

Per l'interoperabilità e la sostenibilità delle risorse digitali dantesche: il progetto LiDa

Cesare Concordia¹, Gaia Tomazzoli², Nicola Aloia¹, Carlo Meghini¹, Luca Trupiano¹

¹ Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo", CNR, Italia - *nome.cognome@isti.cnr.it*

² Sapienza Università di Roma, Italia - *gaia.tomazzoli@uniroma1.it*

ABSTRACT¹

In questo contributo presentiamo LiDa (Linking Dante), un progetto che mira a promuovere una maggiore interoperabilità e sostenibilità dei dati sulle opere dantesche raccolti in progetti precedenti tramite l'utilizzo di tecnologie e linguaggi del Web Semantico. Dopo un'introduzione in cui definiamo gli scopi del progetto (§1), facciamo cenno ai limiti con cui si scontra l'annotazione in XML dei dati linguistici e alle potenzialità di una loro implementazione in RDF (§2); introduciamo poi gli elementi fondamentali della nuova ontologia che abbiamo elaborato a tal scopo (§3), e descriviamo la procedura con cui abbiamo realizzato un nuovo grafo di conoscenza che colleghi il testo della *Commedia* alle risorse linguistiche che lo descrivono (§4); infine, presentiamo le modalità di navigazione e interrogazione di tale grafo (§5) e riflettiamo sui risultati raggiunti e sugli sviluppi futuri (§6).

PAROLE CHIAVE

Ontologie; Web Semantico; Linguistic Linked Open Data; Linguistica italiana; Letteratura italiana

1. INTRODUZIONE

Linking Dante (LiDa) è un progetto che ha l'obiettivo di digitalizzare le opere dantesche e la conoscenza a esse relativa collegando precedenti progetti e implementando nuove funzionalità, per mettere a disposizione dell'utente una piattaforma integrata che renda possibili diverse opzioni di navigazione e interrogazione. La biblioteca digitale di LiDa è basata sulle tecnologie e sui linguaggi del Semantic Web: rispetta i principi FAIR (Findable, Accessible, Interoperable, Reusable) e si ispira alle ontologie di riferimento per la rappresentazione dei dati linguistici e del patrimonio culturale; il testo di Dante diventa così un nodo inserito in una rete dove sono connesse varie risorse che permettono di approfondire le sue caratteristiche linguistiche, semantiche e intertestuali. Per raggiungere questo scopo, LiDa ha tradotto la conoscenza relativa al testo dantesco che si trova attualmente disseminata in diversi strumenti digitali e cartacei in un grafo strutturato secondo una logica calcolabile. In particolare, LiDa integra la conoscenza raccolta dai progetti DanteSearch ([9]), DanteSources ([2]), Hypermedia Dante Network ([1]) e MONT ([5]); in questo contributo ci concentriamo sul dataset di DanteSearch (§2), che abbiamo modellato tramite una nuova ontologia (§3) e collegato al testo della *Commedia* grazie a un algoritmo (§4), rendendolo navigabile e permettendo interrogazioni complesse (§5).

2. DA XML A RDF

Gli studi danteschi sono da diversi decenni all'avanguardia nel campo delle digital humanities (per una rassegna cf. [4]); uno dei progetti più pionieristici in tal senso è stato DanteSearch, una piattaforma digitale che permette la navigazione e l'interrogazione dell'intero corpus delle opere volgari e latine di Dante, che sono state lemmatizzate e annotate per descriverne la morfologia e la sintassi. Il dataset di DanteSearch è costituito da un insieme di file XML annotati secondo la codifica TEI²; ciascun file riporta una sola tipologia di annotazione: sintattica oppure morfologica; all'interno dell'annotazione sintattica sono incluse anche alcune informazioni sui dialoghi della *Commedia* (il tipo di discorso – diretto, indiretto o pensato – e il locutore). Una delle funzionalità mancanti nel sistema DanteSearch, discussa dettagliatamente dal coordinatore del progetto ([9: 609]), è la possibilità di interrogare i testi del corpus definendo contemporaneamente filtri sulla sintassi e sulla morfologia, e questo perché i file con annotazioni di tipo diverso hanno strutture XML-TEI non omogenee tra loro (cf. Fig. 3). Per implementare questa funzionalità in DanteSearch si potrebbero prospettare diverse soluzioni: si potrebbe ad esempio definire una nuova struttura XML-TEI, molto complessa, e mappare in essa le strutture delle diverse annotazioni, oppure creare un indice esterno con un tool specifico, o ancora implementare un modulo software che esegua l'integrazione a runtime, elaborando il risultato di query separate. Ciascuna di queste soluzioni rimane problematica a causa del formato di codifica adottato: XML nasce come formato per la serializzazione dei dati, cioè per la codifica delle informazioni, e per le sue caratteristiche permette di rappresentare facilmente documenti con strutture ad albero, ma richiede soluzioni specifiche per documenti in cui le associazioni tra le parti siano più complesse, come nel caso di un testo annotato. Una delle soluzioni più usate per annotare testi è la codifica XML-TEI, considerata lo «standard internazionale [...] imprescindibile per qualunque impresa di filologia digitale» ([9: 586]) e adottata, come si diceva, in DanteSearch. Benché le linee guida della TEI definiscano un modulo lightweight per l'annotazione morfologica o

¹ Corresponding author: Cesare Concordia, ultimo accesso agli URL il 18/04/2024.

² <https://dantesearch.dantenetwork.it/download.jsp>.

sintattica³ – e non siano dunque incompatibili con l’annotazione di un testo dalla struttura più complessa – i problemi a cui abbiamo accennato ne risultano solo parzialmente risolti, come dimostra il fatto che gli annotatori di DanteSearch hanno elaborato un modello di codifica ad hoc, limitando l’interoperabilità della marcatura con altri corpora.

In anni recenti si sono diffusi progetti che mirano invece a favorire l’interoperabilità tra grandi corpora di testi, e dunque la produzione di Linguistic Linked Open Data (LLOD), grazie ad annotazioni linguistiche fondate su ontologie come Ontolex⁴, elaborate in una prospettiva che non guarda primariamente al testo, ma al linguaggio più in generale, e dunque più attenta alle sue varie componenti e alle relazioni tra queste. Un’iniziativa importante in questa direzione è LiLa (Linking Latin), che si propone di connettere varie risorse linguistiche relative al latino per renderle interoperabili, e che comprende corpora annotati, lemma-banks, tree-banks per l’annotazione sintattica e altri strumenti per il NLP. L’elemento centrale dell’architettura di LiLa è però il lemma, il che permette un buon compromesso tra fattibilità e granularità ([6]: 75); il nostro approccio, invece, assegna una maggior centralità al testo, inteso come stratificazione di strutture di significato che intrattengono diverse relazioni paradigmatiche e sintagmatiche.

Il progetto LiDa ha tra i suoi obiettivi quello di rendere le opere dantesche annotate interoperabili con altri corpora. Per fare questo abbiamo in primo luogo elaborato un’ontologia che descrive il testo letterario nella sua complessa relazione di strutture di significato relative rispettivamente alla partizione dell’opera, alla morfologia e alla sintassi, e che permette di connettere ogni elemento di tali strutture a modelli e vocabolari standard. Per rappresentare le entità individuate abbiamo scelto di usare RDF, un formalismo per la definizione di modelli di dati basato sui principi del Semantic Web, che permette di creare grafi di conoscenza (knowledge graphs), cioè strutture dati in cui le informazioni sono organizzate secondo regole definite da ontologie formali, e sulle quali è possibile applicare ragionatori automatici per inferire nuova conoscenza. Sui grafi di conoscenza è possibile fare ricerche semantiche e tutte le entità sono identificate da un IRI, che le rende disponibili per essere referenziate, e dunque riutilizzate, come LLOD.

3. CONCETTI E ONTOLOGIE

Il progetto LiDa, nello specifico, si basa su un’ontologia applicativa, espressa in OWL 2 DL e chiamata Ontology of Literary Resources (OLiRes), che a sua volta si fonda sul CIDOC CRM⁵ e su Ontolex. OLiRes rappresenta il testo della *Commedia* di Dante secondo la sua struttura testuale, morfologica e sintattica, ed è interoperabile con le ontologie HDN (cf. [1]), MONT (cf. [5]), ORL (Ontologia delle Risorse Lessicali) e SyntIt (la nostra ontologia per la sintassi dell’italiano), che integrano ulteriore conoscenza rispettivamente sulla dimensione intertestuale codificata dal secolare commento, sulle metafore e su lessico, morfologia e sintassi della *Commedia*. OLiRes rappresenta la conoscenza sul testo delle opere letterarie a tre livelli: 1. il *livello dell’occorrenza*, che rappresenta le caratteristiche del testo nella sua concreta realizzazione, compresa la sua struttura (nel caso della *Commedia*, per esempio, la sua divisione in cantiche e canti); 2. il *livello linguistico*, che codifica gli aspetti linguistici del testo, come la sua morfologia e la sua sintassi; 3. il *livello concettuale*, che rappresenta la componente semantica e concettuale del testo, comprese le relazioni tra le forme che occorrono nel testo e i loro significati o tra il veicolo e il tenore di una metafora.

Questi tre livelli sono legati tra di loro da relazioni di *istanziamento* o di *occorrenza*: ogni entità linguistica è un’istanza della corrispondente entità concettuale e si manifesta in un’occorrenza; così, per fare un esempio, la forma ‘cammin’ è un’istanza del lemma ‘cammino’, che a sua volta rimanda al concetto di ‘cammino’, e occorre nel primo verso del primo canto della *Commedia*, in quarta posizione («Nel mezzo del *cammin* di nostra vita», *Inf.* I, 1). Le entità che fanno parte di uno stesso livello – dell’occorrenza, linguistico o concettuale – sono legate tra di loro da una relazione di *composizione ordinata*: una cantica è composta da una sequenza ordinata di canti, un periodo è composto da una sequenza ordinata di frasi. La relazione di composizione si lega a quella di *localizzazione*: per rappresentare le diverse entità che compongono il testo e per permetterne il reperimento tramite interrogazioni, l’ontologia rappresenta il frammento (o i frammenti) di testo in cui occorre ciascuna entità, specificandone le coordinate di inizio e fine all’interno del testo dell’opera, visto come uno spazio monodimensionale continuo e totalmente ordinato, pur con alcuni distinguo. Una determinata forma flessa, infatti, occorre sempre in un unico frammento di lunghezza 1, un determinato periodo occorre sempre in un frammento di lunghezza (almeno) 1, mentre una frase può occorrere in più frammenti se nel periodo si trova intervallata da altre frasi.

All’interno della nostra ontologia, ogni entità, qualunque sia il livello a cui appartiene, è rappresentata dalla classe **olires:Entity**, sottoclasse di **owl:Thing**; poiché nello standard che abbiamo preso a riferimento, il CIDOC CRM, esiste solo la classe **ecrm:E33_Linguistic_Object**⁶, che non distingue tra entità linguistiche e frammenti testuali,

³ <https://www.tei-c.org/release/doc/tei-p5-doc/en/html/AI.html#AILALW>.

⁴ https://www.w3.org/community/ontolex/wiki/Main_Page.

⁵ <https://www.cidoc-crm.org/>.

⁶ Secondo la Scope Note, la classe E33_Linguistic_Object del CIDOC CRM «comprises identifiable expressions in natural language or languages. Instances of E33 Linguistic Object can be expressed in many ways: e.g., as written texts, recorded speech or sign language».

abbiamo introdotto le due sottoclassi **olires:SyntacticEntity**, avente come istanze le entità linguistiche, e **olires:TextFragment**, avente come istanze le occorrenze (Fig. 1). Per rappresentare le proprietà di istanziazione e occorrenza sopra descritte abbiamo introdotto le proprietà **olires:instanceOf**, che lega ogni istanza del livello linguistico alla corrispondente istanza del livello concettuale, e **olires:occurrenceOf**, che lega ogni occorrenza alla corrispondente istanza del livello linguistico; non è stato possibile ricondurre queste due proprietà ad alcuna proprietà del CRM. Per rappresentare la proprietà di composizione, invece, abbiamo usato la proprietà **ecrm:P148_has_component**, che lega le istanze di **ecrm:E89_Propositional_Object** tra di loro e si applica quindi sia alle entità linguistiche che alle occorrenze, essendo entrambe istanze di **ecrm:E33_Linguistic_Object**, che è sottoclasse di **ecrm:E89_Propositional_Object**.

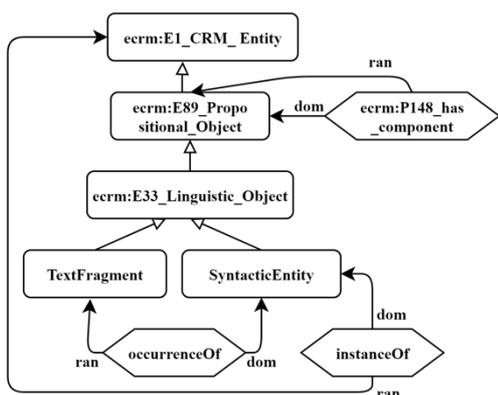


Figura 2 Entità linguistiche e testo

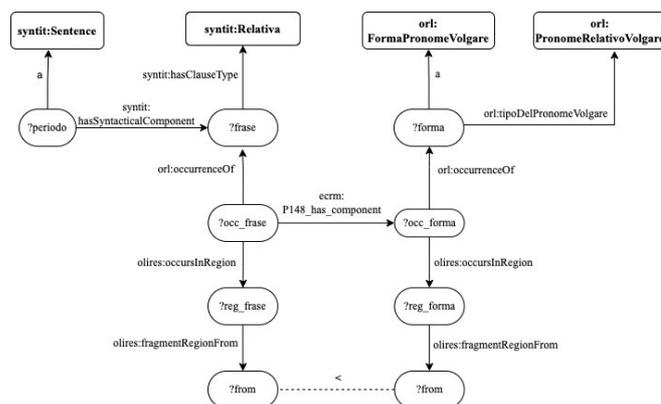


Figura 1 Es. di filtro SPARQL sul grafo della *Commedia*

Tuttavia, la proprietà **ecrm:P148_has_component** non può essere usata in alcun assioma che vincoli la proprietà a specifiche classi (ad esempio, l'assioma che stabilisce che le componenti di un'entità sintattica sono solo entità sintattiche, e viceversa) perché è dichiarata transitiva, quindi è una proprietà *composite*⁷ su cui non possono essere espressi assiomi di cardinalità, pena la violazione della *Restriction on Simple Roles*. Per questo abbiamo introdotto quattro sottoproprietà intransitive di **ecrm:P148_has_component**: **olires:hasStructuralComponent** per la composizione delle entità strutturali; **olires:hasSyntacticalComponent** per la composizione delle entità lessicali e sintattiche; **hasTenor** e **hasVehicle** per la composizione delle metafore.

Come abbiamo ricordato, l'ontologia di LiDa è OLiRes, la quale si appoggia a diverse ontologie di dominio, configurandosi come un sistema composito; le due classi **olires:SyntacticEntity** e **olires:TextFragment** giocano un ruolo fondamentale nell'integrazione delle ontologie che assiomatizzano i diversi fenomeni linguistici che formano tale sistema composito. In particolare, (1) l'ontologia Ontolex, usata per la morfologia, si lega a OLiRes tramite gli assiomi che stabiliscono che **ontolex:Form**, la classe delle forme, è sottoclasse di **olires:SyntacticEntity** e che **ontolex:lexicalForm**, la proprietà che lega una forma al suo lemma, è sottoproprietà di **ecrm:P148_has_component**; per contro, la classe **olires:FormOccurrence**, che ha come istanze le occorrenze delle forme, è sottoclasse di **olires:TextFragment**; (2) l'ontologia SyntIt, usata per la sintassi dell'italiano, si lega a OLiRes tramite gli assiomi che stabiliscono che **syntit:Sentenza**, la classe dei periodi, è sottoclasse di **olires:SyntacticEntity**, mentre **syntit:SentenzaOccurrence**, la classe delle occorrenze dei periodi, è sottoclasse di **olires:TextFragment**; questa classificazione si propaga poi alle relative sottoclassi, quella delle frasi (**syntit:Clause**) e relative occorrenze (**syntit:ClauseOccurrence**), quella dei sintagmi (**syntit:Syntagm**) e relative occorrenze (**syntit:SyntagmOccurrence**); (3) l'ontologia HDN, usata per le relazioni intertestuali, si lega a OLiRes tramite gli assiomi che stabiliscono che **hdn:Reference**, la classe più generale dei riferimenti intertestuali, è sottoclasse di **olires:SyntacticEntity**, mentre **hdn:ReferenceOccurrence**, la classe più generale delle occorrenze dei riferimenti intertestuali, è sottoclasse di **olires:TextFragment**; (4) l'ontologia MONT, usata per le metafore, si lega a OLiRes tramite gli assiomi che stabiliscono che **mont:LinguisticMetaphor**, la classe delle metafore linguistiche, è sottoclasse di **olires:SyntacticEntity**, mentre **mont:MetaphorOccurrence**, la

⁷ Secondo la terminologia introdotta nella sezione 11 (Global Restrictions on Axioms in OWL 2 DL) della specifica di OWL 2 DL (<https://www.w3.org/TR/owl2-syntax/>).

classe delle occorrenze di metafore, è sottoclasse di `olires:TextFragment`. In sostanza, `olires:SyntacticEntity` e `olires:TextFragment` agiscono da elementi cardine nell'articolazione delle ontologie che concorrono a formare la base logica di LiDa, lasciando alle ontologie specifiche appena nominate il compito di rappresentare i rispettivi domini. Questo ci permette di osservare che lo sviluppo di un'ontologia applicativa non può prescindere né dal riferimento a ontologie di dominio, né dal riferimento a un'ontologia top – nel caso di LiDa il CRM e Ontolex, che forniscono i concetti di base, con in testa la classe `ecrm:E33_Linguistic_Object`, di cui `olires:SyntacticEntity` e `olires:TextFragment` sono sottoclassi.

4. REALIZZAZIONE DEL GRAFO DI CONOSCENZA DELLA *COMMEDIA*

Una volta definita l'ontologia OLiRes, il grafo di conoscenza relativo è stato realizzato con un software che estrae ed elabora il testo e le annotazioni presenti nei file XML-TEI con le annotazioni morfologiche e sintattiche pubblicate in DanteSearch. Nei file con le annotazioni morfologiche della *Commedia*, uno per ogni cantica, ogni forma è annotata con un elemento `<LM>`. L'elemento ha due attributi, `@lemma` e `@catg`, i cui valori sono rispettivamente la forma canonica corrispondente alla forma annotata e una codifica delle caratteristiche grammaticali/morfologiche della forma annotata.

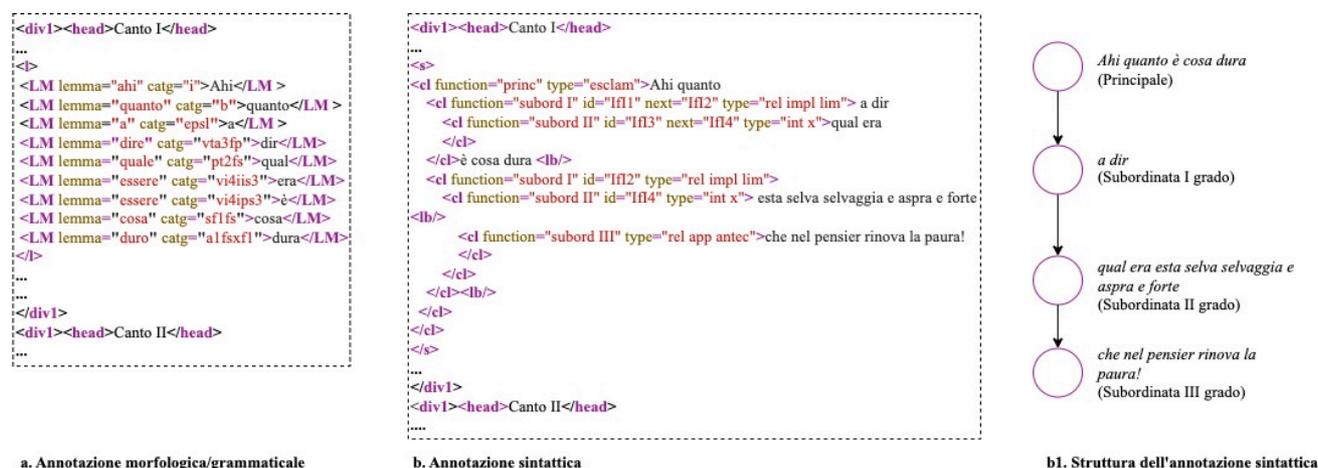


Figura 3. Esempi di annotazioni della *Commedia* in DanteSearch

Gli elementi `<div1>` delimitano le forme che appartengono a ciascun canto e gli elementi `</>` delimitano le forme appartenenti ai versi (cf. Fig. 3a: annotazione morfologica/grammaticale di *Inf.* I, 3). Il software realizzato esegue un parsing dei tre file, processa le annotazioni e genera le entità del livello linguistico e del livello dell'occorrenza per ciascuna delle forme, inserendole nel grafo di conoscenza. L'annotazione sintattica della *Commedia* realizzata da DanteSearch è basata sull'analisi del periodo ([3]): ogni canto è suddiviso in periodi, a loro volta suddivisi in frasi; i frammenti di testo che compongono una frase possono essere non consecutivi. Nel file di DanteSearch con le annotazioni sintattiche i periodi sono identificati dall'elemento TEI `<s>` (s-unit) e le frasi dall'elemento `<cl>` (clause); entrambi gli elementi possono avere l'attributo `@id` che identifica l'annotazione e l'attributo `@next` (di tipo IDREF), utilizzato per collegare in un'unica aggregazione elementi che annotano frammenti di testo non consecutivi. Il tipo sintattico⁸ e la funzione sintattica⁹ di ciascuna frase sono riportate negli attributi `@type` e `@function` dell'elemento `<cl>` che la annota. Le informazioni sui gradi di subordinazione tra le frasi di un periodo sono codificate sia nelle annotazioni sia nell'annidamento degli elementi `<cl>` (cf. Fig. 3b: annotazione sintattica di *Inf.* I, 3-6). Di seguito l'algoritmo per la creazione delle entità del grafo di conoscenza sintattico:

- Trasforma il file con le annotazioni sintattiche in una struttura dati che rappresenta un albero orientato: crea un nodo per ogni elemento TEI di annotazione; fra due nodi c'è un arco se i corrispondenti elementi sono in relazione parent-child (un livello di annidamento) nella struttura del file XML-TEI. Gli elementi che annotano una frase composta da frammenti di testo non consecutivi sono raggruppati in un unico nodo (cf. Fig. 3b1: parte dell'albero relativo al periodo di *Inf.* I, 3-6). La struttura dati ottenuta rappresenta un albero orientato, ed è possibile usare degli algoritmi di visita dell'albero per verificare la correttezza del parsing come pure per ottenere informazioni da aggiungere al grafo di conoscenza. Ad esempio, controllando la matrice delle adiacenze delle funzioni

⁸ Etichette sintetiche che rappresentano categorie desunte dalla Grande Grammatica Italiana di Consultazione ([8]).

⁹ Possibili valori sono: principale, coordinata a una principale, subordinata (I-VII grado), coordinata a una subordinata (I-V grado), parentetica, coordinata a una parentetica, pseudo-coordinata.

sintattiche delle frasi ottenuta visitando l'albero (cf. Fig. 4) sono state rilevate e corrette delle imprecisioni nelle annotazioni.

- Crea le triple delle entità linguistiche sintattiche visitando i sottoalberi che rappresentano i periodi. Parallelamente genera anche le triple del livello dell'occorrenza: per ciascun periodo/frase calcola i frammenti di testo in cui occorre e costruisce le entità del grafo di conoscenza che le rappresentano.
- Utilizzando le entità del grafo generate nella fase precedente individua le occorrenze delle forme la cui posizione è compresa nei frammenti di testo (**olires:TextFragment**) di ciascun periodo e di ciascuna frase; associa tali occorrenze a quelle dei periodi/frasi (tramite **ecrm:P148_has_component**), realizzando il collegamento tra le annotazioni morfologiche e quelle sintattiche.
- Il grafo di conoscenza della *Commedia* è memorizzato in un triple-store e accessibile tramite query SPARQL. Le condizioni di ricerca possono essere create componendo, tramite gli operatori logici, condizioni morfologiche e/o sintattiche. Inoltre, usando le proprietà dell'occorrenza di una entità linguistica, è possibile definire condizioni anche sui frammenti di testo in cui questa si manifesta.

Come esempio di query consideriamo una ricerca sulla *Commedia* descritta in [9], che può essere formulata così: “cerca le frasi relative in cui sia presente un pronome relativo che non occupi la prima posizione nella frase”. In DanteSearch questa ricerca viene eseguita solo in parte con il search engine: per essere completata richiede una fase di controllo manuale del risultato ottenuto con l'engine. Nel grafo creato in LiDa, invece, è possibile eseguire la ricerca con una singola query, definendo nel filtro di ricerca le condizioni sulla posizione nel testo delle occorrenze, sia delle forme sia delle frasi (cf. Fig. 2).

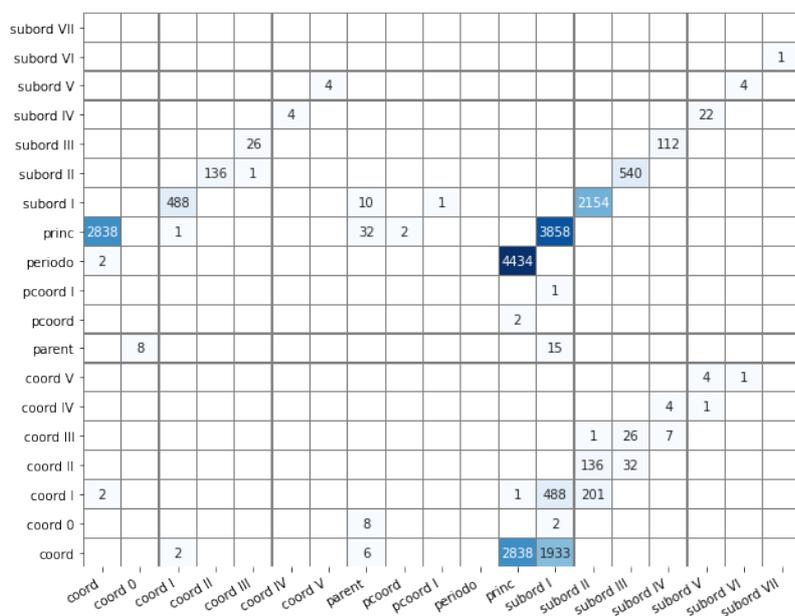


Figura 4 Adiacenze fra funzioni sintattiche

5. VISUALIZZAZIONE E RICERCA

Per permettere, con vari livelli di sofisticazione, l'accesso ai dati del grafo di conoscenza abbiamo realizzato un'interfaccia grafica (GUI) accessibile come Web Application¹⁰. La GUI fornisce funzionalità di navigazione nel testo della *Commedia* e due modalità di ricerca: semplice e avanzata.

¹⁰ <https://lida.dantenetwork.it>.

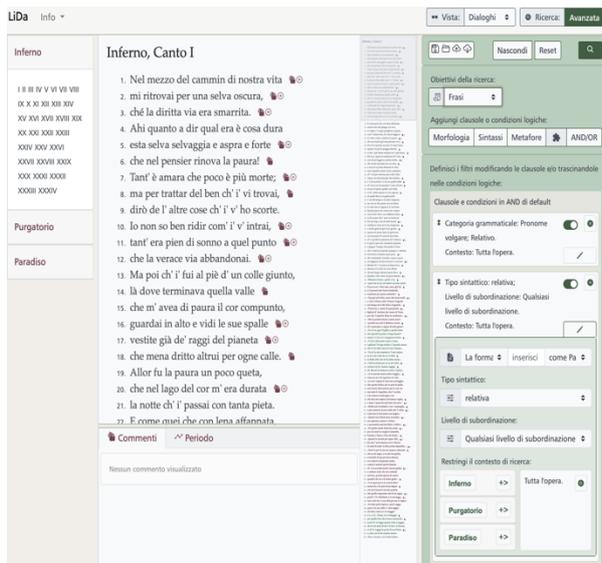


Figura 5 Visualizzazione e ricerca nel testo

La ricerca semplice permette di cercare lemmi, forme o frasi (ma anche prefissi o suffissi) sull'intera opera o su specifiche cantiche e/o canti, esprimendo opzionalmente anche altre condizioni in forma di espressioni regolari. La ricerca avanzata permette di esprimere filtri di ricerca complessi, definendo espressioni booleane i cui operandi possono essere condizioni morfologiche e/o sintattiche. Ricerca e browsing sono visualmente integrate: si può passare da una modalità all'altra senza perdita del contesto, navigando l'opera o esaminando il risultato di una ricerca. Graficamente l'interfaccia è suddivisa in tre aree sovrastate da un menù (cf. Fig. 5): a sinistra si trova un menù di navigazione tra le cantiche e i canti o tra i risultati di una ricerca; l'area di destra presenta gli strumenti per definire le query; l'area centrale contiene la visualizzazione dei dati: vi compaiono il testo del canto selezionato e altre informazioni. Quest'ultima area di visualizzazione è suddivisa orizzontalmente in due zone: nella zona superiore viene visualizzato il canto, con diverse tipologie di visualizzazione, chiamate viste:

- Vista Forme: posizionando il mouse su una forma, vengono visualizzate le informazioni morfologiche e lessicali pertinenti.
- Vista Periodi: tramite simboli e colori vengono delimitati sia i periodi in cui il canto è suddiviso, sia le frasi che compongono ciascun periodo; posizionando il mouse su una frase viene mostrata la sua funzione sintattica.
- Vista Dialoghi: vengono evidenziati i dialoghi (o discorsi) presenti nel canto; per ciascun discorso è possibile visualizzare il parlante e il tipo di discorso (diretto, riportato, pensato).

Nella parte inferiore dell'area di visualizzazione è possibile vedere i commenti relativi alle fonti primarie raccolte dal progetto HDN, facendo un clic sull'icona che compare accanto ai versi per cui esiste almeno un commento, oppure, con un doppio clic su una forma, la rappresentazione grafica, con nodi e archi, della struttura del periodo in cui essa compare.

6. CONCLUSIONI

L'attività di LiDa è ancora in corso: restano da implementare ulteriori funzionalità di ricerca per i commenti relativi alle fonti primarie annotati dal progetto HDN; le metafore di MONT devono ancora essere integrate nel grafo di conoscenza, e dunque anche la loro interrogazione e visualizzazione tramite la GUI dev'essere implementata. Dato lo scopo di LiDa, che è quello di rendere interoperabili tutti i dati che costituiscono la nostra conoscenza sul testo dantesco, il progetto potrà accogliere in futuro altre risorse: speriamo così di valorizzare il lavoro – passato, presente e futuro – della comunità degli studi danteschi, e di contribuire alla sostenibilità delle risorse digitali attraverso la loro riconversione al paradigma del Web Semantico, con i suoi modelli e linguaggi. L'integrazione tra i dati relativi a tanti progetti differenti non sarebbe stata possibile con una struttura rigida come quella delle annotazioni sintattiche e morfologiche basate sull'XML: sarebbe stato estremamente laborioso, se non impossibile, realizzare una struttura ad albero così stratificata da definire nel testo dantesco decine di migliaia di frammenti di natura e granularità diversa. Il problema non è solo quello tecnico della sovrapposizione di gerarchie (cf. [7]): un approccio che si concentri sul testo e sulla sua espressione linguistica rende evidente la varietà di strutture sintagmatiche e paradigmatiche, e dunque di unità, con cui la lingua costruisce i suoi significati. Le triple RDF costituiscono invece una struttura liquida e seguono un pattern semplice, che definisce una regione di testo e permette la sua annotazione a qualunque livello di granularità; hanno inoltre il grande vantaggio di dotare ogni singola risorsa di un IRI, che la rende disponibile per essere referenziata e dunque riutilizzata come LLOD.

BIBLIOGRAFIA

- [1] Bartalesi, Valentina et al., “A formal representation of the *Divine Comedy*’s primary sources: the Hypermedia Dante Network ontology”. *Digital Scholarship in the Humanities*, no. 37/2, 2022, pp. 630–43.
- [2] Bartalesi, Valentina, Carlo Meghini, Daniele Metilli, Mirko Tavoni, and Paola Andriani. “A web application for exploring primary sources: the DanteSources case study”. *Digital Scholarship in the Humanities*, vol. 33/4, 2018, pp. 705–23.
- [3] Gigli, Sara. “La codifica sintattica della *Commedia* di Dante”. *Sintassi dell’italiano antico e sintassi di Dante. Atti del seminario di studi* (Pisa, 15-16 ottobre 2011). Edited by Marta D’Amico, Felici, 2015, pp. 81–96.
- [4] Maselli, Matteo. “Per una rassegna degli strumenti della critica dantesca: dai repertori testuali ai dispositivi digitali”. *Paratesto*, vol. 18, 2021, pp. 299–337.
- [5] Meghini, Carlo, e Gaia Tomazzoli. “Per un’ontologia delle metafore nella *Commedia* di Dante”. *«Per intelletto umano / e per autorità»*. *Il contesto di formazione e diffusione culturale del poema dantesco. Atti del I Convegno HDN (Pisa-Firenze, 29-31 ottobre 2020)*. Edited by Leyla MG Livraghi and Gaia Tomazzoli, Cesati, 2022, pp. 127–52.
- [6] Pedonese, Giulia, Flavio Cecchini, and Marco Passarotti. “Linking the Computational Historical Semantics corpus to the LiLa Knowledge Base of Interoperable Linguistic Resources for Latin”. *Language, Data and Knowledge 2023*. Edited by Sara Carvalho et al., NOVA FCSH - CLUNL, 2023, pp. 74–85.
- [7] Renear, Allen, Elli Mylonas, and David Durand. *Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies*, 1993.
- [8] Renzi, Lorenzo, Matteo Salvi, e Anna Cardinaletti (eds.). *Grande grammatica italiana di consultazione*. Bologna: il Mulino, 1995.
- [9] Tavoni, Mirko. “DanteSearch: il corpus delle opere volgari e latine di Dante lemmatizzate con marcatura grammaticale e sintattica”. *Lectura Dantis 2002-2009. Omaggio a Vincenzo Placella per i suoi settanta anni*. Edited by Anna Cerbo, Università L’Orientale, 2011, vol. 2, pp. 583–608.