

Università degli Studi di Torino

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in Informatica



Tesi di Laurea Triennale

**Raccomandazione di contenuti
musicali: un sistema intelligente
basato sulla combinazione di
concetti**

RELATORE

Prof. Gian Luca Pozzato

CORRELATORE

CANDIDATO

Alberto Marocco

947841

Anno Accademico 2024/2025

DICHIARAZIONE DI ORIGINALITÀ

Dichiaro di essere responsabile del contenuto dell'elaborato che presento al fine del conseguimento del titolo, di non avere plagiato in tutto o in parte il lavoro prodotto da altri e di aver citato le fonti originali in modo congruente alle normative vigenti in materia di plagio e di diritto d'autore. Sono inoltre consapevole che nel caso la mia dichiarazione risultasse mendace, potrei incorrere nelle sanzioni previste dalla legge e la mia ammissione alla prova finale potrebbe essere negata.

ABSTRACT

Questo lavoro presenta un sistema intelligente di raccomandazione musicale basato sulla combinazione di concetti. Il sistema utilizza testi e caratteristiche stilistiche dei brani, acquisiti e arricchiti tramite un crawler automatico di Genius, per costruire prototipi di genere e ibridi cross-genere. La pipeline implementata comprende moduli di analisi delle ripetizioni, generazione di prototipi concettuali e un classificatore che sfrutta “anchors” e soglie adattive per selezionare i contenuti più rilevanti. L’approccio proposto coniuga trasparenza e interpretabilità, fornendo raccomandazioni spiegabili e adattabili a diversi scenari musicali.

INDICE

Introduzione	1
1.a Contesto e motivazioni	1
1.b Problema e idea	1
1.c Contributi	1
1.d Risultati in sintesi	1
1.e Organizzazione del manoscritto	1
Fondamenti teorici: typicalità e combinazione di concetti (TCL)	2
2.a Tipicità e chiusura razionale	2
2.a.I Notazione di base	2
2.b TCL: Typicality-based Compositional Logic	2
2.b.I Ruolo HEAD / MODIFIER	3
2.b.II Perché serve TCL (il “pet-fish”)	3
2.c Semantica probabilistica delle tipicità	3
2.d Il sistema CoCoS (cenni)	3
Probabilità e strumenti per la combinazione concettuale (TCL e CoCoS)	4
3.a Inclusioni probabilistiche e scenari in TCL	4
3.b CoCoS: input, algoritmo, output	4
3.c Sistemi affini e riuso di CoCoS	5
3.d Collocazione nella pipeline della tesi	5
Estrazione e pre-processing dei dati (Genius)	6
4.a Crawler e fonti	6
4.b Pulizia e normalizzazione	6
4.c Dataset finale	6
4.d Limiti dei testi	6
Creazione dei prototipi di genere	7
5.a Proprietà rigide e tipiche	7
5.b Soglie e iperparametri	7
5.c Penalità e boost	7
5.d Output dei prototipi	7

Combinazione di generi musicali con CoCoS	8
6.a Preprocessing head/modifier	8
6.b Generazione scenari	8
6.c Mixing conservativo vs aggressivo	8
6.d Coerenza e controlli	8
Classificatore e generazione di spiegazioni	9
7.a Anchors e selezione delle evidenze	9
7.b Decisione e ranking	9
7.c Spiegazioni	9
7.d Complessità e prestazioni	9
Sistema di raccomandazione	10
8.a Architettura complessiva	10
8.b Interfacce tra i moduli	10
8.c Parametri globali e riproducibilità	10
Risultati	11
9.a Riclassificazione e copertura	11
9.b Ablation su head/modifier	11
9.c Esempi qualitativi	11
9.d Osservazioni principali	11
Discussione	12
10.a Analisi criticità ed errori	12
10.b Confronto con approcci affini	12
10.c Implicazioni pratiche	12
Conclusioni e sviluppi futuri	13
11.a Conclusioni	13
11.b Sviluppi futuri	13
Bibliografia / Sitografia	15

1 INTRODUZIONE

Questa tesi propone DEGARI-Music, un sistema di raccomandazione musicale spiegabile basato su prototipi di genere e combinazione di concetti. L’obiettivo è produrre suggerimenti trasparenti, con spiegazioni legate a proprietà tipiche e rigide dei generi (e loro ibridi).

1.A CONTESTO E MOTIVAZIONI

Le piattaforme raccomandano bene ma spiegano poco. La trasparenza incide su fiducia, controllo e scoperta consapevole.

1.B PROBLEMA E IDEA

Problema: raccomandazioni “black-box”. Idea: prototipi + combinazione (head/modifier) + soglie/anchors → spiegazioni white-box.

1.C CONTRIBUTI

- Implementazione end-to-end (crawler → prototipi → combinazioni → classificatore).
- Spiegazioni leggibili basate su match proprietà↔brano.
- Valutazione: riclassificazione/copertura, ablation, esempi qualitativi.

1.D RISULTATI IN SINTESI

Breve teaser dei risultati più significativi.

1.E ORGANIZZAZIONE DEL MANOSCRITTO

Breve guida ai capitoli 4–13.

2 FONDAMENTI TEORICI: TYPICALITÀ E COMBINAZIONE DI CONCETTI (TCL)

2.A TYPICALITÀ E CHIUSURA RAZIONALE

Le Logiche Descrittive (DL) forniscono un formalismo per rappresentare concetti, ruoli e individui. Nelle DL classiche (ad es. ALC) il ragionamento è monotono: aggiungere nuova informazione non invalida ciò che è già derivabile. Questo è inadeguato per conoscenza “di buon senso” ricca di eccezioni.

Per modellare regolarità con eccezioni si introduce l'operatore di tipicità $T(c \cdot)$: un'assioma del tipo $T(C) \sqsubseteq D$ indica che “tipicamente i C sono D ”. Le inclusioni **rigide** conservano la forma classica $C \sqsubseteq D$.

La **rational closure** estende alle DL la chiusura razionale: i concetti vengono ordinati per grado di eccezionalità e si adottano **modelli minimi** che minimizzano i ranghi di anomalia. In questo modo le inferenze su $T(c \cdot)$ sono conservative, ma consentono di sospendere premesse tipiche quando emergono informazioni più specifiche (ereditarietà difettibile).

2.A.I NOTAZIONE DI BASE

- Inclusioni rigide: $C \sqsubseteq D$.
- Inclusioni tipiche: $T(C) \sqsubseteq D$.
- Preferenza semantica: si privilegiano interpretazioni che minimizzano l'eccezionalità (modelli minimi) e rispettano la gerarchia di specificità.

2.B TCL: TYPICALITY-BASED COMPOSITIONAL LOGIC

TCL combina tipicità, probabilità e combinazione concettuale in stile cognitivo. Le proprietà tipiche sono annotate con un grado:

$$p :: T(C) \sqsubseteq D, p \in (0.5, 1] \quad (1)$$

che si legge: “con grado p , i C tipici sono D ”.

Dato un **HEAD** C_H e un **MODIFIER** C_M , la combinazione produce un concetto composto che non è la semplice intersezione, ma il risultato di una selezione coerente delle proprietà:

1. **Generazione di scenari:** si costruiscono insiemi compatibili di inclusioni tipiche (rispettando tutte le rigide).
2. **Valutazione:** ad ogni scenario si associa un punteggio/probabilità ottenuto combinando le annotazioni delle tipicità considerate attive (ipotesi d'indipendenza sui pesi tipici).
3. **Selezione:** si scelgono gli scenari ammessi e meglio valutati; le proprietà del risultato sono quelle presenti nello/gli scenario/i selezionato/i.

2.B.I RUOLO HEAD / MODIFIER

- L'HEAD fornisce la struttura concettuale di base; in caso di conflitti tra tipicità, hanno priorità le rigide e, tra le tipiche, si privilegiano scelte coerenti con la “fisionomia” dell'HEAD.
- Il MODIFIER introduce restrizioni/aggiustamenti che possono soppiantare tratti non essenziali dell'HEAD se questo aumenta la coerenza globale dello scenario.

2.B.II PERCHÉ SERVE TCL (IL “PET-FISH”)

La combinazione concettuale umana non è additiva: esistono congiunzioni in cui la tipicità dell'insieme composto supera quella dei componenti (es. **pet fish**). Approcci puramente fuzzy o intersezioni rigide non catturano questi effetti; l'uso di tipicità difettibili, modelli minimi e scenari pesati consente invece di selezionare combinazioni plausibili senza contraddizioni.

2.C SEMANTICA PROBABILISTICA DELLE TIPICITÀ

Le annotazioni p sulle inclusioni tipiche possono essere interpretate in modo “distribuzionale”: uno scenario eredita un punteggio combinando (sotto ipotesi d'indipendenza) i pesi delle tipicità attive e penalizzando quelle escluse o in conflitto. Questa lettura consente di ordinare gli scenari e di spiegare perché alcune combinazioni risultano più plausibili di altre dal punto di vista qualitativo e quantitativo.

2.D IL SISTEMA CoCoS (CENNI)

CoCoS implementa il calcolo di scenari di TCL. Dati HEAD e MODIFIER con le rispettive proprietà rigide e tipiche, costruisce gli scenari ammissibili, ne calcola il punteggio e restituisce il/i prototipo/i del concetto composto come mappa proprietà \rightarrow grado, insieme alle informazioni sullo scenario selezionato. (Questa sezione introduce solo l'idea operativa; i dettagli implementativi saranno trattati nei capitoli successivi.)

3 PROBABILITÀ E STRUMENTI PER LA COMBINAZIONE CONCETTUALE (TCL E CoCoS)

In questo capitolo approfondiamo (i) la componente probabilistica di TCL e (ii) il tool CoCoS che implementa la combinazione concettuale basata su tipicità, presentando anche sistemi affini riutilizzatori di CoCoS.

3.A INCLUSIONI PROBABILISTICHE E SCENARI IN TCL

In TCL le proprietà tipiche sono annotate con un grado di credenza:

$$p :: T(C) \sqsubseteq D, \quad p \in (0.5, 1] \quad (2)$$

. L'idea è che le tipicità valgano “per i C normali”, ma possano essere sospese di fronte a informazione più specifica. La semantica probabilistica (stile DISPONTE) assume indipendenza fra tipicità e definisce una distribuzione sugli **scenari**, cioè sulle scelte di quali inclusioni tipiche sono attive.

Definizione operativa. Data una base con tipicità $p_i :: T(C_i) \sqsubseteq D_i$:

- uno **scenario** è un sottoinsieme consistente delle tipicità;
- allo scenario si associa un punteggio/probabilità ottenuto combinando i p_i (attive) e $(1 - p_i)$ (escluse);
- si selezionano gli scenari **ammessi** dalle rigide e **coerenti** con l'euristica HEAD/MODIFIER; il/i prototipo/i del concetto composto eredita/no le proprietà presenti nello/gli scenario/i selezionato/i.

Questa lettura si integra con la **rational closure** di ALC+TR: gli scenari sono valutati solo sui modelli che minimizzano l'eccezionalità, preservando specificità e irrilevanza.

3.B CoCoS: INPUT, ALGORITMO, OUTPUT

Scopo. CoCoS implementa TCL: dato un concetto HEAD C_H e un MODIFIER C_M , costruisce gli scenari ammissibili e restituisce il prototipo del concetto composto $C_H^l \wedge C_M$.

Input.

- **Proprietà rigide** di C_H e C_M (vincoli non derogabili).
- **Proprietà tipiche** con grado p per ciascun concetto.

- (Opzionale) limiti al numero massimo di proprietà da ereditare.

Algoritmo (sketch).

1. **Genera scenari:** combina le tipicità in insiemi consistenti con le rigide.
2. **Valuta:** assegna a ogni scenario un punteggio/probabilità (ipotesi di indipendenza).
3. **Filtra HEAD/MODIFIER:** scarta scenari incompatibili con la dominanza semantica dell'HEAD.
4. **Seleziona:** sceglie lo/gli scenario/i migliore/i; produce una mappa **proprietà** \rightarrow **grado** per il prototipo composito e annota la probabilità/score di scenario.

Output. Un file **Result** con la mappa delle proprietà ereditate (e.g., `{rock: 0.95, high_repetition: 0.8, ...}`) e le info di scenario; opzionale **Scenario** con la maschera delle tipicità attive.

3.C SISTEMI AFFINI E RIUSO DI CoCoS

La stessa combinazione basata su TCL è stata riusata in diversi prototipi per creatività computazionale e XAI (ad es. supporto alla generazione di personaggi/ruoli, raccomandazione culturale). Esempi recenti includono estensioni di DEGARI e strumenti come **EDIFICA** e **GOCCIOLA**, che si appoggiano a CoCoS per la fase di combinazione.

3.D COLLOCAZIONE NELLA PIPELINE DELLA TESI

Nei capitoli successivi useremo TCL/CoCoS come “motore” di combinazione: dopo la costruzione dei prototipi di base (proprietà rigide/tipiche), combineremo HEAD/MODIFIER per ottenere concetti ibridi; questi prototipi composti alimenteranno classificazione e ranking.

4 ESTRAZIONE E PRE-PROCESSING DEI DATI (GENIUS)

Descriviamo raccolta testi/metadata via Genius, pulizia e normalizzazione. Notiamo assunzioni e limiti legati all'uso dei testi.

4.A CRAWLER E FONTI

Pipeline: richieste API/scraping controllato, gestione rate-limit, campionamento per genere.

4.B PULIZIA E NORMALIZZAZIONE

Rimozione markup, lowercasing, gestione ripetizioni/chorus, token/POS (se usato).

4.C DATASET FINALE

Tabella con brani per genere, media lunghezza testo.

4.D LIMITI DEI TESTI

Ambiguità, linguaggio figurato, lingue diverse. Implicazioni sui prototipi.

5 CREAZIONE DEI PROTOTIPI DI GENERE

Definiamo proprietà rigide vs tipiche per ciascun genere, soglie e pesi; introduciamo le penalità/boost per differenziare i profili.

5.A PROPRIETÀ RIGIDE E TIPICHE

Criteri di selezione da tag/keyword/lemmi.

5.B SOGLIE E IPERPARAMETRI

typical_thr, rigid_thr, min_df_words, topk_typical, max_rigid. Razionale delle scelte.

5.C PENALITÀ E BOOST

COMMON_PENALTY, DISTINCTIVE_BOOST,
DISTINCTIVE_MAX_GENRES: effetto intuitivo.

5.D OUTPUT DEI PROTOTIPI

Formato e esempi (estratti di 2–3 generi).

6 COMBINAZIONE DI GENERI MUSICALI CON CoCoS

Come generiamo ibridi (head/modifier), quali scenari produciamo e come variamo il mixing.

6.A PREPROCESSING HEAD/MODIFIER

Criteri per assegnare head e modifier alle coppie di generi.

6.B GENERAZIONE SCENARI

Parametri, numero di combinazioni, esempi (rap-pop, metal-trap, ecc.).

6.C MIXING CONSERVATIVO VS AGGRESSIVO

Cosa cambia in termini di proprietà ereditate; cenno agli switch nel codice.

6.D COERENZA E CONTROLLI

Rimozione proprietà incompatibili, verifica minima copertura.

7 CLASSIFICATORE E GENERAZIONE DI SPIEGAZIONI

Descriviamo il classificatore basato su anchors e soglie adattive e come generiamo spiegazioni white-box per ogni raccomandazione.

7.A ANCHORS E SELEZIONE DELLE EVIDENZE

Come scegliamo gli “ancoraggi” (feature salienti) per brano e per prototipo/ibrido.

7.B DECISIONE E RANKING

Criteri di punteggio, gestione pareggi, filtri minimi.

7.C SPIEGAZIONI

Formato spiegazioni (proprietà/keyword matchate), esempi concreti per 1–2 brani.

7.D COMPLESSITÀ E PRESTAZIONI

Note pratiche: tempi medi, caching, dimensione dataset.

8 SISTEMA DI RACCOMANDAZIONE

Vista end-to-end della pipeline: dal dato grezzo alla raccomandazione con spiegazione.

8.A ARCHITETTURA COMPLESSIVA

Schema dei moduli e flusso dati.

8.B INTERFACCE TRA I MODULI

Formati I/O: dove salviamo prototipi, scenari, risultati.

8.C PARAMETRI GLOBALI E RIPRODUCIBILITÀ

Seed, configurazioni, gestione ambienti/variabili (es. token GENIUS via env).

9 RISULTATI

Presentiamo i risultati quantitativi e qualitativi, con esempi di raccomandazioni spiegate.

9.A RICLASSIFICAZIONE E COPERTURA

Metriche per genere e per ibrido; eventuali heatmap/tabelle.

9.B ABLATION SU HEAD/MODIFIER

Impatto sulla qualità/riclassificazione rimuovendo l'euristica.

9.C ESEMPI QUALITATIVI

2–3 raccomandazioni con relativa spiegazione.

9.D OSSERVAZIONI PRINCIPALI

Sintesi di cosa funziona e cosa sorprende.

10 DISCUSSIONE

Interpretiamo i risultati, evidenziamo punti di forza/debolezza e confrontiamo con la letteratura.

10.A ANALISI CRITICITÀ ED ERRORI

Dove il sistema fallisce e perché (dati, soglie, ambiguità del testo).

10.B CONFRONTO CON APPROCCI AFFINI

Cosa aggiunge il paradigma prototipi+combinazione rispetto a baseline o sistemi simili.

10.C IMPLICAZIONI PRATICHE

Trasparenza, controllabilità, uso per discovery e creazione playlist ibride.

11 CONCLUSIONI E SVILUPPI FUTURI

Ricapitoliamo contributi e risultati e indichiamo le direzioni successive.

11.A CONCLUSIONI

Che cosa abbiamo dimostrato e con quali limiti.

11.B SVILUPPI FUTURI

- Integrazione di feature audio e metadata strutturati.
- Estensione multilingua.
- Lessici d'intensità (es. VAD) e aspetti temporali del brano.
- Studio utente più ampio / A/B test sulle spiegazioni.

BIBLIOGRAFIA / SITOGRAFIA

- [1] *UniTO Typst Template*. (2024). [Online]. Disponibile su: <https://github.com/eduardz1/UniTO-typst-template>
- [2] «Typst — A new markup-based typesetting system». [Online]. Disponibile su: <https://typst.app/>
- [3] Alberto Marocco, *DEGARI-Music*. (2025). [Online]. Disponibile su: <https://github.com/albymar01/DEGARI-Music>
- [4] Alberto Marocco, *Tesi-UniTO (manoscritto)*. (2025). [Online]. Disponibile su: <https://github.com/albymar01/Tesi-UniTO>
- [5] «Genius». [Online]. Disponibile su: <https://genius.com/>
- [6] «Scrapy». [Online]. Disponibile su: <https://docs.scrapy.org/en/latest/>
- [7] «NLTK — Natural Language Toolkit». [Online]. Disponibile su: <https://www.nltk.org/>
- [8] «TreeTaggerWrapper». [Online]. Disponibile su: <https://treetaggerwrapper.readthedocs.io/>
- [9] Helmut Schmid, «TreeTagger». [Online]. Disponibile su: <https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>
- [10] A. Valse, «CoCoS: uno strumento per la combinazione di concetti», 2020.
- [11] «scikit-learn». [Online]. Disponibile su: <https://scikit-learn.org/>
- [12] «pandas». [Online]. Disponibile su: <https://pandas.pydata.org/>
- [13] «NumPy». [Online]. Disponibile su: <https://numpy.org/>
- [14] «Matplotlib». [Online]. Disponibile su: <https://matplotlib.org/>
- [15] Peter Gärdenfors, «Concept Combination and Prototypes», 2004.
- [16] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, e Clifford Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.

RINGRAZIAMENTI

Desidero esprimere la mia sincera gratitudine al Prof. Gian Luca Pozzato per la sua guida e supporto durante lo sviluppo di questa tesi. Un ringraziamento speciale va anche ai miei amici e familiari per il loro incoraggiamento costante.