

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 Prototipi, typicalità e proprietà	2
2.b Logica di typicalità (ALC + T)	2
2.c Probabilità e forza delle proprietà	2
2.d Euristica Head/Modifier	2
2.e Perché è utile nella raccomandazione	2
Probabilità per la combinazione: il framework TCL e gli strumenti	3
3.a Inclusioni probabilistiche nella combinazione	3
3.b Il tool CoCoS	3
3.c Sistemi affini (DENOTER, NERVOUS, ecc.)	3
3.d Collegamento alla pipeline di questa tesi	3
Estrazione e pre-processing dei dati (Genius)	4
4.a Crawler e fonti	4
4.b Pulizia e normalizzazione	4
4.c Dataset finale	4
4.d Limiti dei testi	4
Creazione dei prototipi di genere	5
5.a Proprietà rigide e tipiche	5
5.b Soglie e iperparametri	5
5.c Penalità e boost	5
5.d Output dei prototipi	5
Combinazione di generi musicali con CoCoS	6
6.a Preprocessing head/modifier	6
6.b Generazione scenari	6

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

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)

Presentiamo le basi logiche e cognitive: prototipi, typicalità, inclusioni probabilistiche, euristica head/modifier. Spieghiamo perché questi strumenti sono adatti alla raccomandazione.

2.A PROTOTIPI, TYPICALITÀ E PROPRIETÀ

Concetti con proprietà tipiche (frequentanti) e rigide (vincoli).

2.B LOGICA DI TYPICALITÀ (ALC + T)

Nozioni chiave: eccezioni, non-monotonicità controllata.

2.C PROBABILITÀ E FORZA DELLE PROPRIETÀ

Intuizione su “quanto è tipica” una proprietà; collegamento a soglie nel sistema.

2.D EURISTICA HEAD/MODIFIER

Come si decide cosa ereditare nella combinazione.

2.E PERCHÉ È UTILE NELLA RACCOMANDAZIONE

Trasparenza, controllo, possibilità di creare ibridi “sensati”.

3 PROBABILITÀ PER LA COMBINAZIONE: IL FRAMEWORK TCL E GLI STRUMENTI

Approfondiamo l'aspetto probabilistico della combinazione e gli strumenti impiegati (CoCoS e progetti affini), mettendo in relazione teoria e pratica.

3.A INCLUSIONI PROBABILISTICHE NELLA COMBINAZIONE

Intuizione: $p :: T(C) \sqsubseteq D$.

3.B IL TOOL CoCoS

Scopo, input/output, ruolo nel nostro flusso.

3.C SISTEMI AFFINI (DENOTER, NERVOUS, ECC.)

Breve rassegna per mostrare continuità con la letteratura.

3.D COLLEGAMENTO ALLA PIPELINE DI QUESTA TESI

Dove intervengono p , head/modifier e CoCoS nei nostri capitoli successivi.

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.