Concordanze e NLP: idee, metodi e regole per l'applicazione alla lingua italiana

Pietro Sichera[1], Christian D'Agata[2], Giuseppe Palazzolo[3]
[1]CNR ILIESI, Italia pietro.sichera@cnr.it
[2]Università degli Studi di Catania, christian.dagata@unict.it
[3]Università degli Studi di Catania, giuseppe.palazzolo@unict.it

ABSTRACT (ITALIANO)

La lemmatizzazione automatica dei testi rappresenta uno strumento fondamentale nell'elaborazione del linguaggio naturale (NLP), in quanto consente di associare le occorrenze testuali ai loro lemmi, ovvero alle voci di vocabolario da cui derivano. Questo processo, essenziale per l'analisi linguistica e semantica, presenta tuttavia sfide significative quando applicato alla lingua italiana. In particolare, le peculiarità morfologiche della lingua, come la presenza di enclitiche e omografi, rendono necessaria l'adozione di approcci che coniughino tecnologie avanzate e interventi di disambiguazione personalizzati.
In questo lavoro si analizza l'uso combinato di spaCy e UDPipe, per la lemmatizzazione automatica di testi italiani, evidenziandone i punti di forza e le limitazioni. Tali strumenti sono integrati con il software LiotroConcord_v2, sviluppato su piattaforma 4D, implementando regole personalizzate per la gestione di casi complessi. Questo studio si propone di discutere l'importanza di un approccio ibrido alla lemmatizzazione, che integri metodi automatici e correttivi manuali, al fine di migliorare la qualità dell'analisi linguistica nel contesto delle edizioni digitali e della ricerca filologica. In particolare, vengono approfonditi i contributi metodologici e tecnologici del sistema sviluppato, con uno sguardo alle potenziali applicazioni future e alla possibilità di estendere tali soluzioni a ulteriori ambiti di ricerca. Nella seconda parte viene presentato un caso di studio specifico relativo alla codifica e alla lemmatizzazione de I Viceré.
Parole chiave: lemmatization; concordance; NLP; model; digital scientific editions.

ABSTRACT (ENGLISH)

Concordances and NLP: ideas, methods and rules for application to the Italian language
Automatic text lemmatization represents a fundamental tool in natural language processing (NLP), as it allows textual occurrences to be associated with their lemmas, i.e., the vocabulary items from which they are derived. However, this process, which is essential for linguistic and semantic analysis, presents significant challenges when applied to the Italian language. In particular, the morphological peculiarities of the language, such as the presence of enclitics and homographs, make it necessary to adopt approaches that combine advanced technologies and customized disambiguation interventions. This paper analyzes the combined use of spaCy and UDPipe, for automatic lemmatization of Italian texts, highlighting their strengths and limitations. These tools are integrated with LiotroConcord_v2 software, developed on a 4D platform, implementing custom rules for handling complex cases. This study aims to discuss the importance of a hybrid approach to lemmatization, integrating automatic methods and manual corrections, in order to improve the quality of linguistic analysis in the context of digital editions and philological research. In particular, the methodological and technological contributions of the developed system are discussed in depth, with a look at potential future applications and the possibility of extending such solutions to further areas of research. In the second part, a specific case study related to the encoding and lemmatization of I Viceré is presented.
Keywords: lemmatization; concordance; NLP; model; digital scientific editions

1. INTRODUZIONE

La lemmatizzazione automatica dei testi rappresenta uno dei pilastri fondamentali nell'elaborazione del linguaggio naturale (Natural Language Processing, NLP), in quanto consente di collegare ogni occorrenza testuale al suo lemma, ovvero alla forma di base da cui deriva. Questo processo risulta particolarmente importante nel contesto dell'umanistica digitale, dove l'analisi dei testi richiede una comprensione accurata delle strutture linguistiche, al fine di creare strumenti affidabili per la ricerca filologica, la costruzione di edizioni digitali e l'interrogazione semantica dei corpora.
La lingua italiana, tuttavia, presenta una serie di peculiarità che rendono la lemmatizzazione un compito particolarmente complesso. Tra queste si annoverano la presenza di enclitiche, che si legano a forme verbali generando token composti, e gli omografi, parole con la stessa forma grafica ma significati e categorie grammaticali differenti. Questi fenomeni richiedono soluzioni avanzate che vadano oltre la

semplice associazione tra forma e lemma, includendo anche la disambiguazione contestuale e la segmentazione delle forme complesse.
La difficoltà di questi problemi è ulteriormente amplificata quando le variazioni linguistiche, le forme arcaiche e la mancanza di standardizzazione rendono inapplicabili molti strumenti di NLP progettati per il linguaggio contemporaneo. Pertanto, l'adozione di sistemi combinati, che uniscano le capacità di diverse tecnologie NLP, può rappresentare una soluzione efficace per affrontare tali complessità.
Questo articolo si concentra sull'analisi e l'integrazione di due tra i più avanzati strumenti di lemmatizzazione automatica: spaCy[1] e UDPipe.[2] Entrambi i sistemi sono stati scelti per la loro capacità di gestire testi italiani con un alto grado di accuratezza e per la possibilità di personalizzare le pipeline di analisi linguistica. Tuttavia, per affrontare le limitazioni intrinseche di ciascun sistema, è stato sviluppato un approccio ibrido che combina i risultati di spaCy e UDPipe con interventi di disambiguazione e segmentazione realizzati tramite il software LiotroConcord_v2, una piattaforma progettata per integrare l'automazione e il controllo manuale nell'elaborazione dei testi.
L'obiettivo del presente lavoro è duplice: da un lato, evidenziare le sfide specifiche della lemmatizzazione automatica per la lingua italiana e, dall'altro, presentare un metodo pratico che sfrutti l'integrazione di tecnologie avanzate per migliorare l'affidabilità dell'analisi linguistica. In particolare, si pone l'accento sull'importanza di approcci personalizzabili per il trattamento delle enclitiche, degli omografi e delle forme linguistiche non standard, contribuendo a una migliore comprensione e accessibilità dei testi nel contesto dell'umanistica digitale.

2. STATO DELL'ARTE

Storicamente, la lemmatizzazione è stata affrontata tramite due approcci principali: basati su regole e basati su modelli statistici. Mentre i sistemi tradizionali si affidavano a dizionari e regole morfologiche, le tecnologie contemporanee sfruttano modelli di apprendimento supervisionato e reti neurali per elaborare testi con maggiore accuratezza e flessibilità.
La lingua italiana presenta peculiarità che mettono alla prova le capacità dei lemmatizzatori. Tra queste, si evidenziano le enclitiche e le forme composte, come nel caso di parole quali "trovandotelo", che contengono forme verbali con pronomi oggetto e indiretti annessi, richiedendo una segmentazione corretta per individuare i lemmi "trovare", "te" e "lo". Un ulteriore ostacolo è costituito dagli omografi, ovvero termini come "calcio", che possono avere significati e categorie grammaticali diverse a seconda del contesto, come sostantivo o verbo. Le variazioni diacroniche e arcaiche presenti nei testi letterari e storici, ad esempio "vo" invece di "vado", complicano ulteriormente l'analisi automatica. Sebbene i modelli moderni come quelli di spaCy e UDPipe riescano a gestire alcune di queste problematiche attraverso reti neurali e addestramenti su corpora diversificati, non sono privi di limitazioni. Ad esempio, i modelli pre-addestrati per l'italiano spesso non distinguono correttamente tra parole contenenti enclitiche o forme non standard, richiedendo interventi manuali o l'uso di regole personalizzate.
Uno dei contributi metodologici fondamentali nel trattamento dei testi italiani è il metodo concordanziale di Giuseppe Savoca (Savoca G., 2000), che ha gettato le basi per un'analisi testuale sistematica nell'umanistica digitale. Come descritto da Di Silvestro & al. (2022), questo approccio si concentra sull'analisi lessicografica delle parole, costruendo concordanze dove ogni occorrenza è messa in relazione con il suo lemma attraverso il contesto d'uso. Questo metodo attribuisce particolare importanza alla parola come unità fondamentale dell'analisi, tralasciando informazioni grammaticali come tempo e numero nei verbi. Tale impostazione nasce dall'esigenza di mantenere i dati gestibili nel contesto di un processo quasi completamente manuale, evitando una granularità eccessiva nella catalogazione. Sebbene il metodo non sia automatizzato, ha influenzato lo sviluppo di strumenti di umanistica digitale, evidenziando l'importanza di approcci manuali per affrontare casi complessi.
Nel presente lavoro, il metodo concordanziale trova un'applicazione moderna nell'uso del software LiotroConcord_v2, che combina strumenti di lemmatizzazione automatica (spaCy e UDPipe) con regole

1 spaCy è una libreria Python open-source, progettata per essere altamente scalabile e performante, utilizzando modelli pre-addestrati basati su reti neurali per l'analisi morfosintattica e la lemmatizzazione https://spacy.io/ (cons. 26/01/2025)

2 UDPipe, sviluppato dall'Institute of Formal and Applied Linguistics, sfrutta i dati Universal Dependencies (UD) per addestrare modelli di analisi sintattica e morfologica su corpora annotati. La combinazione di analisi sintattica e morfologica rende UDPipe particolarmente adatto per lingue ricche di morfologia come l'italiano. https://ufal.mff.cuni.cz/udpipe (cons. 26/01/2025)

personalizzate implementate manualmente. Questo approccio consente di colmare le lacune dei modelli NLP, estendendo l'efficacia della lemmatizzazione anche ai testi storici e letterari.
Nonostante i progressi significativi, gli strumenti attuali presentano alcune limitazioni. La mancanza di personalizzazione rimane un problema centrale, poiché i modelli pre-addestrati spesso non tengono conto delle specificità di corpora storici o letterari, come la ridotta efficacia nella risoluzione delle ambiguità semantiche, legate a fenomeni come l'omografia e la polisemia: la dipendenza dai corpora annotati limita l'applicabilità dei modelli a domini linguistici poco rappresentati, evidenziando la necessità di soluzioni più flessibili e adattabili (Ciula & al., 2023).

3. NLP E LEMMATIZZAZIONE AUTOMATICA

Il processo di lemmatizzazione automatica per la lingua italiana può presentare diverse problematiche. I sistemi NLP offrono certamente un valido supporto al processo, ma alcuni passi devono essere trattati con un automatismo limitato, come la gestione delle enclitiche, degli omografi e delle forme linguistiche non standard. Per affrontare tali complessità, il flusso di lavoro integrato sviluppato nel software LiotroConcord_v2 utilizza un approccio ibrido basato su spaCy e UDPipe, combinato con regole personalizzate e interventi manuali. Questo sistema si basa su un'analisi comparativa degli output generati dai due strumenti di NLP, con ulteriori correzioni specifiche implementate all'interno della piattaforma 4D.
Il processo inizia con la preparazione del testo, che può essere fornito in formato TXT o XML-TEI. Nel caso di file XML, LiotroConcord_v2 include un modulo dedicato all'estrazione del contenuto dal tag <body>, escludendo automaticamente le sezioni ritenute non rilevanti, grazie a un sistema di esclusione basato su tag e attributi. Una volta estratto, il testo viene normalizzato attraverso una serie di trasformazioni che includono l'uniformazione dei caratteri di newline, la rimozione degli spazi ridondanti o superflui e la sostituzione di eventuali apici o virgolette non standard con caratteri uniformi. La normalizzazione garantisce un input coerente e privo di elementi che potrebbero interferire con l'analisi linguistica.
La tokenizzazione viene realizzata all'interno della piattaforma 4D mediante espressioni regolari altamente personalizzate. Questo approccio permette di identificare con precisione specificità linguistiche dell'italiano, come parole contenenti enclitiche o apostrofi sia iniziali che finali, che spesso non vengono trattate adeguatamente dai sistemi NLP standard. Un esempio è dato dalla parola 'Ntoni, presente nel testo I Malavoglia. Entrambi i sistemi, spaCy e UDPipe, tendono a ignorare l'apostrofo iniziale, che segnala l'aferesi, riconoscendo il token semplicemente come Ntoni. Questa interpretazione non è corretta, poiché l'apostrofo è parte integrante del nome e ne modifica la struttura morfologica.
Dopo la tokenizzazione, il testo viene analizzato separatamente sia da spaCy[3] sia da UDPipe.[4] Ciascun sistema elabora i token, assegnando loro un lemma e una categoria grammaticale (POS), e gli output generati vengono salvati in file distinti per consentire un confronto diretto.

```
if __name__ == "__main__":
    import argparse

    parser = argparse.ArgumentParser(description="Analizza un file di testo con SpaCy e UDPipe.")
    parser.add_argument("file_path", help="Percorso al file di testo da analizzare.")
    parser.add_argument("udpipe_model_path", help="Percorso al file modello UDPipe.")
    parser.add_argument("language_code", help="Codice lingua (es. 'it' o 'en').")
    args = parser.parse_args()

    analyze_text(args.file_path, args.udpipe_model_path, args.language_code)
```

Figura 1. Entry point dello script Python: esegue l'analisi di un file di testo utilizzando SpaCy e UDPipe, leggendo i parametri dalla riga di comando

Il software confronta quindi i risultati dei due sistemi. Quando entrambi concordano sul lemma e sulla categoria grammaticale, LiotroConcord_v2 accetta automaticamente l'output come valido. Tuttavia, nei casi in cui spaCy e UDPipe producono risultati discordanti, oppure quando il lemma non è presente nel vocabolario di riferimento interno a LiotroConcord_v2, il software, come vedremo nel capitolo successivo, interviene applicando regole personalizzate per risolvere le ambiguità o adattare i risultati al modello di riferimento.

3 Il modello utilizzato è it_core_news_sm

4 Il modello utilizzato è italian-isdt-ud-2.5

4. ERRORI DEI SISTEMI E METODI DI CORREZIONE

Un esempio di adattamento è la forma "del" che in questo modello viene ricondotta unicamente al lemma "di", senza esplicitare la combinazione articolata. Questo approccio differisce da quello adottato da spaCy e UDPipe: per spaCy, "del" è lemmatizzato come "di il ADP", mentre UDPipe separa ulteriormente il token in "di ADP"[5] e "il DET".[6] Per rispettare il modello di riferimento, LiotroConcord_v2 traduce automaticamente l'analisi di spaCy e UDPipe nella rappresentazione consolidata del lemma "di".
Il software effettua anche controlli specifici su differenze semantiche sottili ma sostanziali: se un verbo è preceduto da un articolo, il sistema interpreta la combinazione come un infinito sostantivato.
Come dicevamo, il sistema considera non plausibili i risultati che includono lemmi assenti dal vocabolario di riferimento, come nel caso delle enclitiche o, più in generale, delle forme o dei lemmi inesistenti.
Nel caso del token "tramutandotela", entrambi i sistemi restituiscono un lemma errato: "tramutandotelare". Questo errore dimostra la difficoltà di segmentare correttamente i componenti morfologici, in quanto né spaCy né UDPipe riescono a riconoscere i singoli lemmi "tramutare", "te" e "la". Al contrario, LiotroConcord_v2, grazie all'implementazione di una regola personalizzata che riconosce le enclitiche come le particelle pronominali o avverbiali, scompone correttamente il token nei suoi elementi, associando ciascuno di essi al lemma e alla categoria grammaticale corrette.

```
ARRAY TEXT($arrEnclitiche;17)
$arrEnclitiche{1}:="glie"
$arrEnclitiche{2}:="me"
$arrEnclitiche{3}:="mi"
$arrEnclitiche{4}:="te"
$arrEnclitiche{5}:="ti"
$arrEnclitiche{6}:="le"
$arrEnclitiche{7}:="lo"
$arrEnclitiche{8}:="li"
$arrEnclitiche{9}:="la"
$arrEnclitiche{10}:="ci"
$arrEnclitiche{11}:="ce"
$arrEnclitiche{12}:="vi"
$arrEnclitiche{13}:="ve"
$arrEnclitiche{14}:="si"
$arrEnclitiche{15}:="ne"
$arrEnclitiche{16}:="se"
$arrEnclitiche{17}:="gli"
```

```
: (([Lemmi]Categoria="sm") | ([Lemmi]Categoria="sf")\
    | ([Lemmi]Categoria="ag") | ([Lemmi]Categoria="di")\
    | ([Lemmi]Categoria="ie") | ([Lemmi]Categoria="in")\
    | ([Lemmi]Categoria="po") | ([Lemmi]Categoria="re"))
  $categoriaGiusta:="ar"
  $lemmagiusto:=""
  If ($arrSplitted2{$i-1}#"l'")
    $lemmagiusto:=Lowercase($arrSplitted2{$i-1})
  Else
    If ([Lemmi]Categoria="sm")
      $lemmagiusto:="lo"
    End if
```

Figura 2. Frammenti di codice 4D per la gestione personalizzata delle enclitiche e degli articoli

Un caso di discordanza tra spaCy e UDPipe è rappresentato dal termine "bramosìa" (in "... la vaga bramosìa dell'ignoto..." da I Malavoglia). UDPipe lo analizza erroneamente come aggettivo (bramosio), mentre spaCy lo classifica correttamente come sostantivo.
Gli esempi di output permettono di comprendere l'efficacia del sistema integrato e le difficoltà che spaCy e UDPipe incontrano nell'analisi linguistica italiana. Tuttavia, grazie al confronto con il dizionario di riferimento integrato in LiotroConcord_v2, il software conferma l'interpretazione corretta, avendo già catalogato il lemma "bramosia" e le relative forme "bramosia" e "bramosie", eliminando così ogni ambiguità.

5. UN CASO DI STUDIO: *I VICERÉ* DALL'OCR ALL'OUTPUT DELLA LEMMATIZZAZIONE IN XML-TEI

Un caso di studio utile a comprendere il *workflow* completo (cfr. *Figura 3*) del sistema di lemmatizzazione è quello dell'edizione scientifica digitale commentata de *I viceré* di Federico De Roberto, promossa all'interno del progetto [PROJECT].

[5] "ADP" sta per *Adposition*; in italiano corrisponde alle preposizioni "di", "a", "da" ...

[6] "DET" sta per *Determiner*; in italiano corrisponde a varie categorie grammaticali: articoli determinativi (es. "il", "la", "i", "le"), articoli indeterminativi (es. "un", "una"), pronomi dimostrativi usati come determinanti (es. "questo", "quello"), determinanti indefiniti (es. "alcuni", "ogni").

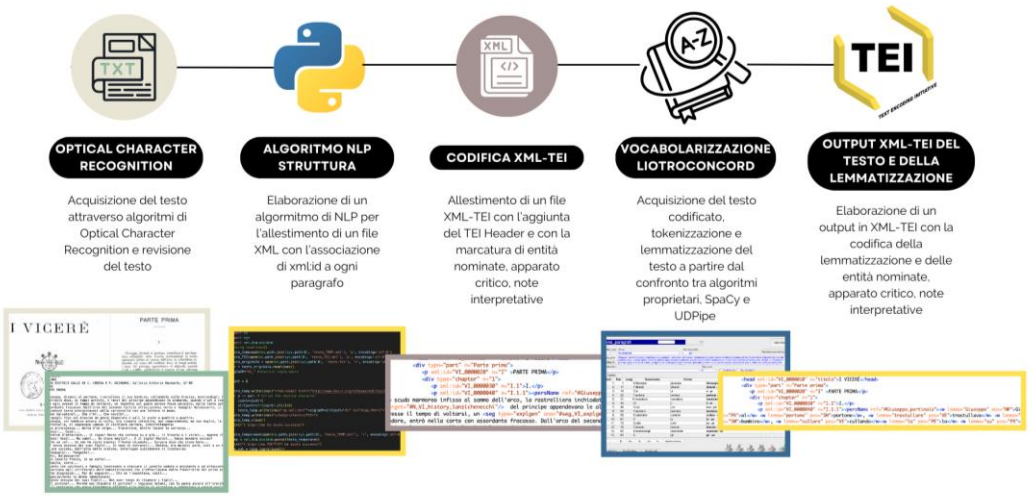

Figura 3. Workflow per la codifica e vocabolarizzazione dei Viceré

A partire dall'acquisizione delle immagini ad alta definizione dell'edizione Galli del 1894 e del conseguente riconoscimento automatico dei caratteri (OCR) è stato prodotto un documento in formato solo testo a cui è stato applicato un algoritmo di *Natural Language Processing* in Python. In particolare, sono state utilizzate le *regular expressions* e il metodo *readline* per costruire la struttura del documento XML associando un @xml:id univoco (fondamentale anche per associare le righe nel sistema di lemmatizzazione). Infine, attraverso l'API del *Document Object Model* è stato *parsato* il testo producendo il documento xml, poi marcato manualmente in linea con il modello di codifica promosso dalla *Text Encoding Initiative* (TEI). In particolare, sono state marcate le entità nominate (<persName>, <placeName>, <orgName>, <eventName>), i riferimenti a personaggi, luoghi, organizzazioni ed eventi (<rs>), alcune parole in disuso o di difficile comprensione (<distinct>), il discorso diretto (<q>), alcuni segmenti testuali a cui associare delle note di esegesi (<seg>) e infine alcune note di carattere storico-bibliografico o interpretativo (con dei *pointer* <ptr> e delle <note>). Si veda come esempio la codifica del primo paragrafo del primo capitolo:

```
<p xml:id="VI_0000040" n="I.1.1"><persName ref="#Giuseppe_portinaio">Giuseppe</persName>, dinanzi al portone, trastullava il suo bambino, cullandolo sulle braccia, mostrandogli lo scudo marmoreo infisso al sommo dell'arco, la rastrelliera inchiodata sul muro del vestibolo dove, ai tempi antichi, i <distinct ana="lanzo">lanzi</distinct><ptr type="history" target="#N_VI_history_lanzichenecchi"/> del principe appendevano le alabarde, quando s'udì e crebbe rapidamente il rumore d'una carrozza arrivante a tutta carriera; e prima ancora che egli avesse il tempo di voltarsi, un <seg type="explgen" ana="#seg_VI_explgen_legnetto">legnetto</seg> sul quale pareva fosse nevicato, dalla tanta polvere, e il cui cavallo era tutto spumante di sudore, entrò nella corte con assordante fracasso. Dall'arco del secondo cortile affacciaronsi servi e famigli: <persName ref="#BaldassarreCrimi">Baldassarre</persName>, il maestro di casa, schiuse la vetrata della loggia del secondo piano, intanto che <persName ref="#SalvatoreCerra">Salvatore Cerra</persName> precipitavasi dalla carrozzella con una lettera in mano.</p>
```

Il documento così marcato è stato acquisito in *input* da LiotroConcord che, dopo la *tokenizzazione,* ha associato il lemma e il POS corrispondente a ogni forma in base al confronto tra il modello utilizzato e quelli promossi da SpaCy e UDPipe, come detto nei paragrafi precedenti. La *Graphical User Interface* (GUI) di LiotroConcord (cfr. Figura 4) permette dunque al concordatore di intervenire correggendo gli omografi, le associazioni o *tokenizzazioni* errate, i neologismi, ecc. Un importante aspetto riguarda anche la decisione del concordatore su alcuni aspetti che necessitano di una decisione interpretativa (come la scelta tra aggettivo e participio nei participi passati, tra articolo e numerale negli indeterminativi, ecc.).

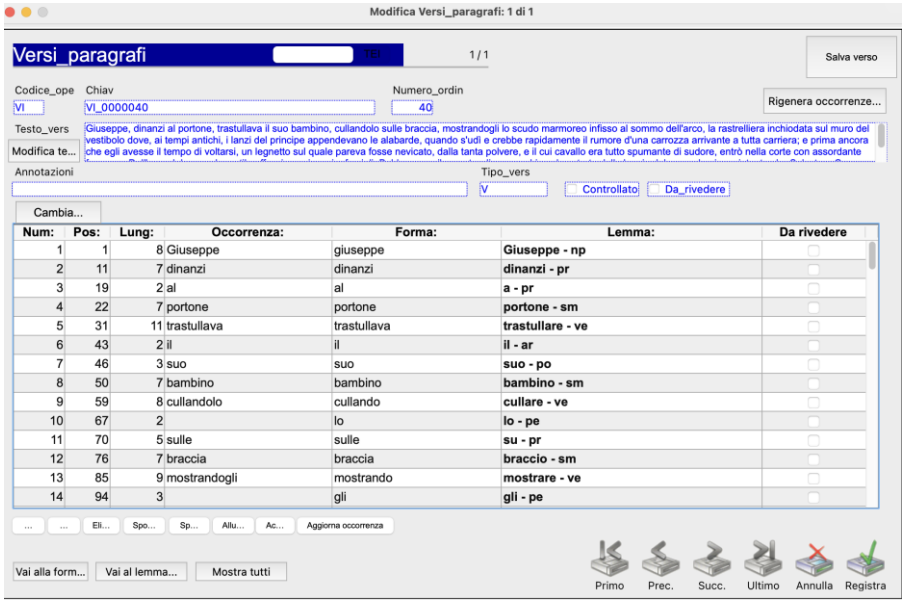

Figura 4. Schermata di LiotroConcord con la lemmatizzazione del primo paragrafo de I Viceré

Infine, la lemmatizzazione viene integrata nella codifica attraverso il tag <w> e gli attributi @lemma e @pos, così da rendere i dati aperti e interoperabili. Si veda dunque l'*output* arricchito del primo paragrafo:

```
<p xml:id="VI_0000040" n="I.1.1"><persName ref="#Giuseppe_portinaio"><w lemma="Giuseppe"
pos="NP">Giuseppe</w></persName>, <w lemma="dinanzi" pos="PR">dinanzi</w> <w lemma="a"
pos="PR">al</w> <w lemma="portone" pos="SM">portone</w>, <w lemma="trastullare"
pos="VE">trastullava</w> <w lemma="il" pos="AR">il</w> <w lemma="suo" pos="PO">suo</w> <w
lemma="bambino" pos="SM">bambino</w>, <w lemma="cullare" pos="VE">cullando</w><w
lemma="lo" pos="PE">lo</w> <w lemma="su" pos="PR">sulle</w> <w lemma="braccio"
pos="sm">braccia</w>, <w lemma="mostrare" pos="VE">mostrando</w><w lemma="gli"
pos="PE">gli</w> <w lemma="lo" pos="AR">lo</w> <w lemma="scudo" pos="SM">scudo</w> <w
lemma="marmoreo" pos="AG">marmoreo</w></p>
```

La possibilità di far dialogare realmente i dati della codifica (come quello sul discorso diretto o sulle entità nominate) e la lemmatizzazione diventa potenzialmente non soltanto un arricchimento della codifica, ma una vera e propria occasione per effettuare interrogazioni complesse con lo sviluppo di concordanze selettive (per capitoli, parti, personaggi) aprendo a un nuovo modo di intendere la vocabolarizzazione, la codifica, *distant* e *close reading* e l'applicazione ermeneutica di tali teorie e modelli.

6. CONCLUSIONI

Questo lavoro ha dimostrato come un approccio ibrido, che integra spaCy, UDPipe e il software personalizzato LiotroConcord_v2, possa affrontare efficacemente le complessità linguistiche della lingua italiana, garantendo una lemmatizzazione accurata grazie all'uso di tecnologie avanzate, combinato con regole manuali mirate.

I prossimi sviluppi prevedono l'addestramento di modelli NLP basati sulle concordanze già gestite da LiotroConcord_v2. Questo processo potrebbe migliorare ulteriormente l'accuratezza degli strumenti esistenti, affinando la capacità di spaCy e UDPipe di gestire le specificità della lingua italiana. I nuovi modelli, addestrati su un corpus così ricco e curato, potrebbero ridurre gli errori di lemmatizzazione e disambiguazione, offrendo risultati più precisi.

BIBLIOGRAFIA

Alishahi, A., Chrupała, G., & Linzen, T. (2019). Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. Natural Language Engineering, 25(4), 543–557. https://doi.org/10.1017/S135132491900024X.

Blaise, A. (2022). Do Large Language Models Understand Us? Daedalus, 151(2), 183–197. https://doi.org/10.1162/daed_a_01909.

Boleda, G. (2020). Distributional semantics and linguistic theory. Annual Review of Linguistics, 6, 213–234. https://doi.org/10.1146/annurev-linguistics-011619-030303.

Borges, J. L. (1998). The library of Babel. In Collected Fictions.

Ciotti, F. (2023). "La codifica del testo, XML e la TEI", in Digital Humanities. Metodi, strumenti, saperi. Carocci.

Ciula, A., Eide, Ø., Marras, C., & Sahle, P. (2023). Modelling between digital and humanities: Thinking in practice. Open Book Publishers. https://doi.org/10.11647/OBP.0369.

Clarke, A. C. (1967). The nine billion names of God. Harcourt.

Di Silvestro, A., D'Agata, C., Palazzolo, G., & Sichera, P. (2022). Conservazione e fruizione di banche dati letterarie: L'archivio della poesia italiana dell'Otto/Novecento di Giuseppe Savoca. In F. Ciracì, G. Miglietta, & C. Gatto (Eds.), Quaderni di Umanistica Digitale. AIUCD 2022 - Proceedings (pp. 1–12). AIUCD. https://doi.org/10.6092/unibo/amsacta/6848.

Floridi, L. (2023). AI as agency without intelligence: On ChatGPT, large language models and other generative models. Philosophical Technology, 36, 15. https://doi.org/10.1007/s13347-023-00621-y.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Cambridge: MIT Press.

Hassija, V., Chamola, V., Mahapatra, A., et al. (2024). Interpreting Black-Box Models: A review on explainable artificial intelligence. Cognitive Computation, 16, 45–74. https://doi.org/10.1007/s12559-023-10179-8.

Hewitt, J., & Manning, C. D. (2019). A structural probe for finding syntax in word representations. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4129–4138). New York: Association for Computational Linguistics.

Kumar, S., Sumers, T. R., Yamakoshi, T., Goldstein, A., Hasson, U., Norman, K. A., et al. (2023). Reconstructing the cascade of language processing in the brain using the internal computations of a transformer-based language model. bioRxiv. https://doi.org/10.1101/2022.06.08.495348.

Lodder, J. (2009). Binary Arithmetic: From Leibniz to von Neumann. https://doi.org/10.5948/UPO9780883859742.023.

Mickus, T., Paperno, D., & Constant, M. (2022). How to dissect a muppet: The structure of transformer embedding spaces. Transactions of the Association for Computational Linguistics, 10, 981–996. https://doi.org/10.1162/tacl_a_00501.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., et al. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys, 56(1), 1–40. https://doi.org/10.1145/3605943.

Palazzolo, G. (2023). Lessicografia letteraria e metodo concordanziale: Storia, modelli e sfide per le Digital Humanities. Linguistica e Letteratura, 48(1–2), 23–45. https://digital.casalini.it/10.19272/202301602006.

Primiero, G. (2020). On the Foundations of Computing. Oxford: Oxford University Press.

Savoca, G. (2000). Lessicografia letteraria e metodo concordanziale. Firenze: Olschki.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4593–4601). Somerset, NJ: Association for Computational Linguistics.

Turing, A. (1950). Computing machinery and intelligence. Mind, 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433.