# Automating Course Articulation
## A Deep Metric Learning Framework Using Public Data

A Thesis submitted to the faculty of

San Francisco State University

In partial satisfaction of the

requirements for

the Degree

Master of Science

in

Data Science and Artificial Intelligence

by

Mark S. Kim

San Francisco, California

May 2025

# Certification of Approval

I certify that I have read Automating Course Articulation: A Deep Metric Learning Framework Using Public Data by Mark S. Kim and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirement for the degree Master of Science at San Francisco State University.

_____

Hui Yang, Ph.D

Professor

Thesis Committee Chair

_____

Arno Puder, Ph.D

Professor

_____

Anagha Kulkarni, Ph.D

Professor

# Abstract

The manual process of determining course equivalency is a significant barrier to student mobility and educational equity, leading to substantial credit loss and delayed graduation that disproportionately harms underrepresented students. Previous automated approaches have been limited by a reliance on sensitive student enrollment records or the operational intractability and opacity of using large language models for direct classification.

This thesis introduces and validates a novel framework that overcomes these challenges by decoupling semantic representation from classification, using only publicly available course catalog data. The methodology leverages deep metric learning to fine-tune contextual embedding models on public course text. These specialized embeddings are then used to construct a novel composite distance vector, which serves as a rich feature set for training traditional machine learning classifiers.

Evaluated on a real-world dataset, the proposed framework achieves state-of-the-art accuracy, with $F_1$-scores exceeding 0.99. The result is a computationally efficient, scalable, and privacy-preserving solution that provides institutions with a practical tool to automate course articulation, reduce administrative burden, and foster a more equitable educational ecosystem.

# Acknowledgments

I would like to express my deepest appreciation to my advisors, Professors Hui Yang, Arno Puder, and Anagha Kulkarni. Their patience, support, and confidence in me were invaluable throughout my graduate journey at SFSU. I also owe a special debt of gratitude to Professor Tao He, whose crucial suggestion to incorporate a global similarity metric with the embedding vectors significantly improved the model's classification performance.

This research was made possible through the contributions of many individuals. I'd like to thank the Program Pathways Mapper (PPM) team, which includes representatives from the Kern Community College District, the Foundation for California Community Colleges, and the California Community Colleges Chancellor's Office, for their partnership in providing the foundational data for this work. I'd like to recognize Natalie Yam, Parth Panchal, and Joanne Park for their assistance with data collection, compilation, and preliminary analysis.

This work was supported in part through the Platform for Open Learning, Academic Research, & Innovative Scientific computing (POLARIS) High-Performance Computing (HPC) cluster at San Francisco State University. We acknowledge these resources, services, and the support provided by the Academic Technology Systems Team.

On a personal note, I am immensely thankful for my dear friends who have been there for me through thick and thin, cheering me on whenever I felt I could not continue. To my good friend, Phil, I offer my deepest appreciation for his incredible generosity and support, which made navigating the financial challenges of graduate school possible. To Julie and

Bita, I am profoundly thankful for their invaluable guidance and perspective, which were instrumental to my personal growth and well-being throughout this journey. To my brother Nick, I am eternally grateful for his quiet strength and unwavering support during times when distance and understanding meant everything.

To all who stood by me with patience, kindness, and belief—I carry your support with deep gratitude.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

California's public higher education system is a titan of American academia, a complex, three-tiered structure comprising the University of California (UC), California State University (CSU), and the California Community Colleges (CCC) [14]. Collectively, these 149 colleges and universities serve nearly 2.9 million students, forming the largest public higher education system in the United States [14, 36, 8, 7]. A foundational principle of this system is the promise of student mobility, particularly the pathway from a two-year community college to a four-year university [14].

However, the mechanism designed to facilitate this movement—the process of determining course equivalency, or articulation, is a formidable, largely manual process that creates significant barriers for students [31]. At the heart of this process is the Articulation System Stimulating Interinstitutional Student Transfer (ASSIST), the state's official public repository for articulation agreements [5]. While ASSIST provides a centralized platform for

students and advisors to view established equivalencies, it is fundamentally a display for agreements that are negotiated and updated manually by articulation officers at each individual campus [4]. Given the sheer number of institutions, defining and maintaining these agreements is a task of bleak combinatorics, rendering it inefficient, slow, and inherently intractable [31]. This manual paradigm places a considerable burden on academic advisors and administrative staff, who must meticulously review course descriptions and syllabi to compare content, rigor, and learning outcomes [31]. The result is a system that struggles to keep pace with the needs of a vast and mobile student body, making California a critical case study for a problem that extends far beyond its borders.

The challenges exemplified by California's system are a microcosm of a systemic problem in American higher education [6]. Transferring between institutions has become a normative component of the modern student's academic journey [1]. Data from the National Student Clearinghouse Research Center reveals that in the fall of 2023, transfer enrollment constituted 13.2% of all continuing and returning undergraduates [11]. This trend represents a post-pandemic resurgence in student mobility, with transfer enrollment growing by 5.3% from Fall 2022 to Fall 2023 and an additional 4.4% in Fall 2024 [11, 12]. This mobile population is increasingly diverse, comprising not only students following traditional two-year to four-year pathways but also a substantial number of returning learners [13]. Over half of these returning students opt to re-enroll at a new institution, underscoring the critical role of the transfer system in providing flexible pathways to degree completion [13].

The consequences of this systemic inefficiency are borne almost entirely by students,

manifesting in significant academic and financial setbacks [44]. The most direct outcome is the loss of earned academic credit. A 2017 report by the U.S. Government Accountability Office (GAO) estimated that students who transferred between 2004 and 2009 lost an average of 43% of their credits [44]. This finding is echoed across studies, with reports indicating that more than half of all transfer students lose some credits, and approximately one-fifth are forced to repeat courses for which they have already received a passing grade [1].

This credit loss creates a cascade of negative consequences. It invariably increases the time-to-degree, delaying graduation and entry into the workforce [44]. Each repeated course also carries a financial cost, increasing the total tuition burden and potentially exhausting a student's eligibility for federal financial aid [44]. A process often undertaken to save money can paradoxically result in a greater overall financial commitment [6]. The frustration inherent in the transfer process also has a measurable impact on student persistence, with transfer students tending to have lower graduation rates than their non-transfer peers [35].

This issue transcends mere administrative inefficiency and becomes a critical matter of educational equity [6]. Low-income students and students from historically underrepresented racial and ethnic groups are more likely to begin at community colleges and rely on transfer pathways [43]. Recent growth in transfer enrollment has been driven disproportionately by Black and Hispanic students [11]. Therefore, the barriers imposed by an inefficient articulation system disproportionately harm the very student populations that institutions are striving to support. A clear and troubling feedback loop emerges: the manual nature of course articulation causes credit loss, which imposes academic and financial burdens that

fall most heavily on underrepresented students, undermining institutional goals of closing equity gaps. The seemingly low-level administrative task of determining course equivalency is thus revealed to be a significant driver of systemic inequity in higher education.

Addressing this challenge through robust automation is not merely an operational optimization; it is a necessary intervention to foster a more equitable, efficient, and supportive educational ecosystem. This thesis confronts this problem by developing and validating a novel computational framework to automate course articulation. By leveraging deep metric learning on publicly available course catalog text, this research introduces a privacy-preserving and scalable method that decouples semantic representation from classification.

The primary contributions of this work are the development of a highly accurate automated framework, an innovative feature engineering technique that improves classification, and a computationally efficient approach that avoids the privacy and scalability issues of previous methods. The remainder of this thesis will detail the methodology, experiments, and results of this approach.

## 1.1 Thesis Roadmap

The remainder of this thesis is structured to provide a comprehensive account of this research.

- **Chapter 2: Background and Related Work** will provide a detailed in-depth survey of the landscape of student transfer automation and the evolution of technological

interventions.

- **Chapter 3: Methodology** will offer a deep dive into the data collection and preparation processes, the specific embedding models evaluated, the construction of the feature vectors, and the theoretical underpinnings of the machine learning classifiers employed.

- **Chapter 4: Experimental Setup and Results** will detail the experimental design, the datasets used for training and validation, and a comprehensive analysis of the classification performance, including ablation studies and model comparisons.

- **Chapter 5: Discussion and Future Work** will interpret the results in a broader context, discuss the limitations of the current study, and outline promising avenues for future research, including the development of a full-scale course recommendation system and the exploration of fine-tuning techniques.

- **Chapter 6: Conclusion** will summarize the key findings of the thesis and reiterate the significance of its contributions to both academic research and the practical administration of higher education.

# Chapter 2

# Background and Related Work

The significant challenges of manual course articulation, detailed in the previous chapter, have prompted a range of research efforts aimed at automating the process [28, 34, 30, 23, 45]. These technological interventions have evolved in sophistication, mirroring broader advancements in natural language processing (NLP) and machine learning [41]. A critical review of this literature reveals a clear trajectory from simple statistical methods to complex deep learning models, with each stage introducing new capabilities while also exposing new limitations [30]. This evolution illuminates the path toward a more robust and scalable solution.

## 2.1   Keyword and Statistical Methods

The earliest attempts at automating course comparison relied on foundational text analysis techniques that, while computationally simple, lack semantic depth. The most basic systems are essentially search engines or databases that depend on exact keyword matching or pre-populated tables of known equivalencies [25]. These systems are inherently brittle; they cannot recognize semantic variations (e.g., equating "Introduction to Programming" with "Fundamentals of Computer Science I") and require continuous manual updates to remain relevant [41].

A more advanced statistical method, Term Frequency-Inverse Document Frequency (TF-IDF), improves upon keyword matching by vectorizing documents and weighting terms based on their importance. A term's frequency within a single document (TF) is balanced against its rarity across a corpus (IDF) [2]. This allows the model to assign higher importance to distinctive terms (e.g., "calculus") and lower importance to common words (e.g., "the," "a," "is") [2]. While TF-IDF has been a workhorse for information retrieval, its fundamental limitation is a complete lack of semantic understanding [2]. Models in this class treat words as discrete, unrelated tokens; they cannot grasp that "calculus" and "differentiation" are related concepts, nor can they distinguish between different meanings of the same word.

## 2.2    Static Semantic Representations

The development of word embeddings represented the first major leap toward a true semantic understanding of course content. Models like Word2Vec and GloVe, trained on vast text corpora, learn to represent words as dense vectors where words with similar meanings are positioned closer to one another in the vector space. This innovation enabled a more nuanced comparison of texts than was possible with TF-IDF. These techniques have been applied to content-based course recommendation by creating a single vector for a course description, typically by averaging the vectors of its constituent words [32].

Despite this advancement, these models produce static embeddings where each word is assigned a single, fixed vector regardless of its context [15]. This is a significant drawback, as it fails to account for polysemy—words with multiple meanings. For instance, the word "bank" would have the same vector in "river bank" and "bank account." Furthermore, the practice of averaging all word vectors to create a document-level representation is a crude heuristic that can dilute or lose critical semantic information [37].

## 2.3    Contextual Semantic Representations

The introduction of the transformer architecture, specifically models like Bidirectional Encoder Representations from Transformers (BERT), revolutionized NLP by enabling the generation of contextual embeddings [15]. In these models, a word's vector representation is dynamically influenced by the words surrounding it, allowing the model to disambiguate

word meanings and capture a much richer semantic representation [15]. Architectures such as Sentence-BERT (SBERT) were subsequently developed to fine-tune these models to produce semantically meaningful embeddings for entire sentences or paragraphs, which can then be efficiently compared using metrics like cosine similarity [37].

## 2.4 Direct LLM Classification

A more recent evolution involves the direct application of large-scale generative models, or Large Language Models (LLMs) like GPT-4 and Gemini, for classification. As explored in preliminary work for this thesis, these models can be instructed via prompt engineering and in-context learning to perform pairwise comparisons of course descriptions and render a judgment on their equivalency. While these experiments yielded promising accuracy, they also uncovered significant practical limitations. The direct use of LLMs for this task is computationally expensive, requiring full text descriptions to be sent to a model API for every comparison. Performance is acutely sensitive to prompt phrasing, and the decision-making process is a "black box," providing a categorical output without a quantifiable similarity score. This opacity makes it difficult to rank matches, set thresholds, or provide transparent justifications. This approach is also ill-suited for handling complex one-to-many or many-to-many articulation scenarios.

## 2.5 Enrollment-Based Approaches

Parallel research efforts have leveraged different data sources entirely. A notable body of work has demonstrated that course similarity can be predicted by analyzing student enrollment data [33, 23]. Models such as *course2vec* learn embeddings from the patterns of which courses students take together, operating on the principle that courses taken in the same semester or sequence likely share a topical relationship [33]. While powerful, this behavioral approach is constrained by its reliance on large-scale, proprietary institutional datasets. This raises significant data privacy concerns and limits the model's generalizability, as it cannot be used to compare courses between two institutions with no history of student transfer between them [42]. This approach is therefore not a universal solution for the broader course articulation problem.

This review of varied approaches reveals a fundamental trade-off: as models gain semantic power, they tend to become more computationally intensive, less interpretable, or more demanding of specialized or private data. The limitations of direct LLM classification (cost, opacity) and enrollment-based methods (data privacy, limited access) point toward a gap in the existing research for a new paradigm [30, 42]. An effective solution must harness the semantic power of large pre-trained models without inheriting their operational burdens. The literature thus indicates a need for an intelligent, hybrid framework that decouples deep semantic representation from final classification. This conceptual gap forms the central motivation for the methodology proposed in this thesis.

Table 2.1: Comparative Taxonomy of Course Equivalency Determination Methods

| Approach | Key Characteristics | Data Source(s) | Semantic Capability | Strengths | Limitations |
|---|---|---|---|---|---|
| Manual Review | Human experts (advisors, faculty) compare syllabi descriptions. | Course Catalogs, Syllabi | High (Human-level) | Nuanced, context-aware, trusted by faculty. | Extremely slow, not scalable, subjective, prone to inconsistency. |
| Keyword/ TF-IDF | Bag-of-words representation, statistical term weighting. | Course Catalogs | None to Low | Simple, computationally cheap, easy to implement. | Fails to capture synonyms, context, or true semantic meaning |
| Static Embeddings (Word2Vec/ GloVe) | Pre-trained word vectors, often averaged for document representation. | Course Catalogs | Medium | Captures word-level semantics, better than TF-IDF. | Context-insensitive, averaging vectors loses information. |
| Enrollment-Based (e.g., course2vec) | Embeddings learned from student co-enrollment patterns. | Proprietary Student Records | High (Be-havioral) | Captures functional relationships between courses, highly predictive. | Requires access to sensitive private data, not generalizable, privacy concerns. |
| Direct LLM Classifica-tion | End-to-end classification using prompt engineering. | Course Catalogs | Very High | High accuracy potential, understands complex language. | Computationally expensive, "black box" opacity, prompt sensitive, no quantifiable similarity score, risk of hallucinations. |
| Proposed Method (Embed-dings + ML) | Deep contextual embeddings as features for traditional classifiers. | Course Catalogs | Very High | State-of-the-art accuracy, computationally efficient, quantifiable, uses public data only. | Relies on the quality of the pre-trained embedding model. |

Table 2.1 summarizes the primary methods for determining course transferability.

# Chapter 3

# Methodology

This chapter delineates the comprehensive and systematic methodology employed to develop and evaluate a sophisticated classification pipeline capable of determining course equivalency from domain-specific textual data. As established in the preceding chapters, the manual nature of course articulation creates significant barriers for students. Furthermore, as discussed in Chapter 2, previous automated approaches have faced significant limitations related to data privacy, scalability, and interpretability, indicating a need for a new paradigm [30, 42].

To address these challenges, the research followed an evolutionary, two-phase process. The investigation commenced with an initial, exploratory phase to assess the feasibility of using Large Language Models (LLMs) as end-to-end classifiers. The findings from this stage were informative; they demonstrated the potential of modern LLMs but also highlighted significant limitations that motivated the development of a more robust and scalable framework.

Consequently, the primary focus of this thesis is a more advanced, decoupled methodology that utilizes deep embedding models as sophisticated feature extraction engines.

This chapter details this entire methodological journey. It begins by describing the initial direct LLM classification approach and the findings that motivated the subsequent pivot. It then details the final pipeline's core components: the data corpora, the feature engineering process, the selection and fine-tuning of deep embedding models, and the suite of downstream classifiers. Finally, by outlining the complete evaluation framework, including performance metrics, statistical analyses, and error analysis methodologies, this chapter sets a clear and rigorous stage for the discussion of results in Chapter 4.

## 3.1 Phase 1: Direct LLM Classification Approach

The research commenced with an exploratory phase designed to determine the feasibility of leveraging Large Language Models (LLMs) as end-to-end classifiers for the course equivalency task. This direct approach was a pragmatic first step, conceived as an initial benchmark to establish a baseline for performance using a small, manually curated dataset. At this stage of the research, a larger, more comprehensive dataset from the Program Pathways Mapper (PPM) was anticipated but not yet available, making a focused, smaller-scale investigation the most logical starting point. This methodology treated the LLM as a holistic reasoning engine, tasked with performing the entire classification from raw text input to a final equivalency judgment without intermediate feature engineering.

## Initial Data Corpus and Pre-processing

The dataset for this initial evaluation was constructed to represent a challenging, real-world scenario using publicly available data. The process began by identifying five required lower-division courses for the Computer Science major at San Francisco State University (SFSU). Using ASSIST, articulation agreements were found for these courses across 63 different California public colleges and universities. The raw course data was then manually collected from the online course catalogs of each respective college. This data consisted of the full, unmodified text including department codes, course numbers, titles, descriptions, and all associated metadata such as prerequisites, unit counts, and grading options. This approach was deliberately chosen to ensure that the analysis could compare the effectiveness of classification using the complete raw text versus more structured, extracted information.

The initial dataset consisted of 228 equivalent course pairs based on the articulation agreements. To create a more robust dataset for binary classification, this set was expanded by assuming symmetry and transitivity for course equivalency, which generated a total of 5,660 equivalent pairs. An equivalent number of non-equivalent pairs was then generated by randomly pairing courses from different subjects. From this expanded corpus of over 11,000 pairs, a final stratified random sample of 400 pairs (200 equivalent and 200 non-equivalent) was created to serve as the evaluation set for the models.

## Model Selection and Prompt Engineering

An preliminary review of various LLMs was conducted to assess their ability to reliably generate structured data from the raw course descriptions. While many open-source and proprietary models were tested, Google's PaLM2 and its successor, Gemini Pro v1.0, were ultimately selected for this phase of the research. This decision was based on their accessibility via a free-tier API and, most importantly, their consistent ability to produce well-formatted, structured data from the unprocessed text.

A systematic, iterative prompt engineering process, illustrated by Figure 3.1, was employed to develop effective prompts for both the extraction of structured data (like course topics) and the final equivalency classification. This process involved starting with simple prompts and gradually refining them based on established design principles from natural language processing research and community guides [46, 24, 38]. The final prompts for data extraction were highly structured, consisting of five parts: a preamble summarizing the task, the raw course data, specific formatting instructions, a JSON model schema defining the desired output, and a postamble with additional clarifying instructions. Despite this careful refinement, the structured data extraction process, particularly for deducing course topics, remained a significant challenge and was prone to occasional contextual errors.
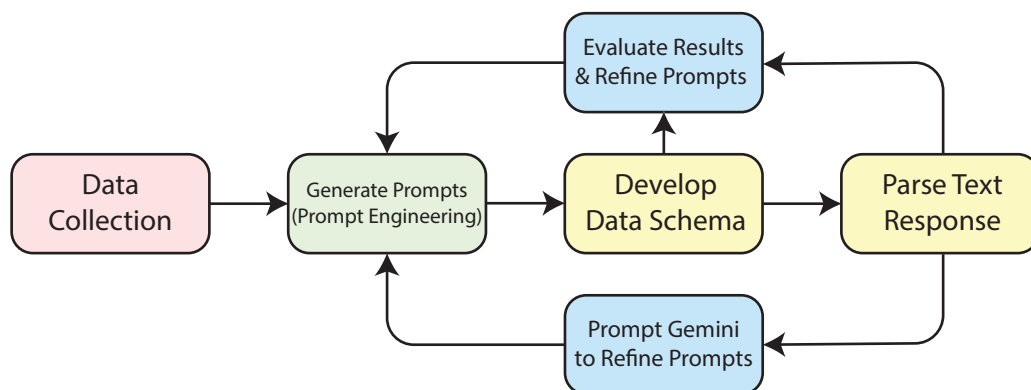
Figure 3.1: Prompt Engineering Process

## 3.2   Phase 2: The Decoupled Pipeline Framework

The limitations identified in the initial study—including high computational cost, lack of quantifiable similarity scores, and prompt sensitivity—directly motivated a pivot to a more sophisticated, decoupled pipeline. This new approach was first developed and prototyped using the initial, manually-curated dataset. This section details this final, more robust methodology, beginning with the larger and more comprehensive data corpus upon which it was ultimately trained and validated.

### The PPM Corpus and Data Preparation

The limitations of the initial study necessitated not only a more sophisticated methodology but also a larger, more comprehensive dataset to robustly train and evaluate it. This required data corpus was procured at a later stage of the research in partnership with the Program Pathways Mapper (PPM). The acquisition of this dataset was a critical step that enabled the full-scale implementation and validation of the decoupled pipeline.

The corpus provided by the PPM initially contained 2,217 courses, each labeled with a Course Identification Numbering System (C-ID) code that serves as the ground truth for course equivalency. To prepare this data for a robust, stratified partitioning, a critical filtering step was applied first. All C-ID classes with fewer than four associated courses were removed from the dataset. This step was essential to guarantee that after splitting the data, both the training and subsequent test sets would contain enough examples to form at least one equivalent course pair for every class, a necessary condition for the fine-tuning process. This filtering resulted in a final, clean corpus of 2,157 courses distributed across 157 distinct C-ID classes.

To ensure a rigorous and unbiased evaluation of the final models, this final dataset was partitioned into two distinct, non-overlapping subsets: a training set and a test set. A stratified 50/50 split was employed, using the C-ID code as the stratification key. This resulted in a training set of 1,078 courses and a test set of 1,079 courses. The stratification ensures that each of the 157 classes is represented in both subsets. The test set is held in reserve and used only once for the final, conclusive evaluation of the optimized classification pipeline, providing an honest estimate of the model's generalization performance on unseen data.

## Input Document Normalization

In a departure from conventional NLP pipelines, this research deliberately eschewed standard text pre-processing techniques such as lowercasing, stop-word removal, or the stripping of special characters. This decision was made to more accurately simulate a real-world use case where input data may be imperfectly formatted. The methodology, therefore, relies on the inherent semantic power and robustness of modern transformer-based embedding models to interpret and handle this "raw" text.

Instead of cleaning the text, a normalization step was performed to create a consistent, structured input document for the embedding models. A new field, "Formatted Course Info," was generated for each course by concatenating four key pieces of information: the department name, the department course number, the course title, and the full course description. This concatenated string serves as the single document representation for each course and is the direct input for the document embedding process, ensuring all relevant textual context is preserved in a standardized format.

## Feature Engineering

The high-dimensional embedding vectors generated by the models, while semantically rich, are not the final features used for classification. To prepare the data for the downstream classifiers, a two-stage feature engineering pipeline was executed. This pipeline first applies various dimensionality reduction techniques and then constructs pairwise difference vectors

from these embeddings to explicitly represent the relationship between two courses.

**Dimensionality Reduction**

The high-dimensional vectors produced by modern embedding models can present challenges for downstream machine learning algorithms, a phenomenon often referred to as the "curse of dimensionality." To address these potential issues, a systematic process of dimensionality reduction was applied. This investigation was motivated by several potential benefits, including improving model generalization by removing redundant or noisy dimensions and reducing computational complexity.

To ensure the integrity of the evaluation and prevent any form of data leakage, the reduction models were governed by a strict protocol. Each reduction technique was fit exclusively on the training data. The same fitted model was then used to transform both the training and the held-out test sets. This methodology guarantees that no information from the test set influences the parameters of the reduction models. A comprehensive exploration was conducted to determine the optimal dimensionality, including using Principal Component Analysis (PCA) to reduce vectors to the number of components required to explain 70%, 80%, and 90% of the original variance, as well as reducing to static 4 and 7 dimensions using PCA, t-SNE, and PaCMAP.

**Composite Distance Vector**

The ultimate goal of this research is to classify pairs of courses, which requires input features that represent the relationship between them. To provide a richer, more discriminative representation than a single metric alone, we designed a composite feature vector, $\Delta_c$. The vector is constructed by concatenating the element-wise difference of the two course embedding vectors ($\mathbf{A}$ and $\mathbf{B}$) with their cosine similarity:

$$\Delta_c = \left( a_1 - b_1, \ldots, a_k - b_k, \frac{\mathbf{A} \cdot \mathbf{B}}{\| \mathbf{A} \| \| \mathbf{B} \|} \right)$$

where $\mathbf{A} = (a_1, \ldots, a_k)$ and $\mathbf{B} = (b_1, \ldots, b_k)$ are the $k$-dimensional embedding vectors for the two courses. This design is powerful because it provides the subsequent classifier with two distinct types of information simultaneously. The element-wise difference captures granular, dimension-specific (local) disparities between the two semantic representations, while the cosine similarity provides a single, normalized measure of their overall (global) alignment in the vector space. A custom `CoursePairGenerator` class, found in the project's codebase, was implemented to systematically generate these feature vectors for all positive and negative pairs across all embedding and reduction variations.

## 3.3 Model Architecture and Training

A central hypothesis of this research is that a generic, pre-trained deep embedding model can be adapted to produce highly specialized and semantically rich embeddings for the

specific domain of course descriptions. These bespoke embeddings are expected to provide a more discriminative feature representation for downstream classification tasks compared to off-the-shelf models. This section details the architecture, learning objective, and training protocol used to achieve this adaptation through a process of deep metric learning, as well as the downstream classifiers used to evaluate the resulting features.

## Embedding Models: Selection and Fine-Tuning

The selection of an appropriate embedding model is a critical first step that influences the entire downstream pipeline. Rather than committing to a single model, this research began with a broad preliminary analysis of a variety of open-source embedding models to identify strong candidates for a more in-depth, comparative study.

### Model Selection

The initial models reviewed, summarized in Table 3.1, spanned a wide range of parameter sizes and characteristics. To screen these models efficiently, the simple but effective cosine similarity accuracy metric was used: for a given anchor course, a model was considered correct if the cosine similarity to an equivalent course was greater than the similarity to a non-equivalent course. This initial screening revealed that while performance varied, many models achieved high accuracy. Based on these results and a desire to evaluate a representative spectrum of model sizes, three models were selected for the primary analysis:

Table 3.1: Initial Embedding Model Review

| Model Name | Rank* | Params† | Dims | Acc |
|---|---|---|---|---|
| GIST-small-Embedding-v0 | 41 | 33 | 384 | 0.9759 |
| bge-small-en-v1.5 | 47 | 33 | 384 | 0.9670 |
| GIST-Embedding-v0 | 33 | 109 | 768 | 0.9768 |
| bge-base-en-v1.5 | 35 | 109 | 768 | 0.9732 |
| gte-base-en-v1.5 | 31 | 137 | 768 | 0.9732 |
| mxbai-embed-large-v1 | 24 | 335 | 1024 | 0.9759 |
| gte-large-en-v1.5 | 21 | 434 | 1024 | 0.9777 |
| multilingual-e5-large-inst | 34 | 560 | 514 | 0.9670 |
| stella_en_1.5B_v5 | 3 | 1543 | 8192 | 0.9857 |
| SFR-Embedding-2_R | 4 | 7111 | 4096 | **0.9839** |
| Agte-Qwen2-7B-instruct | 5 | 7613 | 3584 | 0.9804 |
| nvidia/NV-Embed-v2 | 1 | 7851 | 4096 | 0.9831 |

* Huggingface Overall Leaderboard Rank

† in Millions

- `BAAI/bge-small-en-v1.5` (BGE): Representing a high-performing small model.

- `avsolatorio/GIST-Embedding-v0` (GIST): Representing a medium-sized model.

- `nvidia/NV-Embed-v2` (NVE): Representing a large-scale model.

At a later stage of the research, `Salesforce/SFR-Embedding-2_R` (SFR) was also included for additional comparison due to its strong performance on public leaderboards. The foundational architecture for all these models is the transformer, which allows them to generate rich, contextual embeddings suitable for feature extraction.

**Metric Learning with Batch Triplet Loss Functions**

To determine if performance could be further improved, the BGE model was subjected to a fine-tuning process using a metric learning approach. Metric learning aims to learn an

embedding space where the geometric distance between samples corresponds to their seman-

tic similarity. The concept of Triplet Loss, first introduced in the context of face recognition,

has been widely applied to supervised similarity learning [47]. It operates on (Anchor, Pos-

itive, Negative) triplets. The fundamental goal is to train the embedding function, $f(x)$,

to map inputs into a vector space where the distance between an anchor sample $(A)$ and a

positive sample $(P)$ is smaller than the distance to a negative sample $(N)$ from a different

class, enforced by a margin, $\alpha$. The mathematical formulation is given by:

$$L(A, P, N) = \max(d(f(A), f(P)) - d(f(A), f(N)) + \alpha, 0).$$

Here, $d$ represents the Euclidean distance. The max() component ensures that loss is only

incurred for triplets that violate the margin constraint. A naive "offline" implementation of

Triplet Loss that forms all possible triplets is computationally infeasible. A more effective

approach is "online" mining, where informative triplets are selected on-the-fly from within

each mini-batch. Recognizing that different triplet mining strategies can significantly impact

model performance, this research empirically evaluated all four primary batch-based triplet

loss implementations available in the sentence-transformers library:

- **BatchAllTripletLoss**: Computes the loss for all valid triplets within a batch. While

   comprehensive, the learning signal can be diluted by the high number of "easy" triplets.

- **BatchSemiHardTripletLoss**: Considers only semi-hard triplets, where the negative

   sample is more distant than the positive but still violates the margin. This is a common

   strategy that balances stability and learning effectiveness.

- **BatchHardTripletLoss**: Implements a more aggressive strategy, using the hardest positive and hardest negative for each anchor. This can accelerate convergence but can also be "temperamental" and lead to a noisy optimization landscape.

- **BatchHardSoftMarginTripletLoss**: A variation of `BatchHardTripletLoss` that does not require a manually specified margin parameter, simplifying hyperparameter tuning.

**Optimization and Training Protocol**

The successful fine-tuning of a model with a challenging objective like `BatchHardTripletLoss` is critically dependent on the choice of optimizer and learning rate schedule. The components selected for this research were not chosen in isolation but as parts of a cohesive, synergistic framework designed to ensure stable and effective learning.

The optimization of the model's weights was performed using the **AdamW** optimizer. AdamW improves upon the standard Adam optimizer by decoupling the weight decay from the gradient update step. This ensures a more stable and effective regularization, which is crucial for preventing the model from overfitting, particularly given the noisy gradients that can arise from hard-negative mining [26].

The optimizer was paired with the `CosineAnnealingWarmRestarts` learning rate schedule. This schedule combines two powerful concepts: cosine annealing, which smoothly decays

the learning rate following a cosine curve, and "warm restarts," where the rate is periodically reset to its initial maximum [17]. This technique allows the optimizer to escape suboptimal local minima it may have settled into and explore other regions of the complex loss landscape, increasing the likelihood of finding a higher-quality solution [27].

Finally, the data loading and batching strategy was intrinsically linked to the fine-tuning objective. A crucial constraint of the triplet loss functions is that the training data must contain a minimum of two examples for each class label to form a valid (anchor, positive) pair [21]. To mitigate the risk of creating mini-batches that fail this constraint through simple random sampling, this research leveraged the `GroupByLabelBatchSampler` from the `sentence-transformers` library. This sampler ensures that each mini-batch is constructed by grouping samples with the same label, thereby guaranteeing every batch contains the necessary class diversity to form valid and informative triplets for all anchors [22].

## Downstream Classification Models

To determine the most effective method for classifying the generated pairwise feature vectors, a broad suite of machine learning algorithms was evaluated. The process began with a comprehensive initial evaluation of eight different models to understand which algorithmic families were best suited to the data. This initial suite included:

- **Linear Models (Logistic Regression, Ridge, Lasso)**: To establish a baseline and test for linear separability.

- **Instance-Based Model (k-Nearest Neighbors)**: To probe the local structure and clustering of the feature space.

- **Kernel-Based Model (Support Vector Machine)**: To test for complex, non-linear decision boundaries.

- **Ensemble Model (Random Forest)**: For its robustness and ability to capture complex feature interactions.

- **Probabilistic Models (LDA and QDA)**: To test assumptions about the geometric distribution of the data.

Based on the preliminary results from this comprehensive evaluation, four models were selected for the final, in-depth analysis due to their consistently strong performance: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and XGBoost. XGBoost, a powerful gradient boosting implementation, was added at this stage to include a state-of-the-art boosting algorithm known for its high performance on structured data, ensuring the final comparison was robust and comprehensive.

## 3.4 Evaluation Framework and Metrics

A rigorous, multi-stage evaluation framework was designed to systematically narrow down the optimal combination of embedding models, dimensionality reduction techniques, and classifiers, and to provide a comprehensive view of the entire pipeline's performance. This

section details the design of that framework, the metrics used to measure performance, and the statistical methods planned for the final analysis.

## Multi-Stage Evaluation Design

The evaluation process was designed as a funnel, beginning with a broad preliminary review to efficiently narrow the field of candidates, followed by a more focused and rigorous analysis on the larger, more comprehensive PPM Corpus.

### Stage 1: Initial LLM Feasibility and Benchmark

The process began with a broad analysis to assess the feasibility of using a Large Language Model for direct classification and to contextualize its performance. This stage, conducted on the Initial Dataset, involved two parts:

1. **Feasibility Study**: The primary evaluation tested Google's Gemini Pro v1.0 across different input data types (full raw text vs. extracted topics) and three classification schemes (binary, 3-way, and 4-way) to understand its capabilities and limitations.

2. **Comparative Benchmark**: To contextualize the results, a comparative analysis was conducted against a suite of other prominent open-source LLMs using the same binary classification task.

**Stage 2: Feature Vector Validation (Ablation Study)**

A critical ablation study was designed to validate the efficacy of the novel composite distance vector, $\Delta_c$. This involved training the full suite of initial classifiers on two different versions of the feature set derived from the Initial Dataset: one using the complete composite vector and one using an ablated vector containing only the element-wise differences. By comparing the performance across these two conditions, it was possible to isolate and measure the contribution of the global cosine similarity feature.

**Stage 3: Evaluation During Fine-Tuning**

With the larger, more robust PPM dataset, the most promising non-proprietary embedding models were fine-tuned to specialize them for the course equivalency task. To monitor the quality of the learned embeddings during this process, the `BinaryClassificationEvaluator` from the `sentence-transformers` library was employed. At the end of each training epoch, the evaluator was run on binary-labeled course pairs generated from the training set. The primary metric monitored was Average Precision based on cosine similarity, and the model checkpoint that achieved the highest score on these validation pairs was saved as the best model for that fine-tuning run.

**Stage 4: Final Downstream Classifier Evaluation**

The final and definitive evaluation was conducted on the held-out test portion of the PPM dataset. This stage used the feature vectors generated from the best-performing embedding

models (both the original off-the-shelf versions and the newly fine-tuned versions). These feature sets were then used to train and evaluate the final selection of high-performing classifiers: KNN, Random Forest, SVM, and XGBoost. This ensures an unbiased assessment of the complete pipeline's ability to generalize to new, unseen data. To find the optimal hyperparameters for each classifier, an exhaustive, brute-force hyperparameter grid search was conducted with 5-fold cross-validation.

## Performance Metrics

To facilitate a comprehensive and multi-faceted analysis, a suite of standard evaluation metrics was used to assess the various pipeline configurations across two critical dimensions: classification efficacy and computational efficiency.

### Classification Efficacy

The core assessment of classification performance is based on a standard suite of metrics derived from the confusion matrix, which tabulates the counts of True Positives ($TP$), True Negatives ($TN$), False Positives ($FP$), and False Negatives ($FN$). While Accuracy, defined as $\frac{TP+TN}{TP+TN+FP+FN}$, provides a general overview of correctness, it can be insufficient for capturing the nuances of a classifier's behavior. Therefore, this research also evaluates:

- **Precision**: Calculated as $\frac{TP}{TP+FP}$, this metric measures a model's exactness. High precision indicates that when the model predicts an equivalency, it is likely to be

correct.

- **Recall**: Calculated as $\frac{TP}{TP+FN}$, this metric measures a model's completeness. High recall indicates that the model is effective at identifying the full set of all true equivalencies.

The $F_1$-**Score**, the harmonic mean of Precision and Recall ($2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$), is used as the primary metric for reporting and comparing the final classification performance. The $F_1$-Score provides a single, robust value that balances the trade-off between precision and recall, making it an ideal metric for this task where both avoiding false equivalencies and identifying all true ones are important.

**Efficiency Metrics**

Beyond classification efficacy, the practical viability of each pipeline was assessed by measuring its computational efficiency. Both **Training Time** and **Inference Time** were systematically captured during the hyperparameter tuning process using the detailed statistics provided by `scikit-learn`'s `GridSearchCV` utility. Training time reflects the resources required to fit a model configuration, offering insight into the cost of experimentation. Inference time, reported by `GridSearchCV` as `score_time`, measures the time required for a trained model to make predictions on new data. For the use case of this research, inference time is considered the more critical efficiency metric, as it directly impacts the system's real-world responsiveness and scalability in a production environment.

**Statistical Analysis Plan**

To determine if observed differences in mean performance between models were statistically significant, a formal hypothesis testing procedure was planned for the analysis of the held-out test data. A one-way Analysis of Variance (ANOVA) was planned to first test the null hypothesis that all four finalist classifiers have equal mean $F_1$-scores. A statistically significant result would justify a post-hoc analysis to identify which specific pairs of classifiers differ. Prior to this, the assumption of homogeneity of variances would be checked using Levene's Test (with the Brown-Forsythe correction). This test is critical, as a violation of this assumption would make standard post-hoc tests like Tukey's HSD inappropriate. In the event of unequal variances, the Games-Howell post-hoc test was selected for its robustness, as it does not assume equal variances. This ensures a methodologically rigorous comparison.

## 3.5 Misclassification Analysis Methodology

While the quantitative evaluation presented in the preceding sections establishes the performance of the pipeline based on aggregate metrics like the $F_1$-score, a purely numerical analysis is insufficient for a complete understanding of the system's behavior. High-level metrics are essential for benchmarking and model selection, but they obscure the underlying nature of a model's failures. They quantify what the performance is but fail to explain why and how errors occur. Relying on such metrics alone can be misleading, as they may overestimate a model's robustness while hiding significant, systematic failure modes [18].

A critical step in the development of reliable and trustworthy AI systems is a thorough and systematic error analysis that moves beyond aggregate scores to diagnose specific weaknesses. This section, therefore, outlines the methodology for a qualitative diagnosis of the system. The objective is to perform a systematic error analysis on the misclassifications produced by the embedding models to understand the root causes of their failures. This investigation follows a two-pronged methodology.

## Shared Misclassifications Across Models

To diagnose the source of model errors, it is first necessary to determine whether they are idiosyncratic to individual models or systematic across them. If misclassifications are largely unique to each model, the errors are likely attributable to model-specific factors, such as architectural differences or suboptimal fine-tuning. Conversely, a significant overlap in misclassified pairs across multiple, diverse models would suggest the presence of systematic errors—failures that stem not from the models but from the data itself. Such errors often arise from annotation artifacts, inherent ambiguity in the source text, or insufficient information to support a clear classification, and they represent a fundamental challenge for any model applied to the dataset.

The first prong of the analysis, therefore, examines these systematic error patterns by analyzing the overlap of misclassified course pairs between the different embedding models. This approach, using visualizations such as Venn diagrams and intersection bar charts, helps

distinguish between model-specific weaknesses and challenges that are inherent to the dataset itself.

## Qualitative Case-Based Review

The second prong of the analysis is a granular, qualitative review of specific false positive and false negative examples drawn from the held-out test and cross-validation reports. This manual, case-based review is a cornerstone of effective error analysis, enabling the identification of specific artifacts and conceptual flaws that quantitative metrics cannot reveal. This is particularly important for understanding failures in complex semantic tasks. For example, it allows for the investigation of challenges in short-text semantic similarity, where a lack of rich context inherently increases ambiguity and the likelihood of spurious matches [3].

The overarching goal of this two-pronged methodology is to generate a structured report of the primary error types. This includes identifying and categorizing the root causes of both False Positives (e.g., high topical overlap without true equivalence) and False Negatives (e.g., semantic divergence in descriptions or data quality issues). This diagnostic approach is designed to inform future improvements to the system by revealing where the pipeline is most likely to fail and why.

# Chapter 4

# Experimental Setup and Results

This chapter presents the empirical results that validate the decoupled, deep metric learning framework proposed in Chapter 3. The central objective is to systematically assess the performance of each component of the pipeline—from the choice of embedding model and the impact of fine-tuning to the effects of dimensionality reduction and the selection of a downstream classifier—to identify the optimal configuration for both classification accuracy and computational efficiency.

The analysis is structured to first establish the validity of the core feature engineering approach before proceeding through the primary evaluation sequence. The chapter begins by presenting a critical ablation study to validate the efficacy of the novel composite distance vector ($\Delta_c$), the cornerstone of the feature representation. With the feature vector's design validated, the investigation then establishes a baseline by assessing off-the-shelf models, quantifies the performance gains achieved through fine-tuning, and analyzes the trade-offs

between accuracy and efficiency. The chapter culminates in a definitive comparative analysis on the held-out test data, synthesizing all prior findings to identify the single best-performing pipeline configuration. We begin by outlining the experimental setup that forms the foundation for all subsequent results.

## 4.1 Experimental Environment & Datasets

### Computational Environment

The research presented in this thesis was conducted using a hybrid computational environment, leveraging both cloud-based services for initial language model evaluations and a powerful on-premise high-performance computing (HPC) cluster for the primary, computationally intensive experiments. The initial direct classification tasks were performed using Google's Gemini v1.0, a proprietary cloud-based Large Language Model. All subsequent stages, including model fine-tuning and downstream classifier evaluations, were executed on San Francisco State University's "POLARIS" High Performance Compute Cluster. The POLARIS cluster runs on Rocky Linux 8.9 with the Slurm Workload Manager. GPU-intensive deep learning tasks were performed on a node equipped with four NVIDIA A100 GPUs, while extensive hyperparameter grid searches for traditional machine learning classifiers were run on a CPU cluster with AMD EPYC 9534 CPUs. The entire experimental pipeline was implemented in Python, with the core deep learning components built using PyTorch and the

Table 4.1: Summary of Datasets Used in Evaluation

| Characteristic | Initial Dataset | PPM Corpus |
|---|---|---|
| **Source** | Manually curated via ASSIST | Program Pathways Mapper (PPM) |
| **Purpose** | Preliminary screening, prototyping, and initial classifier evaluation | Definitive fine-tuning and final pipeline evaluation |
| **Ground Truth** | Established articulation agreements | Course Identification Number (C-ID) |
| **Final Size** | 400 course pairs (for evaluation set) | 2,157 courses (across 157 classes) |
| **Partitioning** | Stratified random sample | Stratified 50/50 train/test split |

Hugging Face ecosystem and the classical machine learning experiments conducted using `scikit-learn` and `XGBoost`.

## Datasets

As detailed in Chapter 3, this research utilized two distinct datasets at different stages of the evaluation. A smaller **Initial Dataset**, manually curated from the ASSIST repository, was used for prototyping and the preliminary screening of models and classifiers. The definitive experiments were conducted on the larger **PPM Corpus**, provided by the Program Pathways Mapper (PPM). This corpus was used for the definitive fine-tuning and final pipeline evaluation, with its held-out test set reserved for an unbiased assessment of the optimized pipelines to ensure methodological rigor. Table 4.1 provides a summary of the key characteristics of both datasets.

**Evaluation Metrics**

To facilitate a multi-faceted analysis, a standard suite of metrics was used to assess both classification efficacy and computational efficiency. The core assessment of classification performance was measured using the $F_1$**-Score**, the harmonic mean of Precision and Recall. This metric was chosen because it provides a single, robust value that balances the trade-off between avoiding false equivalencies (precision) and identifying all true ones (recall), making it ideal for this task. For assessing practical viability, both **Training Time** and **Inference Time** were systematically captured. Inference time is considered the more critical efficiency metric for this research, as it directly impacts the system's real-world responsiveness and scalability in a production environment.

## 4.2 Baseline Performance: Direct LLM Classification

The initial phase of this research sought to establish a performance baseline by evaluating the feasibility of using a Large Language Model for end-to-end course equivalency classification, a methodology detailed in Section 3.1. The experiments were conducted on the Initial Dataset, leveraging Google's Gemini Pro v1.0 as the reasoning engine. The results were encouraging, with the model achieving a peak accuracy of 90.5% when using full, unprocessed raw text as input. This approach consistently yielded superior results compared to using only extracted structured topics. This performance gap is likely attributable to the challenges inherent in the data extraction step itself, which proved to be difficult and

Table 4.2: Performance Summary of Direct LLM Classification

| Input Data | Classification Task | Accuracy | F1-Score* | Precision* | Recall* |
|---|---|---|---|---|---|
| Raw Text | Binary | 0.9050 | 0.8973 | 0.9765 | 0.8300 |
| | 3-way | 0.8750 | 0.8877 | 0.9818 | 0.8100 |
| | 4-way | 0.8100 | 0.8596 | 0.9808 | 0.7650 |
| Topics | Binary | 0.8100 | 0.7654 | 1.0000 | 0.6200 |
| | 3-way | 0.7775 | 0.7795 | 0.9847 | 0.6450 |
| | 4-way | 0.7225 | 0.7531 | 0.9839 | 0.6100 |

* F1-Score, Precision, and Recall are reported for the positive class ("Equivalent").

highly sensitive to prompt wording—a finding that aligns with previous studies on prompt engineering [40, 16, 9, 10, 19].

An analysis of the confusion matrices, presented in Figure 4.1, reveals a consistent and telling behavioral pattern: the model exhibits a strong conservative bias. Across all scenarios, it was far more likely to misclassify a truly equivalent course pair as not equivalent (a false negative) than it was to incorrectly approve a non-equivalent pair (a false positive), suggesting the LLM operates with a high internal threshold for declaring equivalency. Furthermore, the experiments that introduced "unsure" and "insufficient data" categories demonstrated the model's ability to effectively isolate ambiguous cases that would require manual review. A comprehensive summary of the performance metrics is provided in Table 4.2, and a comparative benchmark against other prominent LLMs is shown in Table 4.3.

This initial study confirmed that while direct LLM classification can achieve high accuracy, it has fundamental limitations, including high computational cost, lack of a quantifiable

(a) Binary Raw Text

(b) Binary Topics

(c) 3-way Raw Text

(d) 3-way Topics
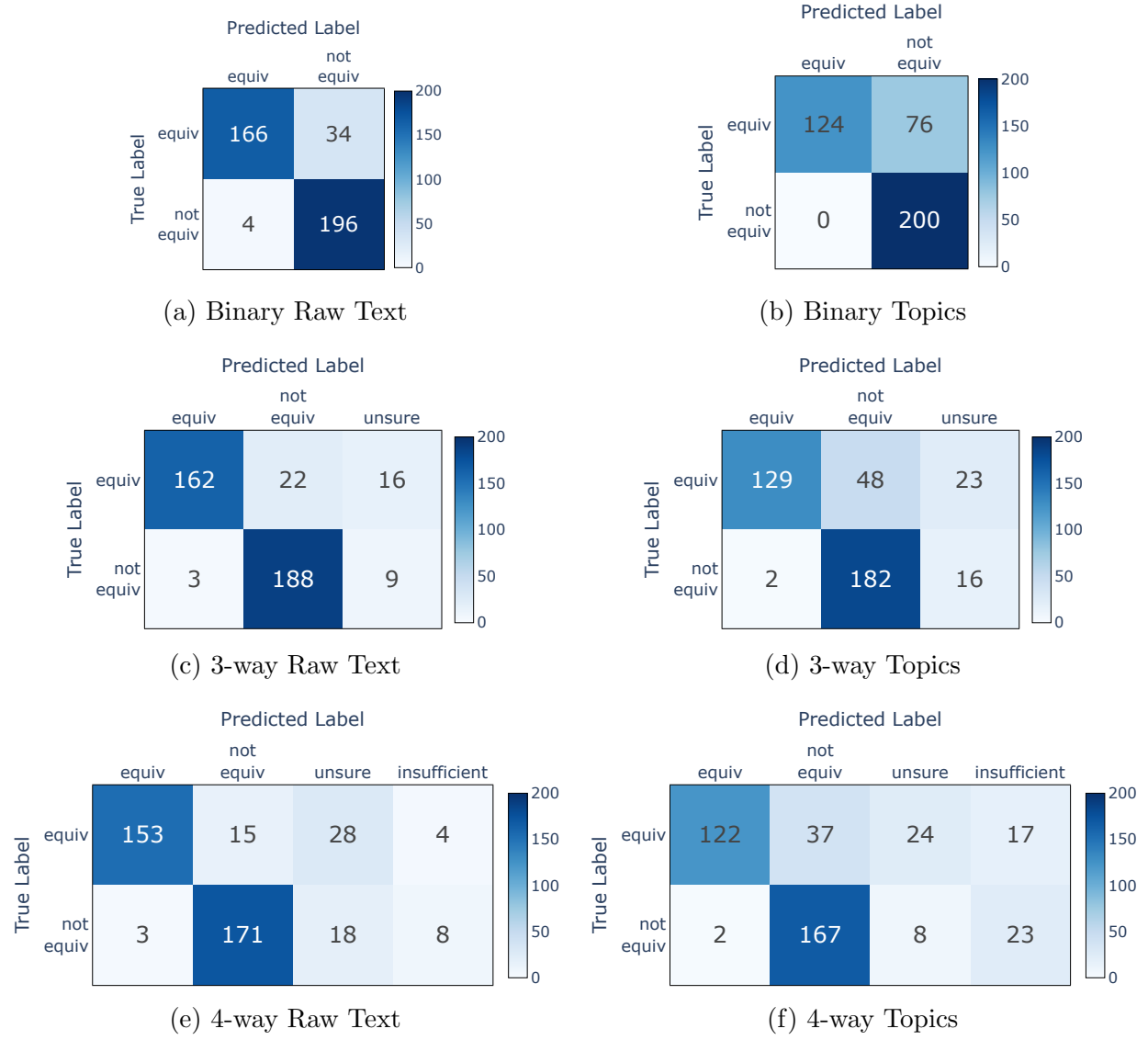
(e) 4-way Raw Text

(f) 4-way Topics

Figure 4.1: LLM Classification Confusion Matrices

similarity score, sensitivity to input quality, and an opaque, "black box" nature. These challenges validated the decision to pivot and motivated the development of the more robust, decoupled pipeline that is the primary focus of this thesis.

Table 4.3: Model Specifications and Performance

| Model Name | Parameters* | Context Length | Support | Accuracy | Precision | Recall | $F_1$-score |
|---|---|---|---|---|---|---|---|
| Google Gemini Pro 1.0 | Unknown | 32,768 | 400 | **0.9050** | 0.9765 | **0.8300** | **0.8973** |
| Meta Llama 3.1 8B Instruct | 8 | 128,000 | 208 (52%) | 0.6250 | 1.0000 | 0.3500 | 0.5185 |
| Microsoft Phi 3 Medium Instruct | 14 | 128,000 | 400 | 0.7100 | 1.0000 | 0.4200 | 0.5915 |
| Google Gemma 2 27b | 27.2 | 8,000 | N/A | N/A | N/A | N/A | N/A |
| Meta Llama 3.1 70B Instruct | 70 | 128,000 | 400 | 0.7350 | 1.0000 | 0.4700 | 0.6395 |
| Qwen 2 72B Instruct | 72.7 | 131,072 | 400 | 0.7650 | 1.0000 | 0.5300 | 0.6928 |
| Anthracite Magnum v1 72B | 72.7 | 32,768 | 400 | 0.8525 | 1.0000 | 0.7050 | 0.8270 |
| CalmeRys 78B Orpo v0.1 | 78 | 32,768 | 400 | 0.7175 | 1.0000 | 0.4350 | 0.6063 |
| Mixtral 8x22B Instruct v0.1 | 141 | 65,536 | 400 | 0.6450 | 1.0000 | 0.2900 | 0.4496 |
| Meta Llama 3.1 405B Instruct** | 405 | 128,000 | 400 | 0.7775 | 1.0000 | 0.5550 | 0.7138 |

*in Billions
**INT4 Quantized Model

# 4.3 Core Component Validation: The Composite Distance Vector

A foundational contribution of this research is the composite distance vector, $\Delta_c$, introduced in Section 3.2. This vector was designed to provide a richer feature set for downstream classifiers by combining granular, dimension-specific differences with a holistic, global cosine similarity metric. To validate this design, a critical ablation study was conducted to quantify the impact of the global cosine similarity term on classifier performance. Classifiers were trained on both the complete vector ($\Delta_c$) and an ablated version containing only the local, element-wise differences ($\Delta_l$).

The results revealed a stark dichotomy between the behavior of linear and non-linear models. The performance of linear models, such as Logistic Regression, showed a dramatic improvement with the inclusion of the cosine similarity term. For example, the $F_1$-score for Logistic Regression surged from 0.686 to 0.901 with the addition of the single global

feature. This demonstrates that these simpler models, which are constrained to learning linear decision boundaries, rely on this explicit global feature and are unable to derive the relationship from the local difference vector alone. In contrast, the performance of more complex, non-linear models (e.g., KNN, SVM, and Random Forest) was not significantly affected. Their exceptionally high performance remained stable with or without the cosine similarity term, suggesting these more powerful models are capable of inferring the global relationship directly from the high-dimensional local difference vector.

The results of this ablation study are conclusive regarding the utility of the composite distance vector, $\Delta_c$. For linear models, its inclusion is critical, providing a substantial boost in performance. While the more powerful non-linear models did not derive a significant benefit, their performance was not negatively impacted. Therefore, to maintain a consistent and robust feature engineering pipeline, the composite distance vector was retained as the standard feature representation for all subsequent experiments. The investigation into alternative composite vectors that could potentially yield improvements for non-linear models is an avenue for future research.

## 4.4 Domain-Specific Adaptation: Embedding Model Fine-Tuning

This section details the empirical investigation into enhancing a pre-trained embedding model by fine-tuning it on the domain-specific PPM Corpus. This process, a form of deep metric learning, aims to restructure the embedding space such that geometric distance directly corresponds to semantic similarity. Such an approach is particularly relevant for scenarios with high intra-class variance and low inter-class variance, a common characteristic of specialized domains where fine distinctions are paramount [29].

### Fine-Tuning Experiment Summary

To identify the optimal fine-tuning configuration, a systematic evaluation was designed to test the impact of different loss functions and learning rate schedules on the `BAAI/bge-small-en-v1.5` (BGE) base model. A matrix of twelve experimental configurations was executed, crossing the four triplet mining loss functions and three learning rate schedulers whose theoretical underpinnings were detailed in Section 3.3. The training was conducted on the POLARIS HPC cluster using a distributed data-parallel strategy. To ensure efficient and robust training, the `GroupByLabelBatchSampler` was employed for effective triplet mining, and model performance was monitored at the end of each epoch using the $F_1$-score on a binary classification task. This was a deliberate choice to evaluate the model on its ultimate downstream application rather than the triplet loss objective itself, providing a more honest assessment
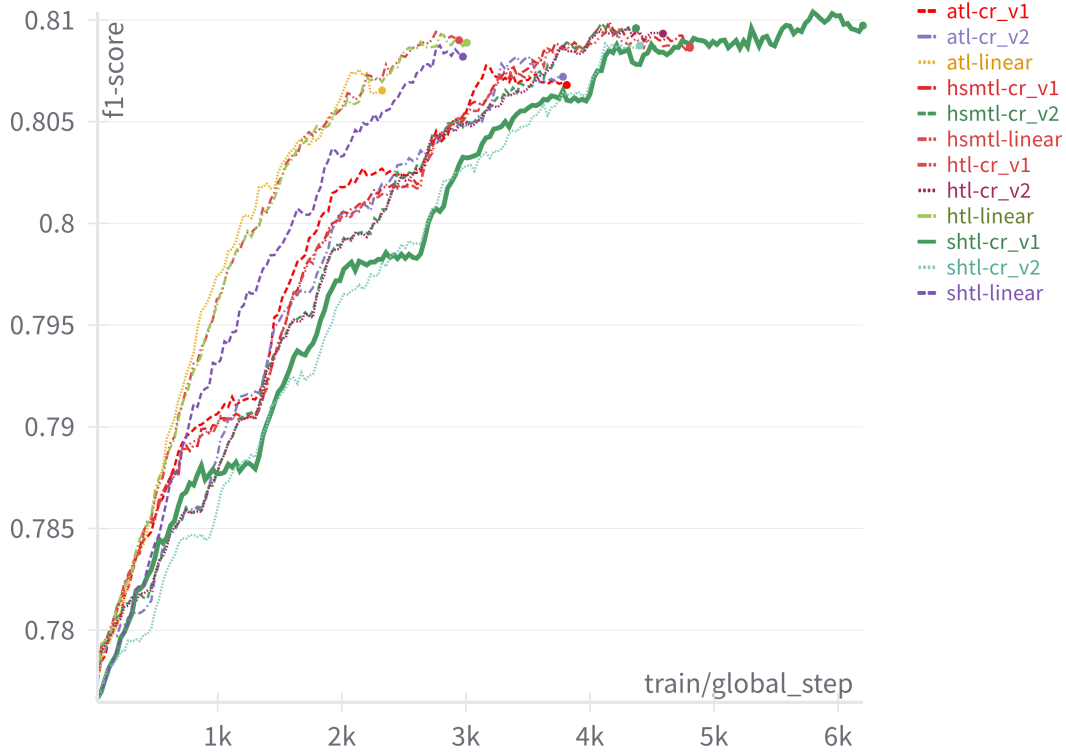
Figure 4.2: Validation $F_1$-Score

of its ability to generalize.

## Empirical Results

The systematic evaluation of the twelve fine-tuning configurations produced a rich set of performance data. Figure 4.2 visualizes the validation $F_1$-score over the training process, providing a qualitative comparison of the learning dynamics for each configuration. For a more precise and formal comparison, the exact peak validation $F_1$-scores achieved by each configuration are summarized in Table 4.4.

The quantitative data reveals a nuanced outcome. While aggressive hard-negative min-

Table 4.4: Peak Validation F1-Score of Fine-Tuning Configurations on BGE Model

| Loss Function | Linear Scheduler | Cosine Restarts v1 | Cosine Restarts v2 |
|---|---|---|---|
| BatchAllTripletLoss (atl) | 0.8076 | 0.8078 | 0.8082 |
| BatchHardSoftMarginTripletLoss (hsmtl) | 0.8094 | 0.8099 | 0.8098 |
| BatchHardTripletLoss (htl) | 0.8093 | 0.8097 | 0.8096 |
| BatchSemiHardTripletLoss (shtl) | 0.8088 | **0.8104** | 0.8088 |

ing strategies such as `BatchHardTripletLoss` and `BatchHardSoftMarginTripletLoss` consistently provided a high performance floor, the single best-performing configuration was the combination of `BatchSemiHardTripletLoss` with the `cr_v1` cosine annealing schedule, which achieved the highest peak validation $F_1$-score of 0.8104. This result suggests that a more moderate mining strategy, when paired with a patient and robust optimization schedule, can find a slightly better optimum for this specific task.

## Statistical Validation and Selection of the Optimal Model

The superior performance of the winning configuration arises from a synergistic framework: a stable learning objective from semi-hard mining, a robust exploration strategy from the `CosineAnnealingWarmRestarts` scheduler that allows the optimizer to escape local minima, and effective regularization from the AdamW optimizer [26, 20, 39, 27].

To definitively validate the benefit of this fine-tuning process, the performance of the resulting model (bge-ft) was compared against the suite of off-the-shelf models using the held-out test portion of the PPM Corpus. The fine-tuned bge-ft model not only achieved the highest mean $F_1$ test score ($\mu = 0.9786$) but also exhibited the lowest standard deviation ($\sigma = 0.0045$), indicating it is both more accurate and more consistent on unseen data. A

one-way Analysis of Variance (ANOVA) confirmed that a statistically significant difference exists between the model groups ($F = 6.2856, p < 0.001$). A subsequent Games-Howell post-hoc test, justified by a significant Levene's test for unequal variances, revealed that the bge-ft model demonstrated a statistically significant improvement over its base model, bge, with a mean $F_1$-score increase of 0.0109 ($p < .001, 95\% CI[0.0054, 0.0164]$). Notably, bge-ft significantly outperformed all other evaluated models, including those that were orders of magnitude larger ($p = 0.0009$ vs. gist; $p = 0.0003$ vs. nve; $p = .0115$ vs. sfr).

This rigorous evaluation leads to a clear and unambiguous conclusion. The single best model for this task is the `BAAI/bge-small-en-v1.5` model fine-tuned using `BatchSemiHardTripletLoss` and a `CosineAnnealingWarmRestarts` learning rate schedule. This model, hereafter designated as **bge-ft**, was proven to be statistically superior to all off-the-shelf models evaluated in this study and is carried forward as the primary fine-tuned embedding model for the comprehensive classifier evaluation detailed in the next section.

## 4.5 Final Stage Evaluation: Downstream Classifier Performance

This section presents the definitive evaluation of the downstream classifiers, representing the final component of the proposed framework. The objective is to systematically identify the optimal classification model by balancing predictive accuracy against computational

efficiency. The evaluation proceeds from a broad preliminary review to a focused analysis of four finalist classifiers on the PPM Corpus, culminating in a statistical comparison on held-out test data.

## Preliminary Classifier Performance Review and Finalist Selection

The initial assessment, conducted on the Initial Dataset, tested eight different classifiers against feature sets derived from the off-the-shelf embedding models. This wide-ranging review established that non-linear models (KNN, SVM, RF) performed exceptionally well, suggesting they could effectively capture the complexity of the feature space. In contrast, linear models showed a profound dependency on the engineered cosine similarity feature, highlighting their limitations.

The preliminary review also systematically tested multiple dimensionality reduction techniques, including PCA, t-SNE, and PaCMAP. For the top-performing non-linear models (KNN, SVM, RF), applying dimensionality reduction generally had a neutral or slightly negative impact on performance. This suggests that while the techniques reduce complexity, they may also discard some useful, discriminative information contained within the original high-dimensional embeddings, as evidenced by instances where scores dropped after reduction was applied.

Based on these preliminary results, the most robust and successful classifiers—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF)—were selected

as finalists. To ensure the final comparison was comprehensive, XGBoost was added to this slate to include a state-of-the-art gradient boosting algorithm.

## Efficacy vs. Efficiency Analysis

Transitioning to the PPM Corpus and using features generated by the superior **bge-ft** model, the analysis next investigated the crucial trade-off between model efficacy and computational efficiency for the four finalist classifiers. Figure 4.3 visualizes the median $F_1$-scores from the cross-validation grid search against both training and inference times. While all four models demonstrate the ability to achieve high accuracy, Random Forest and XGBoost emerge as the clear winners on the critical metric of inference time. Their inference times are tightly clustered under 0.1 seconds—an order of magnitude faster and more predictable than the wider, more variable times of KNN and SVM. This reliability makes RF and XGBoost inherently more robust choices for a scalable, low-latency production system.

## Statistical Analysis of Classifier Performance on Test Data

The definitive comparison was conducted on the held-out test data from the PPM Corpus. As shown in the descriptive statistics (Table 4.5) and score distributions (Figure 4.4), all four classifiers demonstrate exceptionally high and stable performance. Notably, the Support Vector Machine (SVM) model achieved both the highest mean $F_1$-score ($\mu = 0.975973$)
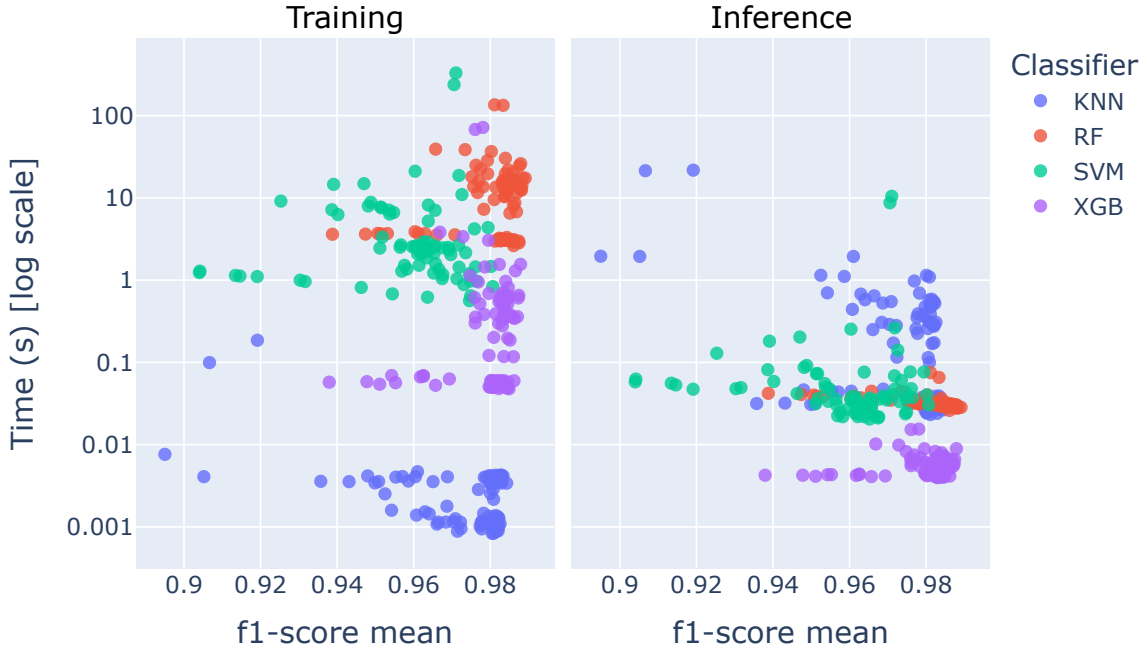
Figure 4.3: Classifier $F_1$-Score vs. Training and Inference Time

Table 4.5: Model Specifications and Performance

| Classifier | Count | Mean | Std | Min | 25% | 50% | 75% | Max |
|---|---|---|---|---|---|---|---|---|
| KNN | 80.0 | 0.971147 | 0.015208 | 0.900304 | 0.971809 | 0.977110 | 0.979110 | 0.982437 |
| RF | 80.0 | 0.970397 | 0.014636 | 0.934443 | 0.957361 | 0.978490 | 0.982108 | 0.984572 |
| SVM | 80.0 | 0.975973 | 0.009120 | 0.939885 | 0.970523 | 0.979408 | 0.982786 | 0.987702 |
| XGB | 80.0 | 0.971026 | 0.012316 | 0.942860 | 0.959382 | 0.977547 | 0.981376 | 0.984898 |

and the lowest standard deviation ($\sigma = 0.009120$), indicating it is the most accurate and consistent performer.

To determine if these observed differences were statistically significant, a one-way Analysis of Variance (ANOVA) was performed, which confirmed a significant difference among the classifiers ($F(3, 316) = 3.1293, p = 0.02596$). A subsequent Games-Howell post-hoc test, justified due to unequal variances, revealed that the SVM classifier performed statistically
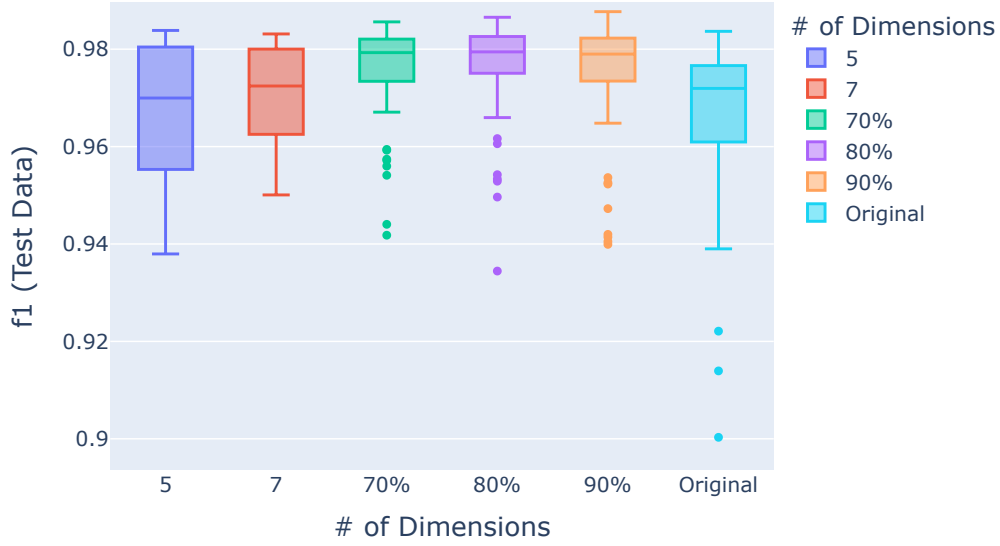
Figure 4.4: $F_1$-Score Distribution of Finalist Classifiers

significantly better than both Random Forest (mean difference $= 0.0056, p = 0.0229$) and XGBoost (mean difference $= 0.0049, p = 0.0229$). While all four are top-tier performers, the formal statistical analysis identifies SVM as the definitive winner in terms of raw accuracy.

## Final Classifier Selection

This multi-stage evaluation reveals a classic trade-off between peak performance and operational efficiency. The statistical superiority of SVM is clear; however, it corresponds to a mean $F_1$-score improvement of only 0.5% over the far more efficient tree-based models. The final choice is therefore context-dependent. Based on the comprehensive evidence, the following recommendations are made:

- **For Maximum Accuracy**: The Support Vector Machine (SVM) is the recommended

classifier for any scenario where achieving the absolute highest accuracy and consistency is the paramount objective.

- **For Optimal Efficiency**: Random Forest and XGBoost are compelling and practical alternatives for applications where inference speed, low latency, and predictable performance are critical for scalability. They provide nearly equivalent accuracy with demonstrably superior and more reliable computational efficiency.

## 4.6 Qualitative Diagnosis: Misclassification Analysis

While quantitative metrics are essential for benchmarking and model selection, they obscure the underlying nature of a model's failures. Relying on aggregate scores alone can be misleading, as they may overestimate a model's robustness while hiding significant, systematic failure modes [18]. This section, therefore, transitions from quantitative assessment to a qualitative diagnosis of the misclassifications produced by the embedding models to understand their root causes. The analysis examines shared misclassifications across models and conducts a granular, case-based review of false positives and false negatives.

### Shared Misclassifications Across Models

To diagnose the source of model errors, it is first necessary to determine whether they are idiosyncratic to individual models or systematic across them. A significant overlap in misclassified pairs across multiple, diverse models would suggest the presence of errors that

stem not from the models but from the data itself, such as inherent ambiguity or annotation artifacts.

The analysis of misclassifications on the held-out test data reveals a high degree of overlap, providing strong evidence for the prevalence of such systematic errors. As illustrated in Figure 4.5, a total of 211 distinct course pairs were misclassified by every single embedding model evaluated, from the large, general-purpose models to the small, domain-specific **bge-ft**. The intersection bar charts in Figure 4.6 reinforce this finding, showing high counts of shared errors across all model combinations. The interpretation is clear: a significant portion of the model failures are not random but are systematic products of the course catalog corpus. These "hard" examples consistently challenge a range of semantic embedding models and may define the performance ceiling of any approach relying solely on course descriptions.

## Qualitative Analysis of False Positives

A False Positive (FP) occurs when two non-equivalent courses are incorrectly classified as equivalent. This type of error carries significant real-world consequences, as it could lead a student to waste time and tuition on a non-transferable course. The analysis identified two primary causes:

- **Topical Overlap without True Equivalence:** This error arises when courses cover the same broad subject but differ in critical dimensions such as academic level or their position in a curricular sequence. The model correctly identifies high semantic similar-
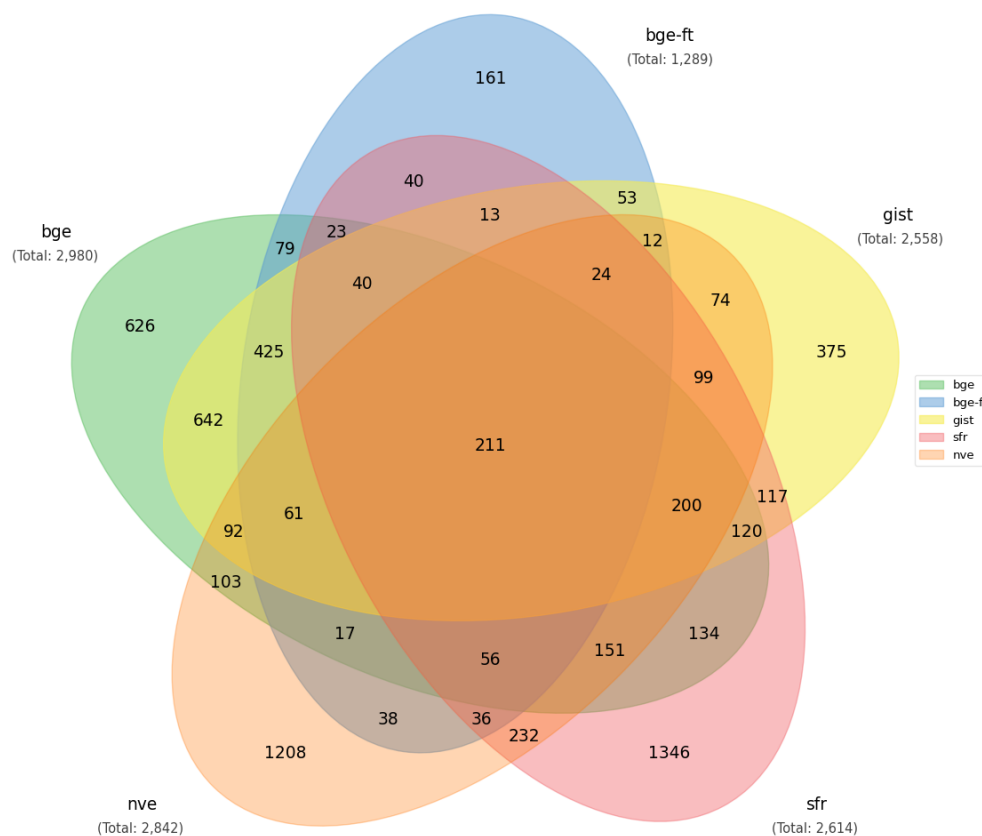
Figure 4.5: Venn diagram illustrating the overlap of misclassified pairs on the test dataset between all five embedding models. The central intersection shows 211 pairs were misclassified by all models, indicating systematic data challenges.

ity but fails to capture the nuanced distinctions that make the courses non-equivalent. An example is the pair of PHYS-4D and PHYS-4A from Foothill College, which are sequential calculus-based physics courses covering modern and classical mechanics, respectively. Their shared title and subject matter create high semantic overlap, but they are not equivalent for transfer purposes.

- **Ambiguous or Vague Course Descriptions:** This occurs when descriptions are too
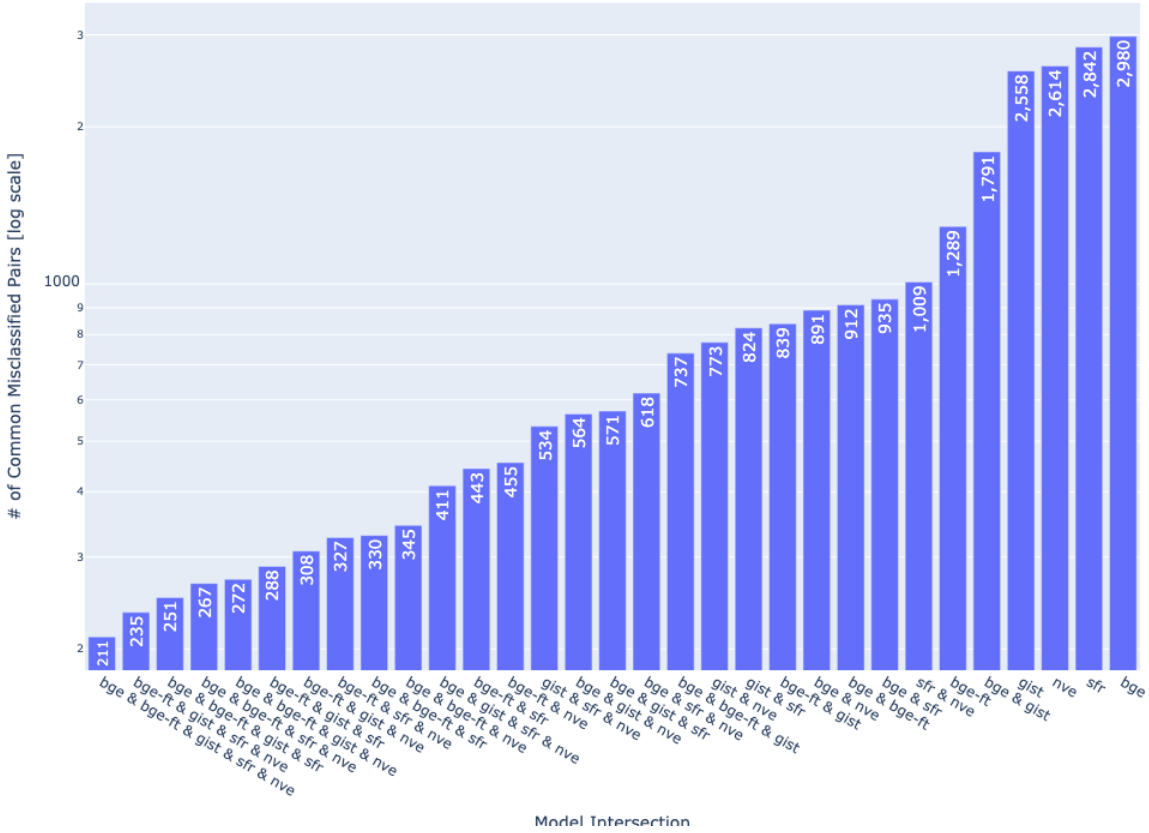
Figure 4.6: Number of common misclassified pairs between all model combinations on the test dataset. The high counts of shared errors across different models point to systematic challenges inherent in the data.

brief or use generic language, lacking the specific detail needed for differentiation. This is a known challenge in the field of short-text semantic similarity, where a lack of rich context inherently increases ambiguity [3]. For example, the description for COMM-1 at Saddleback College uses abstract language like "processes of communication" and "development of ideas," which could be semantically close to a wide range of introductory courses in public speaking, rhetoric, or even philosophy.

## Qualitative Analysis of False Negatives

A False Negative (FN) occurs when the system fails to identify a true equivalence, representing a missed opportunity that could cause a student to unnecessarily retake a course. These errors are predominantly a consequence of inconsistencies and information gaps in the source data itself:

- **Semantic Divergence in Descriptions:** This error occurs when two officially equivalent courses are described using vastly different terminology, phrasing, or pedagogical focus. For example, a pair of courses with the same C-ID (CDEV-100) are described differently: one (Foothill College's CHLD-2) uses the language of traditional developmental psychology ("milestones," "psychosocial"), while the other (Cerritos College's CD-110) uses the language of social justice pedagogy ("diversity and inclusion," "anti-bias curriculum"). The model correctly assesses the texts as semantically dissimilar; the failure lies in the inconsistency of the source data, where the ground truth asserts an equivalence not present in the text.

- **Incomplete or Minimalist Descriptions:** This error arises when one or both course descriptions in an equivalent pair are too sparse to provide sufficient textual signal for the model to establish a confident match. An example is a pair of English literature courses (both C-ID ENGL-120), where one description from Bakersfield College is rich with keywords like "critical analysis" and "prose fiction," while the corresponding description from Santa Barbara City College is a brief, high-level survey summary,

lacking the conceptual depth for the model to map the two courses closely.

- **Data Quality and Labeling Errors:** In some cases, a false negative can be traced back to a fundamental error in the ground-truth data. A striking example is a pair of courses labeled with C-ID SOCI-125, where the description for one course is for an LGBTQ+ studies class, while the description for the other is for an introduction to statistics. This is almost certainly a data entry error in the source corpus. The model correctly identifies the courses as non-equivalent but is penalized with a false negative because the ground-truth label is erroneous.

## 4.7 Summary

This chapter presented a comprehensive, multi-stage empirical evaluation that validated the proposed decoupled, deep metric learning framework. The investigation began by establishing a performance baseline using a direct Large Language Model approach, which, while capable, highlighted critical limitations that motivated the development of the primary pipeline. The subsequent validation of the framework's core components proved highly successful. A critical ablation study confirmed that the novel composite distance vector, $\Delta_c$, is a demonstrably superior feature representation.

The most significant performance gain was achieved through domain-specific fine-tuning. The resulting **bge-ft** model, adapted to the PPM Corpus using a `BatchSemiHardTripletLoss` objective, was shown to be statistically superior to all off-the-shelf models, providing strong

empirical evidence for the efficacy of applying deep metric learning to engineer a bespoke embedding space for the course catalog domain. In the final evaluation stage, while all four finalist classifiers achieved exceptionally high $F_1$-scores, a nuanced trade-off between peak accuracy and computational efficiency emerged. Statistical analysis identified the Support Vector Machine (SVM) as the most accurate and consistent classifier, whereas Random Forest and XGBoost proved to be significantly more efficient, making them compelling alternatives for real-world deployment.

Finally, a qualitative misclassification analysis provided a essential perspective on the system's performance. The analysis revealed that despite the high accuracy of the optimized pipeline, the remaining errors are not random but are largely systematic, stemming from challenges inherent to the data itself, such as semantic divergence in the descriptions of equivalent courses, ambiguity from vague descriptions, and fundamental data quality errors. The comprehensive evaluation, therefore, concludes that while the proposed framework is highly effective, the primary bottleneck for further improvement likely does not lie in the model architecture or classification algorithm, but in the quality and consistency of the source data.

# Chapter 5

# Discussion, Future Work, and

# Conclusion

This chapter transitions from the empirical validation of the proposed framework to a broader discussion of its implications. Having systematically evaluated each component and demonstrated the model's high performance in Chapter 4, this chapter now seeks to interpret the significance of these results in the context of the course articulation problem. The objective is to discuss the key findings, acknowledge the inherent limitations of the study, and outline promising avenues for future research that build upon this work. The chapter will culminate in a final conclusion, summarizing the primary contributions of the thesis and reiterating its significance for fostering a more equitable and efficient higher education ecosystem.

## 5.1 Discussion of Results

The empirical results presented in Chapter 4 offer a robust validation of the core hypothesis of this thesis: that a decoupled, deep metric learning framework can overcome the limitations of prior automated approaches to course articulation. This section discusses the significance of these findings, focusing on the vindication of the proposed framework, the critical impact of domain-specific fine-tuning, and the resulting shift in focus from model-centric to data-centric challenges.

### A Vindicated Framework

The proposed pipeline successfully addresses the distinct challenges that have hindered previous attempts at automation. By relying exclusively on publicly available course catalog data, the framework is an inherently privacy-preserving alternative to enrollment-based methods like *course2vec*, which are constrained by their need for sensitive, proprietary student records and are not generalizable to institutions with no prior transfer history [33, 42]. Furthermore, by decoupling semantic representation from classification, the framework is significantly more scalable, efficient, and interpretable than using large language models for direct, end-to-end classification. It avoids the high computational costs, "black box" opacity, and prompt sensitivity associated with direct LLM approaches while still providing a quantifiable similarity score for each pair. Finally, its use of deep contextual embeddings represents a fundamental advance over older statistical methods like TF-IDF, which lack any

true semantic understanding and cannot grasp synonymous or related concepts [2].

## The Critical Impact of Domain-Specific Fine-Tuning

Perhaps the most significant finding from the experimental evaluation is the statistical superiority of the fine-tuned **bge-ft** model over all off-the-shelf competitors, including those that are orders of magnitude larger. This result provides powerful evidence that for specialized domains, targeted adaptation is more effective than sheer scale. General-purpose models, despite being trained on vast swaths of the internet, lack the specialized "vocabulary" to appreciate the fine-grained distinctions in course catalog text. For example, they may not understand the subtle but critical differences between a "survey" course, an "introductory" course, and a "foundations" course. The process of fine-tuning with a triplet loss objective effectively retrained the model's attention mechanism, teaching it the specific semantics and nuances of the academic domain. This allowed it to generate a far more discriminative embedding space, ultimately leading to higher accuracy in the downstream classification task.

## The Bottleneck Has Shifted from Model-Centric to Data-Centric

With the optimized pipeline achieving $F_1$-scores approaching or exceeding 0.98, the framework has pushed the limits of what can be achieved with the available data. The qualitative misclassification analysis in Section 4.6 revealed that the vast majority of the remaining

errors are not due to failures in the model's semantic understanding. Instead, they are artifacts of the source data itself. The model fails when officially equivalent courses are described with vastly different pedagogical language (semantic divergence) or when descriptions are too vague or minimalist to contain a clear signal. In these cases, the model is performing correctly—it accurately reports that the texts are not semantically similar. The error lies in the ground-truth expectation that an equivalence should be found where none is textually supported.

This leads to a critical insight: the primary bottleneck for achieving near-perfect automation has shifted from being model-centric to data-centric. Simply using a larger or more complex model is unlikely to resolve these data-inherent issues. This suggests that the most promising path to further improvement lies not in novel architectures, but in methodologies that directly address the quality and consistency of the input data [18], a point that will be explored further in the following sections.

## 5.2 Limitations of the Current Study

While the proposed framework represents a significant advance, it is essential to acknowledge the limitations that define the boundaries of the current study. These limitations, primarily rooted in the nature of the data and the scope of the task, provide critical context for the results and inform the directions for future work.

## Data Quality as a Performance Ceiling

The primary limitation, as identified in the discussion, is that the framework's performance is fundamentally capped by the quality and content of the public course descriptions. The system can only analyze the text that is provided; it cannot infer information that is absent. As the misclassification analysis demonstrated, when course descriptions are vague, minimalist, or use semantically divergent language to describe functionally equivalent courses, the model's ability to determine a correct match is severely hindered. This reliance on the source text means the system is vulnerable to inconsistencies and information gaps in how institutions write and publish their catalogs.

## Generalizability of the Fine-Tuned Model

The **bge-ft** model was fine-tuned and evaluated on a corpus drawn exclusively from California's public colleges and universities. While the framework itself is general, the specific fine-tuned model has been specialized for the linguistic patterns, terminology, and pedagogical styles common to this system. Its performance may not be as high "out-of-the-box" on data from private institutions or different state systems or non-domestic systems, which may have distinct catalog writing conventions. Achieving similar performance in a new institutional context would likely require re-tuning the model on a sample of local data.

## Scope of "Equivalency"

This research operationalizes course equivalency as a function of the semantic similarity of their catalog descriptions. This is a powerful proxy, but it does not encompass the full spectrum of factors that human articulation officers may consider. Decisions made by faculty and administrators can be influenced by factors beyond the written content, such as the rigor of the assessment methods, specific lab equipment, faculty credentials, or overarching institutional agreements. The current model is not designed to capture this external, non-textual context.

## Handling of Complex Articulation Rules

The framework simplifies the articulation task into a binary classification of course pairs (equivalent or not-equivalent). This approach does not natively handle the more complex articulation scenarios that exist in practice, such as one-to-many (e.g., one university course is equivalent to two community college courses), many-to-many, or non-symmetrical agreements. While the similarity scores produced by the system could inform the discovery of such relationships, the current classification pipeline is not designed to identify them directly, a challenge that persists for many automated systems [30].

## 5.3 Future Work

The findings and limitations of this study give rise to several promising avenues for future research. These directions aim to address the remaining challenges by improving the quality of the input data, expanding the framework's capabilities, and exploring more advanced modeling techniques.

Given that data quality was identified as the primary performance bottleneck, the most critical future work involves data-centric strategies. An interactive, human-in-the-loop system could be developed where the model flags ambiguous pairs for an expert reviewer. This system could be further enhanced with dynamic data augmentation; for instance, if a classification falls below a predetermined confidence threshold, the framework could automatically request a more detailed syllabus for the course pair to enable a more informed re-assessment. In addition to manual review, research could explore methods to automatically enrich or standardize course descriptions before analysis, perhaps by using a large language model in a controlled, pre-processing step to create more consistent inputs.

Beyond improving the data, the framework itself can be expanded to be more useful and tunable for institutional needs. In fact, active development is currently underway to evolve the framework beyond binary classification and build a full-scale course recommendation engine, which is currently being actively researched and developed by our team and will provide a conversational interface for a more intuitive user experience. Such a system would use the vector similarity scores to provide students and advisors with a ranked list of the

most likely equivalent courses at a target institution. Likewise, Furthermore, future work could investigate the classifier's decision threshold as a mechanism for institutional control, allowing administrators to tune the system's behavior from a more lenient stance to one that errs on the side of caution. To address the limitation of complex articulations, another avenue of research involves modeling curricula as graphs and applying graph neural networks to identify one-to-many and many-to-many relationships.

Finally, there remain opportunities to refine the core machine learning components of the pipeline. Future work could systematically investigate alternative composite distance measures and feature combinations to determine if a more optimal representation exists for the downstream classifiers. This could be complemented by exploring multi-modal learning, extending the model to analyze not just the catalog description but also the full text of course syllabi or textbook lists. Likewise, more room still remains to explore task-optimized loss functions for fine-tuning. Because the embedding vectors are not being used directly, but as upstream feature engines, creating loss functions that behave more similarly to the downstream classifier may have a significant impact on training efficacy. As models evolve, more sophisticated training methods, such as instruction-tuning on a small, expert-curated dataset, could also be employed to teach the model the explicit task of explaining course equivalency, potentially yielding more interpretable results.

## 5.4 Conclusion

The manual process of determining course equivalency remains a significant impediment to student mobility in higher education, creating administrative burdens and systemic inequities that lead to student credit loss and delayed graduation [44, 6]. This thesis confronted this challenge by designing, developing, and validating a novel computational framework that successfully automates course articulation using only publicly available data. The work's primary contribution is a privacy-preserving, scalable, and computationally efficient pipeline that overcomes the limitations of previous automated approaches. This was achieved through two key technical innovations: the application of deep metric learning to fine-tune a bespoke embedding model for the specific semantics of the academic domain, and the design of a novel composite distance vector that provides a richer feature set for downstream classification.

The result of this research is a highly accurate framework, capable of achieving state-of-the-art performance on a real-world dataset. More importantly, it represents a practical tool that institutions can use to reduce administrative workload, provide faster and more consistent guidance to students, and ultimately foster a more transparent and equitable educational ecosystem. By mitigating the barriers faced by transfer students, particularly those from underrepresented backgrounds [43], this work contributes a meaningful step toward fulfilling the promise of accessible and efficient pathways through higher education.

# Bibliography

[1] Public Agenda. *Beyond Transfer: Insights from a Survey of American Adults.* `https://publicagenda.org/resource/beyond-transfer/` (visited on 06/30/2025).

[2] Akiko Aizawa. "An information-theoretic perspective of tf-idf measures". In: *Information Processing & Management* 39.1 (2003), pp. 45–65. ISSN: 0306-4573. DOI: `https://doi.org/10.1016/S0306-4573(02)00021-3`. `https://www.sciencedirect.com/science/article/pii/S0306457302000213`.

[3] Zaira Hassan Amur et al. "Short-Text Semantic Similarity (STSS): Techniques, Challenges and Future Perspectives". In: *Applied Sciences* 13.6 (2023). ISSN: 2076-3417. DOI: `10.3390/app13063911`. `https://www.mdpi.com/2076-3417/13/6/3911`.

[4] ASSIST. *Frequently Asked Questions.* `https://resource.assist.org/FAQ` (visited on 06/30/2025).

[5] ASSIST. *General Information.* `https://resource.assist.org/About/General-Information` (visited on 06/30/2025).

[6]  Leticia Tomas Bustillos et al. *The Transfer Maze: The High Cost to Students and the State of California*. The Campaign for College Opportunity, Sept. 17, 2017.

[7]  California Community Colleges Chancellor's Office. *Management Information Systems Data Mart*. 2024. `https://datamart.cccco.edu/Students/Student%5C_Headcount%5C_Term%5C_Annual.aspx` (visited on 09/13/2024).

[8]  California State University Office of the Chancellor. *Enrollment*. 2024. `https://www.calstate.edu/csu-system/about-the-csu/facts-about-the-csu/enrollment` (visited on 09/13/2024).

[9]  Yi-Pei Chen, Kuanchao Chu, and Hideki Nakayama. "LLM as a Scorer: The Impact of Output Order on Dialogue Evaluation". In: *ArXiv* abs/2406.02863 (2024). `https://api.semanticscholar.org/CorpusID:270258565`.

[10]  Ming Cheung. "A Reality check of the benefits of LLM in business". In: *ArXiv* abs/2406.10249 (2024). `https://api.semanticscholar.org/CorpusID:270560730`.

[11]  National Student Clearinghouse. *College Transfer Enrollment Grew by 5.3% in the Fall of 2023*. `https://www.studentclearinghouse.org/news/college-transfer-enrollment-grew-by-5-3-in-the-fall-of-2023/` (visited on 06/30/2025).

[12]  National Student Clearinghouse. *College Transfer Enrollment Grew for Third Straight Year*. `https://www.studentclearinghouse.org/news/`

`college-transfer-enrollment-grew-for-third-straight-year/` (visited on 06/30/2025).

[13] National Student Clearinghouse. *DATA DIVE: Returning Learners Lead Transfer Population.* `https : / / www . studentclearinghouse . org / nscblog / data-dive-returning-learners-lead-transfer-pop/` (visited on 06/30/2025).

[14] Kevin Cook. *California's Higher Education System.* 2024. `https://www.ppic.org/ publication/californias-higher-education-system/` (visited on 09/06/2024).

[15] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: `1810.04805 [cs.CL]`. `https://arxiv.org/abs/ 1810.04805`.

[16] Federico Errica et al. "What Did I Do Wrong? Quantifying LLMs' Sensitivity and Consistency to Prompt Engineering". In: *ArXiv* abs/2406.12334 (2024). `https:// api.semanticscholar.org/CorpusID:270562829`.

[17] The Pytorch Foundation. *CosineAnnealingLR.* `https://docs.pytorch.org/docs/ stable/generated/torch.optim.lr_scheduler.CosineAnnealingLR.html` (visited on 06/30/2025).

[18] Gabrielle Gauthier-melancon et al. "Azimuth: Systematic Error Analysis for Text Classification". In: Jan. 2022, pp. 298–310. DOI: `10.18653/v1/2022.emnlp-demos.30`.

[19] Walter Gerych et al. "Who Knows the Answer? Finding the Best Model and Prompt for Each Query Using Confidence-Based Search". In: *AAAI Conference on Artificial Intelligence*. 2024. `https://api.semanticscholar.org/CorpusID:268717587`.

[20] Alexander Hermans, Lucas Beyer, and Bastian Leibe. *In Defense of the Triplet Loss for Person Re-Identification*. 2017. arXiv: `1703.07737 [cs.CV]`. `https://arxiv.org/abs/1703.07737`.

[21] Inc. Hugging Face. *Losses*. `https://sbert.net/docs/package_reference/sentence_transformer/losses.html` (visited on 06/30/2025).

[22] Inc. Hugging Face. *Samplers*. `https://sbert.net/docs/package_reference/sentence_transformer/sampler.html` (visited on 06/30/2025).

[23] Weijie Jiang and Zachary A Pardos. "Evaluating Sources of Course Information and Models of Representation on a Variety of Institutional Prediction Tasks." In: *International Educational Data Mining Society* (2020).

[24] Pengfei Liu et al. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: `10.1145/3560815`. `https://doi.org/10.1145/3560815`.

[25] Shamrock Solutions LLC. *Transfer Credit Automation: How Universities Are Simplifying Course Equivalency*. Overland Park, KS, USA. `https://www.shamrocksolutionsllc.com/post/transfer-credit-automation-universities` (visited on 06/30/2025).

[26]  Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: `1711.05101` [`cs.LG`]. `https://arxiv.org/abs/1711.05101`.

[27]  Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: Aug. 2016. DOI: `10.48550/arXiv.1608.03983`.

[28]  H Ma et al. "Course recommendation based on semantic similarity analysis". In: *2017 3rd IEEE International Conference on Control Science and Systems Engineering*. 2017, pp. 638–641.

[29]  Deen Dayal Mohan et al. *Deep Metric Learning for Computer Vision: A Brief Overview*. 2023. arXiv: `2312.10046` [`cs.CV`]. `https://arxiv.org/abs/2312.10046`.

[30]  Z. A Pardos, H Chau, and H Zhao. "Data-assistive course-to-course articulation using machine translation". In: *Proceedings of the Sixth Conference on Learning@ Scale*. 2019, pp. 1–10.

[31]  Zachary Pardos, Hung Chau, and Haocheng Zhao. "Data-Assistive Course-to-Course Articulation Using Machine Translation". In: (July 2019). DOI: `10.1145/3330430. 3333622`.

[32]  Zachary A. Pardos, Hung Chau, and Haocheng Zhao. "Data-Assistive Course-to-Course Articulation Using Machine Translation". In: *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*. L@S '19. Chicago, IL, USA: Association for Computing Machinery, 2019. ISBN: 9781450368049. DOI: `10.1145/3330430.3333622`. `https://doi.org/10.1145/3330430.3333622`.

[33] Zachary A. Pardos, Zihao Fan, and Weijie Jiang. *Connectionist Recommendation in the Wild: On the utility and scrutability of neural networks for personalized course guidance.* 2018. arXiv: `1803.09535 [cs.AI]`. `https://arxiv.org/abs/1803.09535`.

[34] Zachary A. Pardos, Zihao Fan, and Weijie Jiang. "Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance". In: *User Modeling and User-Adapted Interaction* 29.2 (Apr. 2019), pp. 487–525. ISSN: 0924-1868. DOI: `10.1007/s11257-019-09218-7`. `https://doi.org/10.1007/s11257-019-09218-7`.

[35] Stephen Porter. "Assessing Transfer and Native Student Performance at Four-Year Institutions". In: *39th Annual Forum of the Association for Institutional Research.* June 1999.

[36] Regents of the University of California, The. *Fall enrollment at a glance.* 2024. `https://www.universityofcalifornia.edu/about-us/information-center/fall-enrollment-glance` (visited on 09/13/2024).

[37] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Nov. 2019. `https://arxiv.org/abs/1908.10084`.

[38] Elvis Saravia. "Prompt Engineering Guide". In: *https://github.com/dair-ai/Prompt-Engineering-Guide* (Dec. 2022). `https://www.promptingguide.ai`.

[39] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[40] Melanie Sclar et al. "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting". In: *ArXiv* abs/2310.11324 (2023). `https://api.semanticscholar.org/CorpusID: 264172710`.

[41] Natenaile Asmamaw Shiferaw et al. *BERT-Based Approach for Automating Course Articulation Matrix Construction with Explainable AI*. 2024. arXiv: `2411.14254 [cs.LG]`. `https://arxiv.org/abs/2411.14254`.

[42] Sharon Slade and Paul Prinsloo. "Learning Analytics: Ethical Issues and Dilemmas". In: *American Behavioral Scientist* 57.10 (2013), pp. 1510–1529. DOI: `10. 1177/0002764213479366`. eprint: `https://doi.org/10.1177/0002764213479366`. `https://doi.org/10.1177/0002764213479366`.

[43] The National Task Force on the Transfer and Award of Credit. *Reimagining Transfer for Student Success*. Report to Congressional Requesters. American Council on Education, Mar. 2020. `https://www.gao.gov/products/gao-17-574`.

[44] United States Government Accountability Office. *Higher Education: Students Need More Information to Help Reduce Challenges in Transferring College Credits*. Report

to Congressional Requesters GAO-17-574. United States Government Accountability Office, Aug. 14, 2017. `https://www.gao.gov/products/gao-17-574`.

[45]  Yinuo Xu and Zach A. Pardos. "Extracting Course Similarity Signal using Subword Embeddings". In: *Proceedings of the 14th Learning Analytics and Knowledge Conference*. LAK '24. Kyoto, Japan: Association for Computing Machinery, 2024, pp. 857–863. ISBN: 9798400716188. DOI: `10.1145/3636555.3636903`. `https://doi.org/10.1145/3636555.3636903`.

[46]  Qinyuan Ye et al. *Prompt Engineering a Prompt Engineer*. 2024. arXiv: `2311.05661` `[cs.CL]`. `https://arxiv.org/abs/2311.05661`.

[47]  Jian Yu et al. "Deep metric learning with dynamic margin hard sampling loss for face verification". In: *Signal, Image and Video Processing* 14.4 (June 2020), pp. 791–798. ISSN: 1863-1711. DOI: `10.1007/s11760-019-01612-3`. `https://doi.org/10.1007/s11760-019-01612-3`.