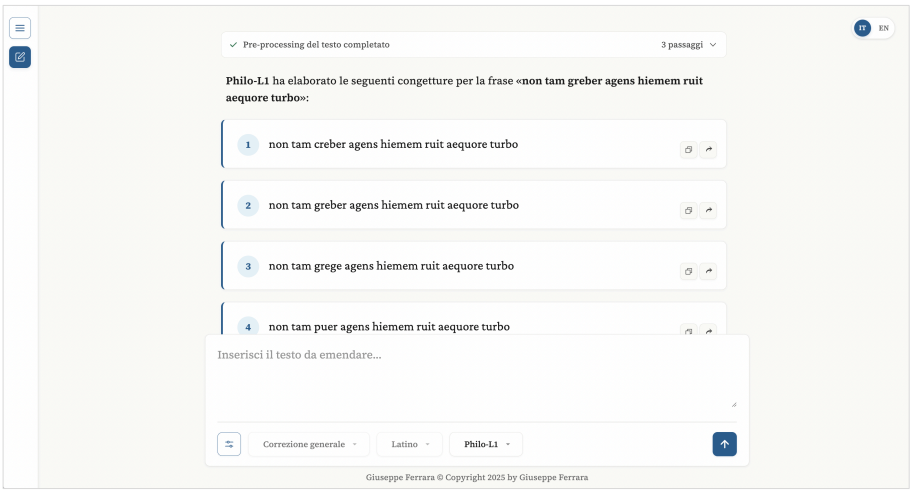

Fig. 4 - Schermata di *Ianus AI* con area dei risultati visibile per l'esempio di Verg. georg. 3, 470 (con numero di correzioni alternative impostato a 5).

L'ultimo componente è rappresentato dall'area dei risultati (vd. Fig. 4), in cui vengono visualizzate le congetture proposte dal modello in blocchi distinti per ciascuna richiesta inviata. Ogni blocco è introdotto da un elemento espandibile che sintetizza le operazioni di *pre-processing* realizzate sul testo immesso dall'utente, riportato di seguito insieme al nome del modello utilizzato. Subito sotto sono presentate le singole congetture elaborate dal modello, accompagnate da due pulsanti di azione, l'uno pensato per copiare il testo negli appunti di sistema, l'altro per riportare il testo della congettura direttamente nella *text area* della sessione, dove potrà essere utilizzato come nuovo *input* del modello.

L'applicazione, infine, implementa un sistema di notifiche che fornisce un *feedback* immediato all'utente sulle principali operazioni svolte all'interno della piattaforma e un meccanismo di persistenza locale dei dati, basato sul *localStorage* del *browser*, che permette il salvataggio automatico delle sessioni e delle preferenze generali della piattaforma.

5. Limiti e sviluppi futuri

Come si è cercato di dimostrare, i risultati ottenuti da *Philo-L1* sono incoraggianti. Tuttavia, il modello presenta alcuni limiti che è opportuno indagare e che possono orientare gli sviluppi futuri di questa ricerca.

Il primo limite riguarda il numero di classi di errori corrette dal modello. Allo stato attuale, *Philo-L1* è in grado di correggere soltanto 9 delle 16 categorie di corruttele effettivamente riscontrabili nei testi letterari in lingua latina. Per questo motivo, sarà necessario espandere la rosa di errori corretti dal modello, in modo da ampliarne il campo di applicazione.

Il secondo limite riguarda la correzione di combinazioni di errori di natura diversa all'interno della stessa parola o della stessa frase. Sebbene *Philo-L1* mostri questa abilità come proprietà emergente, essa non è stata oggetto di un addestramento sistematico. Un ulteriore obiettivo, dunque, sarà sviluppare una versione del modello ottimizzata anche per questo *task*.

Il terzo limite riguarda la natura sintetica dei *dataset* utilizzati per il *fine-tuning*. Sebbene gli *script* Python sviluppati per la generazione dei *record* siano stati progettati per riprodurre in maniera accurata i meccanismi di generazione delle singole categorie di errori, le corruttele reali presenti nei manoscritti possono risultare talvolta più complesse. In futuro, dunque, si prevede di integrare i tre *dataset* utilizzati con esempi estratti direttamente dalle edizioni critiche delle opere antiche, in modo da addestrare e valutare il modello anche su alcuni di essi.

Il quarto limite, infine, riguarda l'interpretabilità del processo di generazione delle congetture. Attualmente, *Philo-L1* restituisce la frase emendata senza esplicitare il ragionamento sottostante alla congettura proposta e quali indizi testuali abbiano guidato l'emendazione. L'integrazione di una *chain of thought* (CoT) [19], combinata con tecniche di *Explainable AI* (XAI), potrebbe risolvere questa criticità, rendendo *Ianus AI* uno strumento più avanzato da utilizzare nella pratica ecdotica di livello accademico.

Riferimenti

[1] Assael, Yannis, Thea Sommerschield, Alison Cooley, Brendan Shillingford, John Pavlopoulos, Priyanka Suresh, Bailey Herms, et al. 2025. "Contextualizing Ancient Texts with Generative Neural Networks". *Nature* 645 (8079): 141–147. https://doi.org/10.1038/s41586-025-09292-5.

[2] Assael, Yannis, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, e Nando de Freitas. 2022. "Restoring and Attributing Ancient Texts Using Deep Neural Networks". *Nature* 603 (7900): 280–83. https://doi.org/10.1038/s41586-022-04448-z.

[3] Assael, Yannis, Thea Sommerschield, e Jonathan Prag. 2019. "Restoring ancient text using deep learning: a case study on Greek epigraphy". arXiv preprint aXiv:1910.06262.

[4] Bamman, D., e P. J. Burns. 2020. "Latin BERT: A Contextual Language Model for Classical Philology". arXiv preprint arXiv:2009.10053.

[5] Braccini, Tommaso. 2017. *La scienza dei testi antichi. Introduzione alla filologia classica*. Le Monnier Università.

[6] Cowen-Breen, Charlie, Creston Brooks, Johannes Haubold, e Barbara Graziosi. 2023. "Logion: Machine Learning for Greek Philology". arXiv preprint arXiv:2305.01099.

[7] Ferrara, Giuseppe. 2025. "Philo-1-preview. Un modello T5-Base per l'emendazione dei testi antichi". In *Diversità, Equità e Inclusione: Sfide e Opportunità per l'Informatica Umanistica nell'Era dell'Intelligenza Artificiale, Proceedings del XIV Convegno Annuale AIUCD2025*, a cura di Simone Rebora, Marco Rospocher, e Stefano Bazzaco, 404-410. AIUCD. https://doi.org/10.6092/unibo/amsacta/8380.

[8] Graziosi, Barbara, Johannes Haubold, Charlie Cowen-Breen, e Creston Brooks. 2023. "Machine Learning and the Future of Philology: A Case Study". *TAPA* 153 (1): 253–84. https://doi.org/10.1353/apa.2023.a901022.

[9] Havet, Louis. 1911. *Manuel de critique verbale appliquée aux textes latins*. Hachette.

[10] Johnson, Justin M. e Taghi M. Khoshgoftaar. 2019. "Survey on Deep Learning with Class Imbalance". *Journal of Big Data* 6 (1): 27. https://doi.org/10.1186/s40537-019-0192-5.

[11] Kernighan, Mark D., Kenneth W. Church, e William A. Gale. 1990. "A spelling correction program based on a noisy channel model". In *Proceedings of the 13th conference on Computational linguistics - Volume 2* (USA), 205–10. https://doi.org/10.3115/997939.997975.

[12] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, e Peter J. Liu. 2023. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". arXiv preprint arXiv:1910.10683.

[13] Riemenschneider, Frederick e Anett Frank. 2023. "Exploring Large Language Models for Classical Philology". arXiv preprint arXiv:2305.13698.

[14] Shannon, Claude E. 1948. "A Mathematical Theory of Communication". *Bell System Technical Journal* 27 (3): 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[15] Shannon, Claude E. e Warren Weaver. 1998. *The Mathematical Theory of Communication*. University of Illinois Press. https://books.google.it/books?id=IZ77BwAAQBAJ.

[16] Singh, Pranaydeep, Gorik Rutten e Els Lefever. 2021. "A Pilot Study for BERT Language Modelling and Morphological Analysis for Ancient and Medieval

Greek". In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, a cura di Stefania Degaetano-Ortlieb, Anna Kazantseva, Nils Reiter, Stan Szpakowicz, 128–37. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.latechclfl-1.15.

[17] Sommerschield, Thea, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, e Nando de Freitas. 2023. "Machine Learning for Ancient Languages: A Survey". *Computational Linguistics* 49 (3): 703–47. https://doi.org/10.1162/coli_a_00481.

[18] Straka, Milan, Jana Straková, e Federica Gamba. 2024. "ÚFAL LatinPipe at EvaLatin 2024: Morphosyntactic Analysis of Latin". arXiv preprint arXiv:2404.05839.

[19] Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, e Denny Zhou. 2023. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". arXiv preprint arXiv:2201.11903.

[20] Wróbel, Krzysztof, e Krzysztof Nowak. 2022. "Transformer-based Part-of-Speech Tagging and Lemmatization for Latin". In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, a cura di Rachele Sprugnoli e Marco Passarotti, 193–97. European Language Resources Association. https://aclanthology.org/2022.lt4hala-1.31/.