Neural Approaches to Decode Semantic Similarities in Spanish Song Lyrics for Enhanced Recommendation Systems

Master's Thesis

Adrián Ghajari Espinosa
Master's degree in Language Technologies

Department of Languages and Computer Systems

National University of Distance Education

Supervised by

Dr. Víctor Fresno Fernández

Dr. Alejandro Benito-Santos

February 2024

Acknowledgments

I extend my hearfelt gratitude to my family, for enduring my journey as though they had any other choice.
Immense thanks to my supervisors, Víctor Fresno Fernández and Alejandro Benito-Santos, for shaping my writing into something that resembles academic literature, and for reminding me that the most valuable lessons often come from friends. Much of my affection for our field can be traced back to their influence.
Lastly, a nod to my coffee machine, for always knowing exactly what to do. Your contribution, though silent, has been indispensable.

Abstract

This dissertation explores the enhancement of music recommendation systems by integrating semantic similarity in Spanish song lyrics, utilizing advancements in machine learning and natural language processing (NLP), including both supervised and unsupervised approaches. It addresses the gap in current recommendation practices, which often overlook the rich semantic content of lyrics, despite their potential to significantly personalize music recommendations. Through theoretical insights into word embeddings and transfer learning, the development of the LyricSIM dataset for assessing lyric similarity, and empirical evaluations of models designed to distinguish between similar and non-similar song pairs, this research proposes a novel, lyrics-driven approach to music recommendation. Focused on the Spanish-speaking market, where Latin music is prevalent, this study contributes to the field by demonstrating how NLP technologies can refine music recommendations, addressing challenges like the cold start problem and enhancing the diversity of music recommendations, thereby offering a more personalized and engaging user experience in the streaming era.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Music streaming services have become the dominant platform for music consumption, with Latin music accounting for a significant percentage of the global market share. According to the 2022 report from the Recording Industry Association of America (RIAA)[1], streaming accounts for 62% of the total music industry revenue, while Latin music carved out a significant 6.6% of the total US market in the first half of 2022 alone. Moreover, a staggering 97% of all Latin music revenues were derived from music streaming formats during this period. This growth is largely attributable to the personalized playlists offered by these services, built using Recommender Systems (RSs) which apply big data and machine learning to massive user datasets. Such playlists, tailored to users' listening habits, is the source of almost three fourths (74%) of the music experienced by the users, as highlighted in the Nielsen's 2017 Music survey[2].

The overwhelming reliance on streaming services, coupled with the demonstrable shift towards personalized content, forms the bedrock of this work. By integrating advanced machine learning techniques and Natural Language Processing (NLP), we propose methods to capture the essence of song similarity, to enrich and enhance RSs. This approach not only seeks to bridge the existing gap in the adaptation of these systems to the unique characteristics of Latin music but also aims to set a precedent for future research in the application of Artificial Intelligence (AI) in music streaming services globally. The present introductory chapter lays the groundwork by detailing the motivations behind the study and outlining the novel contributions we aim to make in the field of RSs.

1.1 Motivation

The increasing prominence of streaming services, as discussed earlier, has made it evident that there is a growing interest in broadening the scope of song recommendation. This

[1] https://www.riaa.com/wp-content/uploads/2023/03/2022-Year-End-Music-Industry-Revenue-Report.pdf

[2] https://www.nielsen.com/wp-content/uploads/sites/2/2019/04/us-music-360-highlights.pdf

recognition underscores the importance of investigating the untapped potential of lyrics, which encapsulate a wealth of semantic information within the song.

Songs are fundamentally dual-natured, composed of both music and lyrics. At present, the majority of industrial music RSs rely on usage patterns, whether implicit feedback or explicit ratings, leveraged by collaborative filtering models, crowdsourcing and manual tagging, to compute personalized recommendations (Deldjoo et al., 2024; Knees et al., 2006; Nanopoulos et al., 2010), or look into the analysis of soundwave similarities (Schedl et al., 2014; Deldjoo et al., 2020, 2022). Other RSs include content-based methods that focus on metadata such as title, composer, genre, tempo, pitch, mood (acoustic) and similarity aspects (cultural), like genre, or listening situation (Aucouturier et al., 2007; Baumann and Hummel, 2003; Sordo et al., 2007; Knees and Schedl, 2013). However, these systems overlook the semantic information embedded in song lyrics, an unstructured yet rich data source unattainable from metadata. While music is often the primary focus in RSs, lyrics represent a vital aspect of national identity and social consciousness, reflecting cultural experiences, societal issues, and even political affiliations (Knees and Schedl, 2013). Genres like rap exemplify the growing importance of lyrics, especially when they connect to nationally relevant topics. Despite this, lyrics' semantic content remains underutilized in current recommendation practices on on-demand music streaming platforms.

In the contemporary landscape, no existing music recommendation services harness the potential of NLP to automate the use of song lyrics in their systems. Those that incorporate lyrics do so by merely identifying fragments of the lyrics, such as significant words. Leading music recommendation services presently include Spotify[3], YouTube Music[4], Amazon Music[5], Apple Music[6], SoundHound[7], Gracenote[8], Melboss[9], and Genius[10]. While each platform possesses its unique strengths, only a small number of them currently facilitate lyric search functionalities, namely Genius and SoundHound. However, these services merely consider the literal text and do not delve into a profound analysis of its meaning. Consequently, they fail to offer search capabilities based on the semantics of the lyrics, which could potentially enrich classification and recommendation systems or augment content itself to extract more granular information. Shazam[11], while housing a repository of basic song information such as the artist and genre, does not permit lyric-based searches, offering recommendations based merely on similar characteristics, like soundwave signatures, genre and mood. Spotify, though not facilitating lyric searches either, presents a richer recommendation experience than its counterparts, incorporating attributes like the song's positivity or negativity, rhythm, or loudness,

[3]https://www.spotify.com/
[4]https://music.youtube.com/
[5]https://music.amazon.com/
[6]https://www.apple.com/apple-music/
[7]https://www.soundhound.com/
[8]https://www.gracenote.com/
[9]https://www.melboss.com/
[10]https://genius.com/
[11]https://www.shazam.com/

beyond basic metadata. SoundHound enables users to vocalize an 'a cappella' rendition or employ a text search engine to locate a fragment of the lyrics; however, it too neglects the use of song lyrics for recommendations. Another music recognition service, Gracenote, despite its extensive metadata (including mood, song age, the gender of the author, among others), does not base its recommendations on lyrical content. Genius, a lyric-centric service utilized by companies such as Apple, mirrors SoundHound's functionality in allowing searches by literal segments of lyrics. Additionally, it offers supplemental content, including insights into song creation and explanations of verse meanings, contributed by both the songwriters and the user community. Consequently, the overarching trend is that existing music recommenders have not capitalized on the potential of automatic in-depth lyrical analysis to refine song recommender systems or to infuse the content itself for more granular information extraction.

In view of the overlooked potential of lyrics, the objective of this work is to delve into the estimation of semantic similarity in Spanish song lyrics for the integration of NLP technologies into RSs, facilitating a deeper understanding and utilization of the semantic contents of Spanish song lyrics to foster a more enriched, personalized user experience in music streaming services. A semantic approach that focuses on the lyrics could address common challenges, such as the cold start problem, arising when new songs are introduced into the system without any prior user interaction data, making it difficult for traditional RSs to recommend these tracks effectively (Knees and Schedl, 2013; Deldjoo et al., 2024) while, at the same time, increasing diversity in other areas (e. g. genre).

To achieve this, we propose exploring word embedding unsupervised composition functions (Amigó et al., 2022) and comparing them with fine-tuning-based approaches in Large Language Models (LLMs). Our goal is to identify songs whose lyrics exhibit semantic similarity —songs that, in essence, convey similar themes or ideas—. In doing so, we aim to broaden the dimensions of music recommendations beyond the traditional markers, offering a more nuanced, lyrics-driven approach. By harnessing the power of linguistic similarity and NLP technologies, we aim to refine and complement existing music recommendation systems, presenting an innovative pathway for personalization in music streaming services.

1.2 Main Contributions

This work offers a foray into the uncharted domain of leveraging semantic similarity in Spanish song lyrics to enhance music recommendation systems. The contributions of this research span several dimensions:

- Theoretical Insights: Including a deep dive into the underlying mathematical and computational principles governing word embeddings, semantic composition, transfer learning, and domain adaptation. By examining transfer learning and domain adaptation strategies specific to this domain, our analysis sheds light on the nuanced requirements for effectively leveraging these technologies. These insights,

detailed in Chapters 2 (Related Work) and 4.1 (Experimental Setup: Word Embeddings), lay the groundwork for our comparative analysis of embedding techniques, emphasizing the importance of specificity in enhancing the accuracy and relevance of song recommendations.

- Practical Methodologies: This work details the development and application of training strategies tailored to the task of semantic analysis in song lyrics, focusing on the integration of both static, non-contextual embedding models and contextual, transformer-based models. Within Chapter 4, "Experimental Setup," we outline the semantic composition methodologies implemented, alongside an analysis of sentence similarity metrics, further enriching the discourse by explicating the evaluation metrics and classifiers deployed. This work emphasizes the selection and fine-tuning of algorithms to address the unique characteristics of semantic similarity in song lyrics, demonstrating a bespoke approach to algorithm application in the context of music recommendation systems.

- Dataset Development: The creation of LyricSIM (Benito-Santos et al., 2023), a novel dataset specifically designed for exploring semantic similarity in Spanish song lyrics. This dataset represents a step forward in the domain of music and lyrics analysis, comprising a diverse collection of human annotated song pairs based on various aspects such as theme, message, emotions, literal meaning, and cultural context. The development of LyricSIM, which will be further detailed in Chapter 4.6 (Experimental Setup: Datasets), addresses the need for domain-specific resources in the Spanish-speaking world. By facilitating the assessment of state-of-the-art (SOTA) models on this unique dataset, our work bridges the gap between general-purpose semantic similarity tasks and the specific nuances of music-related applications, contributing to the advancement of NLP techniques in the context of music recommendation systems.

- Empirical Contributions: Offering a comprehensive evaluation of the developed models through a binary classification task aimed at distinguishing between similar and non-similar song pairs, and introducing novel approaches to semantic analysis through a rich experimental setup encompassing a variety of training scenarios. The empirical findings, illustrated Chapter 5 (Experimental Results), underscore the practical implications of the No Free Lunch theorem (Wolpert and Macready, 1997), which posits that no single algorithm outperforms all others across every possible task. This theorem underpins our comparative analysis of diverse architectural models, underscoring the necessity of tailoring model selection and optimization to the specific challenges of semantic-based song recommendations.

Through these contributions, this dissertation extends the field of music recommendation systems, adopting a detailed approach to model selection and evaluation aimed at refining semantic-based recommendations. In continuation, Chapter 3, "Hypothesis and Research Questions," carefully presents the hypotheses that guide our exploration and the research questions we aim to address.

Chapter 2

Related Work

Building upon the motivation and contributions outlined earlier, this chapter sets the stage for a deeper examination of the existing landscape and scholarly discourse surrounding Music RSs and Text Representation. By establishing the context in which our work is situated, we aim to bridge the gap between the foundational challenges identified in music recommendation systems and the innovative approaches proposed in our research.

We initiate with an Overview of Music Recommendation Systems (Section 2.1), dissecting the operational challenges, technological frameworks, and the distinctive attributes that characterize these systems. This section not only examines the traditional methodologies employed in music recommendation but also highlights the novel potential of incorporating semantic similarity between song lyrics as a means to refine and personalize recommendations. By bringing to light this underexplored aspect, we aim to illustrate how a deeper semantic understanding of lyrics can substantially enhance recommendation systems' effectiveness.

Following this, the discussion transitions to Text Representation (Section 2.2), where we delve into the mathematical and computational principles that facilitate the representation of textual meaning, particularly focusing on the framework for semantic composition. This examination is pivotal in understanding how lyrics, as a form of unstructured text, can be quantitatively analyzed and utilized within the context of music recommendations. By exploring the development and application of compositional distributional semantics, we set out to address the critical role that text representation plays in enabling a nuanced approach to song similarity assessment.

Through this analytical journey, we aim to establish a solid foundation for our research, situating our contributions within a broader academic dialogue and affirming the significance of advancing music RSs through the integration of semantic lyric analysis. This chapter, therefore, not only contextualizes our work within the existing body of knowledge but also delineates the theoretical and methodological pathways that our study endeavors to explore.

2.1 Overview of Music Recommendation Systems

Music streaming platforms have taken over the music industry, resulting in a shift in music consumption patterns. Two significant challenges faced by music recommendation systems are the cold start problem and the automatic playlist generation (Deldjoo et al., 2024). The cold start problem, mentioned in Chapter 1, refers to the challenge of recommending songs to new users or songs with no existing user or song history. Playlists are personalized lists of favored songs, representing specific moods, artists, or genres, and generating these automatically requires accurately deducing the purpose of the current playlist. This extends to the challenge of ranking songs in response to a user-selected metadata query (Biswas et al., 2023).

Given the vast volume of music available online, recommendation systems must adapt to the rapid increase in music data and growing online demand. Music RSs have to take into account certain factors, such as the duration of items, magnitude of items, sequential consumption, repeated recommendations, consumption behavior, listening intent, occasion, context, and associated emotions (Kim et al., 2018). These are often addressed using methods like content-based recommendations, hybridization, and cross-domain recommendations. In this context, one factor that has not been fully explored is the semantic similarity between song lyrics. Leveraging the different degrees of semantic similarity—from completely different lyrics to outstandingly similar ones—could add another layer to these recommendation systems, potentially improving their performance.

Content-based recommendation systems typically use song metadata for recommendations, with user preferences modeled using the history of user interactions and preferences (Velankar et al., 2020). These systems rely on item similarity based on identified features, recommending items with similar attributes. Standard methods to compute similarity include K-means clustering (Han et al., 2018) and Monte Carlo sampling (Li and Zhang, 2018). Here, the opportunity to incorporate semantic similarity between lyrics could complement existing methods. For instance, lyrics with a 'basic similarity' or 'outstanding similarity' might indicate a higher likelihood of user preference for the songs, which could refine recommendations further.

Hybrid recommendation systems address the shortcomings of both content-based and collaborative models (Thorat et al., 2015). These systems operate based on the two-dimensional user vs item matrix, essentially merging the predictions from content-based and collaborative methods (Adomavicius and Tuzhilin, 2005). Such hybrid approaches could be significantly enhanced by integrating lyric similarity as a feature, providing an additional connection point between items in the user-item matrix.

Streaming platforms have caused a significant shift in the music industry, being the largest source of recorded music revenue (O'Dair and Fry, 2020). They aid users in music discovery through various tools, from text searching for song playlists, artist, and release-related metadata, to grouping albums by themes or highlighting latest releases. The integration of semantic similarity between lyrics could be an innovative addition to these tools, possibly facilitating more nuanced and engaging user experiences.

The future of RSs lies in advanced systems like context-awareness systems, group

recommendation systems, systems based on social networks, and techniques based on computational intelligence (Dong et al., 2020; Schedl et al., 2017). In this context, considering the semantic similarity between song lyrics could provide a novel dimension for these future systems, further enhancing their potential to deliver personalized, engaging, and accurate recommendations.

The user's specific intent, personality, novelty-seeking behavior, and context all influence the effectiveness of music recommendations (Swearingen and Sinha, 2001). For instance, the type of music users listen to often depends on their mood and emotions (Moscato et al., 2020). Incorporating the semantic similarity between song lyrics could help in capturing these factors more accurately, enhancing the relevance and appeal of recommendations.

In conclusion, while existing recommendation models have made significant strides, there is a compelling case for integrating the semantic similarity between song lyrics as a complementary feature. The varying degrees of semantic similarity could provide a rich source of information, potentially enhancing user engagement, satisfaction, and the overall effectiveness of music recommendation systems.

2.2 Text Representation

The conflict arising from the principles of compositionality and contextuality, as proposed by Frege, reveals a fundamental tension between the paradigms of meaning representation: *symbolic* and *distributional* (Maruyama, 2019). The Principle of Compositionality, a cornerstone of the symbolic paradigm, posits that the meaning of a whole is determined by the meanings of its constituent parts and the way they are syntactically combined. Conversely, the Principle of Contextuality, which underpins the distributional approach, argues that the meaning of words and utterances depends on their context, supporting the distributional representation paradigm, where meaning is inferred from usage context.

This tension between systematic compositionality, which characterizes the symbolic approach by emphasizing structured, rule-based construction of meaning, and the flexible, usage-based inference of meaning inherent to the distributional approach, reflects a fundamental challenge in meaning representation. Systematicity, a concept tied to the symbolic paradigm, refers to the predictable and orderly combination of semantic units, a property that allows for the generation of an expansive range of expressions from a finite set of elements.

Due to the complementary properties of distributional and compositional representation approaches (contextuality versus systematicity), it is hypothesized that the meaning of linguistic forms lies in a reciprocal flow between their context and components. As a result, several authors have proposed incorporating a compositional layer over distributional representations, leading to the development of Compositional Distributional Semantic models (Mitchell and Lapata, 2010; Arora et al., 2017; Coecke et al., 2010). Our research is situated within this evolving dialogue, with a focus on unraveling the semantic similarity between song lyrics. To this end, representing the lyrics across various

songs to ascertain their semantic likeness constitutes the initial step of our investigation, aiming to harness the synergies of compositional and distributional approaches for a deeper semantic analysis.

Early semantic models were based on logicist approaches that assumed the Principle of Compositionality, such as Preference Semantics (Wilks, 1968). These models established an unambiguous relationship between a symbol (or a set of symbols) in context and its meaning (Boleda and Erk, 2015). From the 1980s onwards, the Statistical Paradigm gained prominence, with the Vector Space Model (VSM) (Salton and Lesk, 1965) serving as its foundational representation system. This paradigm represents texts as bags of independent words, disregarding word order and grammar. The subsequent generation of meaning representation introduced Count-based Language Models, where texts are represented as word sequences along with their probability distribution (Andreas et al., 2013).

Unlike Language Models, which represent texts as sequences of tokens rather than points in continuous space, these models do not offer a direct notion of similarity between word sequences but infer semantic relationships by comparing statistical characteristics across models or by evaluating how well specific sequences align with the probabilistic patterns a model has learned. In recent years, Neural Language Models have emerged as successful approaches, leveraging neural networks pre-trained on vast text corpora based on usage context (Devlin et al., 2019; Brown et al., 2020; Mikolov et al., 2013; Pennington et al., 2014).

In 2008, (Collobert and Weston, 2008) demonstrated that word embeddings generated from sufficiently large datasets carry both syntactic and semantic meaning, enhancing performance on subsequent NLP tasks. Building upon this, some researchers proposed extending the approach to represent longer linguistic units, such as sentences (Kiros et al., 2015) or documents (Le and Mikolov, 2014; Kenter et al., 2016). Static Neural Models, also known as Non-Contextual Neural Models, such as Skip-Gram with Negative Sampling (SGNS) or Global Vectors (GloVe) (Pennington et al., 2014), optimize the correspondence between the scalar product of embeddings and their distributional similarity (Mutual Information) (Levy and Goldberg, 2014; Arora et al., 2016; Le and Mikolov, 2014). By assuming the distributional hypothesis, which posits that similar words appear in similar contexts, these models ensure a certain isometry between the embedding space and meanings. However, these static embedding approaches assign fixed representations to each word, irrespective of its specific context.

Conversely, the second generation of encoders comprises sequential models that are sensitive to the order of words in a sequence. The Transformer model (Vaswani et al., 2017b) exemplifies a successful implementation of this idea. In the Transformer model, the neural network is pre-trained on a large text corpus using various self-supervised tasks such as Masked Language Modeling (MLM), Sequence to Sequence (Seq2Seq), or Permuted Language Modeling (PLM), among others. This pre-training phase has shown to facilitate *inductive transfer learning*, where the model, having learned a broad understanding of language from the large corpus, can then apply this knowledge to improve performance on related but different tasks through fine-tuning (Ruder, 2019). Several

combined approaches, like BERT (Devlin et al., 2019) or GPT (Brown et al., 2020), have been proposed as well. Transformer-based Neural Language Models exhibit high accuracy in solving tasks with limited training samples, achieved through fine-tuning. Moreover, they possess remarkable predictive power over word sequences (Radford et al., 2019). However, existing literature has shown that contextual models do not consistently maintain isometry with respect to semantic similarity of word utterances. These models tend to concentrate word representations in hypercones within multidimensional space, a phenomenon known as the representation degradation problem (Ethayarajh, 2019; Gao et al., 2019; Li et al., 2020; Wu et al., 2020; Cai et al., 2021). It is argued that the underlying cause of this limitation stems from the optimization challenges encountered with low-frequency words within extensive vocabularies, a common attribute of natural language generation tasks (Gao et al., 2019). The stochastic nature of the training process and the minuscule likelihood of sampling rare words in any given mini-batch lead to a situation where these words' embeddings are not optimally refined. This lack of refinement causes the embeddings to cluster within a narrow cone in the multidimensional space, severely restricting the model's expressive capabilities. Consequently, while contextual models excel as Language Models, their effectiveness in text representation within a semantic space is limited. In other words, although neural networks can predict words based on preceding sequences and classification labels, the embedding space used to represent texts does not align coherently with their meanings.

To address this issue, some approaches, such as Sentence-BERT (Reimers and Gurevych, 2019b) and the Universal Sentence Encoder (Cer et al., 2018), train networks on sentence pairs as a similarity classification task. However, the effectiveness of these models may decline when applied to texts with characteristics different from the units on which they were trained. Notably, (Raffel et al., 2020) found that supervised transfer learning from multiple tasks does not outperform unsupervised pre-training. (Yogatama et al., 2019) conducted an extensive empirical investigation to evaluate SOTA Natural Language Understanding models, exploring the task-independence of the acquired knowledge during the learning process. They concluded that model performance is sensitive to the choice of supervised training task. Other experiments conducted through probes suggest that Neural Language Models fail to capture the systematic nature of language (Talmor et al., 2020; Pimentel et al., 2020; Goodwin et al., 2020; Hupkes et al., 2020; Bender and Koller, 2020). In these probing experiments, researchers select a linguistic task and train a supervised model to predict annotations for that task using the network's learned representations. In summary, although Neural Language Models are remarkably powerful, they alone cannot adequately represent previously unseen textual information through composition.

The objective of this paper is to examine the semantic similarity in song lyrics and explore its integration into a subsequent song recommendation system. Initially, we will explore unsupervised composition functions that combine static and contextual word vectors. Furthermore, we will investigate the semantic composition in lyrics with varying degrees of granularity, such as sentences, stanzas, or entire songs. Additionally, we will study the similarity between songs as a downstream task through supervised approaches.

2.2.1 Information-Theoretic Compositional Distributional Semantics

Following the thread of compositional and distributional semantics, where the former emphasizes the structured, rule-based construction of meaning from the syntactic combination of elements, and the latter draws meaning from the contextual usage of words, as discussed earlier, we leverage a method that aims to marry these paradigms. This approach is grounded in the Information-Theoretic Compositional Distributional Semantics (ICDS) framework (Amigó et al., 2022), which provides us with the *lingua franca* to describe many of the methods presented throughout this research, offering a rich theoretical background that revolves around the Shannon Information Theory. This theory facilitates the derivation of representation properties, similarity functions, and a parameterizable generalization of these functions.

ICDS is predicated on the principle that the meaning and information content of linguistic units can be quantified and analyzed through a formal computational framework. It is formalized as a tuple of three critical functions: embedding, composition, and similarity. These functions collectively aim to capture the semantic richness and informational specificity of language in a mathematically rigorous manner. The foundational hypothesis of ICDS posits that there exist minimal linguistic units, the semantics of which are determined by their contextual usage, while their information content is directly related to their specificity. This perspective aligns with the notion that language's systematic nature can be effectively captured through compositional mechanisms, which simultaneously preserve the informational content of composite utterances. Such an approach underscores the balance between the discrete and combinatorial aspects of language semantics.

At the core of ICDS lies the embedding function $\pi : S \to \mathbb{R}^n$, where S represents the space of basic linguistic units, and $\mathbb{R}^n$ denotes the n-dimensional real vector space. For any basic linguistic unit $x \in S$, the function $\pi(x)$ yields a vector representation that encapsulates both the semantic and informational essence of x. In this context, the functions derived from language models serve as the practical instantiation of π, thereby bridging theoretical constructs with empirical NLP methodologies.

In applying the ICDS framework to our study, we propose to view the semantics of a song as the aggregate composition of the meanings of its parts, considered at various levels of granularity, such as verses, stanzas, and the entire song, as will be further elaborated in Section 4.3 (Experimental Setup: Composing Song Lyrics). This perspective aligns with the ICDS hypothesis that the semantic value of linguistic units is contingent upon their contextual usage, while their informational content correlates with their specificity. The composition function within ICDS allows for the aggregation of these vectorized representations, enabling us to construct a coherent semantic representation of larger textual units, like entire stanzas or songs. This compositional approach is vital for capturing the layered meanings that emerge from the syntactic and structural arrangement of lyrics, reflecting how the sum of a song's parts creates a richer semantic tapestry than the individual components alone. By adopting this framework, we aim to dissect the complex semantic landscape of song lyrics, parsing their meaning through both their individual components and their collective assembly.

Furthermore, the similarity function in ICDS facilitates the comparison of these compositional representations, allowing us to quantify the semantic likeness between different songs. This aspect is particularly relevant for our goal of exploring semantic similarity in song lyrics as a basis for a song recommendation system. By measuring the similarity between songs at different levels of granularity, we can identify thematic and semantic resonances that transcend mere lexical correspondence, tapping into the deeper emotional and narrative connections that songs share.

The uniqueness of ICDS is evident in its insistence that both composition and similarity functions adhere to the embedding's Information Content, typically represented by vector norms. This requirement ensures that the integration of semantic elements through the composition function and their comparative analysis via the similarity function are informed by and consistent with the underlying informational properties encoded in the embeddings.

2.2.2 Embedding Function and its Properties

Having established the theoretical underpinnings and the practical relevance of the ICDS framework in the analysis of song lyrics, the following sections will delve into the mathematical properties that govern the representation and composition functions within this framework, starting with the two properties that affect the embedding function:

- **Information Measurability**: Given a linguistic unit x, the norm of its vector representation is approximately equal to the Information Content (Information Content (IC)) of x, mathematically expressed as:

$$\|\pi(x)\| \approx IC(x) = -\log(p(x))$$

- **Angular Isometry**: There exists an isometry between the angular position of the basic units' representations and their expected similarity according to human perception, described by:

$$\cos(\pi(x), \pi(y)) \propto \mathbb{E}(\text{SIM}(x, y))$$

2.2.3 Composition Function and its Properties

As we transition to the composition function, we outline the specific constraints that the IC of a composite expression must adhere to:

- **Composition with the Neutral Element**: components with null information content (vector norm of zero) do not affect the composition.

$$\|\vec{v}_2\| = 0 \implies \|\vec{v}_1 \odot \vec{v}_2\| = \|\vec{v}_1\|$$

- **Composition Norm Lower Bound**: The norm of the vector of the composite representation is greater than or equal to the norm of each component, that is, the composition never reduces the IC.

$$\|\vec{v}_1 \odot \vec{v}_2\| \geq \|\vec{v}_1\|$$

$$\|\vec{v}_1 \odot \vec{v}_2\| \geq \|\vec{v}_2\|$$

- **Composition Norm Monotonicity**: the norm of the composition vector is monotonic with respect to the angle between the composed vectors.

$$\left\{ \begin{array}{l} \|\vec{v}_1\| = \|\vec{v}_2\| = \|\vec{v}_3\| \\ \cos(\vec{v}_1, \vec{v}_2) > \cos(\vec{v}_1, \vec{v}_3) \end{array} \right\} \implies \|\vec{v}_1 \odot \vec{v}_2\| < \|\vec{v}_1 \odot \vec{v}_3\|$$

- **Sensitivity to Structure**: Given three representations $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$ with equal norm and angularly equidistant, their composition is not associative.

$$\left\{ \begin{array}{l} \|\vec{v}_1\| = \|\vec{v}_2\| = \|\vec{v}_3\| > 0 \\ \cos(\vec{v}_1, \vec{v}_2) = \cos(\vec{v}_1, \vec{v}_3) = \cos(\vec{v}_2, \vec{v}_3) > 0 \end{array} \right\} \implies (\vec{v}_1 \odot \vec{v}_2) \odot \vec{v}_3 \neq \vec{v}_1 \odot (\vec{v}_2 \odot \vec{v}_3)$$

2.2.4 Vector-Based Information Contrast Model

Building on the formulation of the Information Contrast Model for similarity from "On the foundations of similarity in information access" (Amigó et al., 2020), which is a generalization of PointWise Mutual Information, the work presented in ICDS introduces the Vector-Based Information Contrast Model aiming to satisfy the three properties of the similarity function enumerated in the composition section where the publication was explained.

The metric is defined as:

$$ICM^V_\beta = \|\vec{v}_1\|^2 + \|\vec{v}_2\|^2 - \beta(\|\vec{v}_1\|^2 + \|\vec{v}_2\|^2 - \langle \vec{v}_1, \vec{v}_2 \rangle)$$

Properties of the ICM similarity function:

- **Angular Distance Similarity Monotonicity:** Given equal vector norms (same Information Content), the similarity is monotonic, decreasing in relation to the angular distance and the proximity of semantic orientation.

$$\left\{ \begin{array}{l} \cos(\vec{v}_1, \vec{v}_2) > \cos(\vec{v}_1, \vec{v}_3) \\ \|\vec{v}_1\| = \|\vec{v}_2\| = \|\vec{v}_3\| > 0 \end{array} \right\} \implies \delta(\vec{v}_1, \vec{v}_2) > \delta(\vec{v}_1, \vec{v}_3)$$

- **Orthogonal Embedding Similarity Monotonicity:** For a set of independent and orthogonal representations, the greater the norm (their specificity), the lower their similarity.

$$\left\{ \begin{array}{l} \cos(\vec{v}_1, \vec{v}_2) = \cos(\vec{v}_3, \vec{v}_4) = 0 \\ \|\vec{v}_1\| < \|\vec{v}_2\|, \|\vec{v}_3\| < \|\vec{v}_4\| \end{array} \right\} \implies \delta(\vec{v}_2, \vec{v}_4) > \delta(\vec{v}_1, \vec{v}_3)$$

- **Equidistant Embedding Similarity Monotonicity:** Given two pairs of vectors $(\vec{v}_1, \vec{v}'_1)$ and $(\vec{v}_2, \vec{v}'_2)$, with $\vec{\epsilon}$ being a vector representing their equidistance then

$$\left\{ \begin{array}{l} \vec{v}'_1 = \vec{v}_1 + \vec{\epsilon}, \vec{v}'_2 = \vec{v}_2 + \vec{\epsilon} \\ \|\vec{v}_2\| > \|\vec{v}_1\| >> \|\vec{\epsilon}\| \end{array} \right\} \implies \delta(\vec{v}_2, \vec{v}'_2) > \delta(\vec{v}_1, \vec{v}'_1)$$

2.2.5 Generalized Composition Function

Concluding the discourse on the ICDS framework, we introduce the generalized version of the composition function defined in ICDS and employed in this research. This generalization is expressed through the function $F_{\lambda,\mu}$, defined as follows:

$$F_{\lambda,\mu} = \frac{\vec{v}_1 + \vec{v}_2}{\|\vec{v}_1 + \vec{v}_2\|} \cdot \sqrt{\lambda(\|\vec{v}_1\|^2 + \|\vec{v}_2\|^2) - \mu\langle\vec{v}_1, \vec{v}_2\rangle}$$

The initial term on the right side of this equation represents the unit vector resultant from the summation of the two vectors, thereby dictating the direction of the composite vector. The subsequent term specifies the norm (or magnitude) of this vector, which is contingent upon both the norms of the individual vectors and their dot product. This formulation offers a generalization of prevalent composition functions found in existing literature, such as the sum and average of vectors, allowing for the adjustment of the composition's characteristics through the parameters μ and λ.

Chapter 3

Hypothesis and Research Questions

The focal point of this dissertation is the exploration and estimation of semantic similarity in Spanish song lyrics through a nuanced approach leveraging unsupervised word embedding composition functions alongside supervised fine-tuning methodologies applied in LLMs. Unsupervised learning discovers patterns or structures from unlabeled data; in contrast, supervised learning involves training a model on a labeled dataset, where each example is paired with an output label. This research hypothesizes that:

> *Utilizing neural networks in NLP to analyze and draw from the semantic content present in song lyrics, by leveraging existing information from related domains, can significantly enhance music recommendation systems, offering a more enriched, lyric-driven approach that outperforms current methods.*

We address several research questions that are poised to drive the experimental enquiries discussed in the remainder of this article.

RQ1 Could word embeddings unsupervised semantic composition prove a viable method for resolving the song recommendation problem, particularly when viewed as a contributing factor within a broader song recommendation system?

RQ2 In tackling the task of unsupervised semantic composition, do static or contextual word embeddings offer a more effective approach? This composition process aims to encapsulate sentence semantics through an Information Theory-based Compositional Distributional Semantics approach.

RQ2.1 When utilized in tandem with unsupervised semantic composition, do static embeddings, which maintain a certain level of semantic isometry with the ideal meaning space, enable the resolution of the classification problem of song similarity?

RQ2.2 When paired with unsupervised semantic composition, would the contextual word embeddings, even though their representations do not preserve semantic isometry, allow for effective song similarity classification?

RQ3 In both the cases of static and contextual word embeddings, would domain-specific transfer learning offer benefits to the classification of song similarity?

RQ4 Regarding the supervised methodology, which centers on the fine-tuning of pre-trained models, which strategy yields superior results for song similarity classification? How would these supervised methods compare to the unsupervised approach?

This research aims to validate this hypothesis through a comprehensive exploration of various semantic composition methods and sentence similarity metrics, coupled with different supervised training strategies involving cross-encoders and bi-encoders, and evaluating their efficacy in identifying semantically similar song lyrics. These supervised strategies, leveraging recent advances, like the self-attention mechanism, have defined the SOTA in tasks requiring fine-grained understanding of text pairs, such as question-answering, text similarity or relevance ranking (Reimers and Gurevych, 2019a; Devlin et al., 2019).

Regarding our annotation experiment, we utilized a 6-point numeric scale akin to that developed for the SemEval tasks Agirre et al. (2012). Similar to the structure employed in the original SemEval tasks, our scale includes a specific level for complete dissimilarity (level 0) and five additional levels (levels 1-5) to delineate varying degrees of semantic similarity, arranged in ascending order of intensity. Despite the potential for nuanced analysis offered by this scale, the decision was made to proceed with a binary classification of song lyrics as either similar or dissimilar for several compelling reasons rooted in both our data analysis and relevant literature.

The distribution of the annotations across the 6-point scale revealed a notable concentration of data at the least similar point, with intermediate and upper levels seeing fewer examples. This pattern, coupled with the total sample size available for our study, suggested that binary classification would provide a firmer foundation for both training and evaluating our models due to the more balanced distribution of data it afforded. Furthermore, when examining inter-annotator agreement, it became evident that consensus was more readily achieved at the scale's extremes. Annotators found it relatively straightforward to identify texts as clearly similar or dissimilar, whereas agreement on the finer distinctions of intermediate similarity levels proved more elusive. This variation in agreement underscores the challenges inherent in maintaining consistency and reliability across annotators when dealing with a scale that offers more nuanced options.

The literature on scale design and annotation complexity also supports the simplification of the scale. Studies have indicated that scales with too many options can introduce cognitive strain, leading to decreased accuracy and consistency among annotators (Tversky, 1977; Nowak and Rüger, 2010). The challenge of making precise distinctions, particularly for assessments as subjective as semantic similarity, is exacerbated by a larger number of scale points. If the scale's complexity has led to inconsistent

annotations, consolidating similar labels after the fact could help mitigate the impact of any potential annotator errors. This approach can streamline the dataset, thereby enhancing the signal-to-noise ratio for learning algorithms. Reducing the number of categories not only simplifies the task for future annotations but also refines the existing dataset, which may result in improved model performance due to the increased clarity and consistency in the training data.

In light of these considerations, transitioning to a binary classification system emerged as a strategic choice aimed at bolstering the robustness and clarity of our analysis.

We also explore both supervised and unsupervised approaches. In the supervised realm, we try both cross-encoder and bi-encoder techniques. In the unsupervised context, we look into non-supervised semantic composition approaches based on static and contextual embeddings. Within these methods, we examine the performance of both multilingual and monolingual pre-trained models, as well as domain-adapted and general-domain models. Multilingual language models, are trained on datasets comprising multiple languages. These models are designed to understand and process information across linguistic boundaries, offering the flexibility to work with texts in various languages without the need for separate models. Monolingual language models, on the other hand, are trained exclusively on datasets in a single language, enabling them to specialize and deeply understand the linguistic nuances, idioms, and syntax specific to that language. This specialization often results in higher performance on NLP tasks within the same language because the model has a focused knowledge base that aligns closely with the intricacies and unique characteristics of the language it was trained on (Agerri et al., 2020; Agerri and Agirre, 2023; Armengol-Estapé et al., 2021; Martin et al., 2020). In the context of analyzing Spanish song lyrics, a monolingual model trained on Spanish texts is expected to capture the subtleties of Spanish language usage more accurately than its multilingual counterparts.

We expect this comprehensive approach to enhance our understanding of semantic similarity in song lyrics. This endeavor is, nonetheless, not just an academic exercise; it has tangible implications for the development of song RSs.

Chapter 4

Experimental Setup

Building on the comprehensive exploration of semantic similarity within song lyrics outlined in the previous sections, this chapter delves into the experimental setup designed to validate our hypothesis and address the research questions (RQs) posited at the outset of this study. The foundational hypothesis—that various semantic composition methods and sentence similarity metrics, and advanced supervised training strategies involving cross-encoders and bi-encoders, can significantly enhance the efficacy of identifying semantically similar song lyrics—sets the stage for a detailed examination of the methodologies employed in our research.

Our objectives are twofold: first, to assess the relative performance of different semantic composition methods in capturing the nuanced semantic similarities between song lyrics; and second, to evaluate how various sentence similarity metrics and supervised training strategies can improve the identification of semantic similarities within this unique textual domain. The experimental setup, designed to address these objectives, incorporates a blend of both supervised and unsupervised learning strategies, leveraging the latest advancements in NLP such as self-attention mechanisms that have redefined the SOTA in fine-grained text analysis tasks.

To systematically approach these objectives, we structure our experiment around several key components: the selection and preparation of a diverse dataset of song lyrics, the deployment of pre-trained and fine-tuned models encompassing both cross-encoders and bi-encoders, and the application of a binary classification scheme grounded in our preliminary data analysis and literature review. This binary classification not only aligns with the distribution patterns observed in our annotated data but also simplifies the model training and evaluation process without compromising the depth of semantic analysis required for our study.

Moreover, the exploration extends to contrasting the effectiveness of monolingual and multilingual models within the unsupervised framework, offering insights into the influence of language-specific nuances on the task of semantic similarity detection. The decision to employ both static and contextual word embeddings, derived from established methodologies like the Continuous Bag of Words (CBOW) and Skip-gram models, as well as the more recent transformer-based models like BERT and RoBERTa, encapsu-

lates our comprehensive approach to understanding the multifaceted nature of semantic composition in song lyrics.

As we move forward in this chapter, we will outline the specific experimental conditions, the dataset preparation and annotation process, the selection criteria for the models and embeddings, and the evaluation metrics employed to assess model performance. This setup not only aims to validate our hypothesis but also to contribute valuable insights and methodologies to the broader field of NLP, with potential implications for the development of more effective semantic similarity detection systems across various domains of text.

4.1 Word Embeddings

Word embeddings represent a fundamental concept in NLP, especially in the context of applying neural networks to language tasks. They involve mapping each word w_i from a vocabulary V to a dense vector representation x_i. These vectors, known as word embeddings, capture the semantic and syntactic properties of words in a high-dimensional space. The collection of all such vectors forms the word embedding matrix $\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{|V| \times d}$ with $|V|$ representing the vocabulary size and d denoting the dimensionality of the embeddings.

In practice, an input text sequence $w_1, w_2, \ldots, w_T$ is converted into a sequence of corresponding word embeddings $x_1, x_2, \ldots, x_T$. This sequence serves as the input to a neural network, allowing it to process and understand the textual information. Word embeddings provide several key advantages over sparse representations like one-hot encoding, including semantic richness, better generalization capabilities, and computational efficiency.

These embeddings can be obtained in multiple ways. Two widely recognized methods involve using pre-trained embeddings and learning embeddings specific to a task. Pre-trained embeddings, such as GloVe and Word2Vec, are learned from large external corpora and provide a general representation of linguistic properties. These will be discussed in greater detail in subsequent sections. The rationale behind utilizing pre-trained embeddings lies in the principle of transfer learning, which offers a solution to the limitations of the traditional supervised learning paradigm when there is insufficient labeled data for the desired task or domain. Transfer learning addresses this challenge by leveraging data from a related task or domain, known as the source task and source domain. The knowledge gained from solving the source task in the source domain is stored and applied to the target task and domain. This process is depicted in the transition from the traditional supervised learning setup to the transfer learning setup. The core objective of transfer learning is to learn the target conditional probability distribution in the target domain, utilizing the information gleaned from the source domain and source task (Ruder, 2019).

Pre-trained embeddings are a manifestation of transfer learning in the context of NLP. By using embeddings that have been pre-trained on large, diverse corpora, we can transfer the general linguistic knowledge captured by these embeddings to our specific

task or domain, even when labeled data is scarce. This approach enables the model to benefit from a rich, pre-existing understanding of language, facilitating better performance on the target task through the nuanced representation of words and their relationships.

In the context of this thesis, we will also explore the use of transformer models for generating word embeddings. Transformers represent a significant advancement in the field of NLP and offer a dynamic approach to generating context-sensitive embeddings. The specifics of using transformers, along with GloVe and Word2Vec embeddings, will be elaborated in the following sections, highlighting their application and relevance to our research.

4.1.1 Static Word Embeddings

In our study, we employ two fundamental methods to obtain static vector representations of words: the CBOW and Skip-gram architectures from Word2Vec, and the GloVe model. These methods are crucial for generating embeddings that capture the linguistic context and semantic relationships of words in our corpus.

Word2Vec Methodologies: Developed by Mikolov et al., Word2Vec encompasses two distinct approaches for creating word embeddings:

- **Continuous Bag of Words (CBOW)**: This method involves predicting the current word based on its context. Our implementation of CBOW aims to maximize the following objective function:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t|w_{t-1}, w_{t+1})$$

  Here, w_t denotes the current word, with w_{t-1}, w_{t+1} representing the surrounding contextual words. The model's focus is on understanding the likelihood of a word given its surrounding words.

- **Skip-gram**: In contrast to CBOW, Skip-gram predicts the context given a word. This approach is effective for capturing a broader range of word associations. The objective function for Skip-gram in our study is formulated as:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t)$$

  where w_t represents the current word, and w_{t+j} are the context words.

Both CBOW and Skip-gram are implemented as shallow, two-layer neural networks, trained to reconstruct linguistic contexts of words effectively.

GloVe Model: Another method we utilize is GloVe, developed by Pennington et al. (2014). GloVe combines global and local statistical information from the corpus

to produce word embeddings. Our application of GloVe involves constructing a co-occurrence matrix X, where X_{ij} indicates the frequency of word j appearing in the context of word i. The objective of GloVe in our context is to minimize the following function:

$$J(\theta) = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2$$

Here, V represents the vocabulary size, w_i and $\tilde{w}_j$ are word vectors, and b_i, $\tilde{b}_j$ are the respective bias terms. This method effectively captures both the local context and global statistical properties of words in our dataset.

4.1.2 Contextual Word Embeddings

Model Name	**No. of Parameters**	**Language**	**Model Architecture**
BERT base	110M	Multilingual	BERT
Alberti base	110M	Multilingual	BERT
Bertin base	125M	Spanish	RoBERTa
MarIA base	125M	Spanish	RoBERTa
MarIA large	355M	Spanish	RoBERTa
STSB	270M	Multilingual	XLM-RoBERTa

Table 4.1: Model Comparison: Parameters and Architecture

Contextual word embeddings, as a key innovation in natural language processing, have been revolutionized by the development of transformer models. Unlike static embeddings like Word2Vec and GloVe, which produce a fixed embedding for each word, contextual embeddings provide dynamic representations that change based on the surrounding text. This ability to capture linguistic context is crucial for understanding the complexities of language, including polysemy and varied syntactic arrangements.

Transformers, introduced by Vaswani et al. (2017a), are central to this advancement. Their unique feature, the self-attention, is a mechanism where the relevance of different tokens is dynamically assessed through learned weights. This relevance is quantified by the attention scores, derived from the dot product of Query (Q) and Key (K) vectors, followed by a softmax operation to ensure these scores sum to one. The attention scores are then used to create a weighted sum of Value (V) vectors, producing the output representations that are contextually aware, effectively modulating the representation of token embeddings as they propagate through the model's layers.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The Transformer architecture typically relies on multiple attention heads within the self-attention mechanism, each with its own set of learnable parameters. These multiple

heads allow the model to simultaneously process token representations from different perspectives, subdividing the embedding space into as many heads as there are, and then concatenating these processed representations before passing them to the feed-forward layer.

In addition to this self-attention mechanism, transformers incorporate positional encodings to account for the order of tokens in the sequence, paramount for understanding language structure and syntax, as it allows the model to recognize patterns and relationships that depend on the sequence of words or phrases. The incorporation of positional information with attention-driven contextual embeddings ensures that the model's output is not only contextually aware but also structurally informed, providing a robust framework for generating and understanding complex linguistic constructs.

In the realm of transformers, we specifically leverage encoder models, shown in Table 4.1. Unlike decoders or generative models that learn the joint probability distribution $P(x, y)$, encoders are designed to learn the conditional probability $P(y|x)$ directly from the raw data. This approach is particularly advantageous in tasks where the goal is to understand or classify input data rather than generate new data. For instance, BERT, a notable transformer-based model, uses multiple layers of encoders. It processes the entire input sequence in one go, allowing each word to be contextualized based on its complete surrounding context, both preceding and following.

This bidirectional nature of BERT and similar models represents a significant enhancement over earlier unidirectional models. The ability to consider the full context of a sentence enables a more accurate and nuanced representation of language, significantly improving performance on various complex NLP tasks. The embeddings from transformer encoders are profoundly contextual, varying in accordance with the input sentence. As a result, they offer a more detailed and precise understanding of language.

Training these transformer models involves refining both the embeddings and the self-attention weights through backpropagation and gradient descent methods. This training process ensures that the model effectively captures and represents the contextual relationships intrinsic to the text.

In summary, the use of encoder models in transformers for generating contextual word embeddings marks a paradigm shift in natural language processing. By employing advanced deep learning architectures and self-attention mechanisms, these models provide sophisticated, context-aware representations of language, surpassing the static embedding techniques and enhancing the ability to interpret complex linguistic patterns.

4.2 Monolingual and Multilingual models

The unprecedented advances in NLP have underscored the pivotal role of large pretrained language models, developed and trained on extensive corpora through resource-intensive processes predominantly by major corporations. This trend has largely favored English, resulting in the most advanced language models for English being publicly released by these entities. For languages other than English, the question of performance equivalence arises, especially when considering the multilingual models like multilingual

BERT and XLM-RoBERTa, which purportedly support over 100 languages. Despite their proficiency in high-resource languages, it is observed that monolingual models often outperform their multilingual counterparts in language-specific tasks, as mentioned in Section 3. This discrepancy is attributed to the tailored training designs and corpus selections that cater to the linguistic subtle differences of each language, leading to the development of superior monolingual models.

Given the focus on encoder-only masked language models, and the evident gap in performance between monolingual and multilingual models, our methodology encompasses a comparative analysis of these models for Spanish. This approach is informed by the recent findings, who highlight the nuanced performance differences and advocate for a deeper investigation into the factors influencing these outcomes (Agerri and Agirre, 2023). Our evaluation extends to include prominent multilingual models, alongside monolingual models (detailed in Table 4.1), to provide a comprehensive assessment of their efficacy across a spectrum of Spanish NLP tasks:

BERT models were trained on MLM and next sentence prediction (NSP) objectives. The original MLM implementation relies on random masking of tokens in each input sequence. With a vocabulary of 30,000 tokens, it uses a sliding window approach with a fixed sequence length that can cross document boundaries.

- BERT multilingual base model (cased)[1] (Devlin et al., 2019). The BERT model was pre-trained using a 3.3 billion word corpus composed of the BooksCorpus (800 million words) and Wikipedia (2.5 billion words). The training procedure involved generating input sequences by sampling two spans of text from the corpus, referred to as "sentences". One sentence received the A embedding and the other received the B embedding, with 50% of the time B being the actual next sentence following A and 50% of the time being a random sentence. The input sequences were then tokenized using the WordPiece tokenization with a uniform masking rate of 15%. The training was done with a batch size of 256 sequences and over 1 million steps, approximately 40 epochs, using Adam optimization with various hyperparameters such as learning rate, weight decay, and dropout probability. The training loss was the sum of the MLM likelihood and the NSP likelihood. BERT learned the [SEP], [CLS], and sentence A/B embeddings during pre-training and chose a task-specific fine-tuning learning rate that performed the best on the development set.

- Alberti[2](de la Rosa et al., 2023) is a BERT based multilingual model trained on poetry for stanzas and verses. The model was pre-trained on a large corpus of multilingual poetry datasets, including resources from English, German, Russian and, particularly relevant to this work, Spanish. The pre-training of Alberti is a form of domain adaptation that allows the model to learn and capture specific patterns and features present in poetry datasets, which are not necessarily present in other datasets.

[1]https://huggingface.co/bert-base-multilingual-cased
[2]https://huggingface.co/flax-community/alberti-bert-base-multilingual-cased

RoBERTa (Liu et al., 2019) discards NSP task during pre-training relying solely on MLM and introduces dynamic masking, a method that changes the masking pattern for each training instance. The model was trained on full sentences rather than fixed-length segments. It is worth noting that the model's vocabulary consists of 50,265 tokens, including the line-break character; we used this character to delimit song verses and stanzas.

- Bertin[3] (de la Rosa et al., 2022)

  In this model, it was found that traditional methods of pre-training, which rely on large amounts of data from sources such as Common Crawl, can contain enough noise to produce sub-optimal results. To overcome the challenge of training a language model on a large corpus of Spanish text, the authors used a technique called perplexity sampling to create a smaller subset of the Spanish mC4 dataset for training the model. The mC4 dataset is a multilingual variant of C4, containing natural text in 101 languages from the public Common Crawl web-scrape, and was used to train the mT5 multilingual model. The Spanish portion of mC4 contains about 416 million documents and 235 billion words in approximately 1TB of uncompressed data.

  The Bertin model was trained using the MLM objective, with hyperparameters and setup similar to those in RoBERTa (Liu et al., 2019). The model was trained for 250k steps, with a batch size of 2048 for 128 sequence length and 384 for 512 sequence length. The training was divided into two stages, with 230k steps of training with 128 sequence length, followed by a few more steps of training with 512 sequence length from previous checkpoints. The number of warm-up steps for 512 sequence length was reduced to 500. The training process lasted approximately one week, with MLM accuracy scores reported at the end of training for both 128 and 512 sequence length.

- MarIA[4] (Gutiérrez-Fandiño et al., 2022).

  MarIA was pre-trained on a corpus derived from the National Library of Spain's selective crawls carried out between the years 2009 and 2019, which covers a wide range of themes, relevant events, and domains at risk of disappearing. The corpus was processed to generate 59TB of JSON files containing text extracted from the WARC files, including paragraphs, headers, and hyperlinks' texts. The model was evaluated on 9 tasks, including text classification, Named Entity Recognition and Classification, Paraphrase Identification, Part-of-Speech Tagging, Semantic Textual Similarity, Textual Entailment, and Question Answering. The evaluation datasets include Multilingual Document Classification Corpus, CoNLL-NERC, CAPITEL-NERC, PAWS-X, Universal Dependencies Part-of-Speech, Spanish textual similarity datasets, Cross-Lingual NLI Corpus, and a newly created Spanish Question Answering dataset.

[3]https://huggingface.co/bertin-project/bertin-roberta-base-spanish
[4]https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne

The MarIA model was trained using fine-tuning methodology based on the usual practices in NLP and AI literature. The evaluation was done using the HuggingFace Transformers library and a single linear layer was added to the model being fine-tuned for each task. A grid search was conducted for all models and tasks with the same settings, including batch size, weight decay, learning rate, and epochs, to ensure a fair comparison. The best checkpoint was selected based on the downstream task metric on the development set and then evaluated on the test set.

XLM-RoBERTa (Cross-Lingual Language Model RoBERTa) (Conneau et al., 2020), shares MLM training objective with BERT and RoBERTa, but includes a cross-lingual alignment objective called Translation Language Modeling in which parallel sentences in different languages are concatenated and a mask is applied to tokens in both languages. The vocabulary is also the largest, with 250,000 tokens, utilizing a SentencePiece with BPE tokenizer.

- Sentence-Transformers[5](STSB) (Reimers and Gurevych, 2019a).

  The original model presents a modification of the pre-trained BERT network, known as Sentence-BERT (SBERT), that derives semantically meaningful sentence embeddings. Sentence-Transformer uses a siamese and triplet network structure and can be compared using cosine similarity, reducing the computational overhead from 65 hours with BERT to just 5 seconds while maintaining accuracy. SBERT outperforms other state-of-the-art sentence embedding methods on semantic textual similarity tasks and transfer learning tasks. The model used in this work is a multilingual variation of the original model, that substitutes BERT with XLM-RoBERTa.

  An examination of sentence embeddings utilizing Wikipedia as its corpus, this study leverages the articles to build a significant collection of weakly labeled sentence triplets. The Triplet Objective serves as the framework while SBERT, a cutting-edge sentence embedding technique, is trained on 1.8 million training triplets and evaluated on a substantial 222,957 test triplets.

  To fine-tune the BERT/RoBERTa networks, the authors create siamese and triplet networks and use either a classification objective function, a regression objective function, or a triplet objective function to update the weights so that the produced sentence embeddings are semantically meaningful. SBERT was trained on a combination of the SNLI and Multi-Genre NLI datasets using a 3-way softmax classifier objective function for one epoch. The training details include a batch-size of 16, Adam optimizer with learning rate 2e-5, and a linear learning rate warm-up over 10% of the training data.

This comparison is pivotal, as it not only challenges prevailing assumptions about model performance but also elucidates the potential of multilingual models in contexts previously dominated by monolingual counterparts.

[5]https://huggingface.co/sentence-transformers/stsb-xlm-r-multilingual

4.3 Composing Song Lyrics

In the advancement of our research, the construction of vector representations for song lyrics plays a pivotal role, utilizing the foundational work laid out in the word embeddings and ICDS discussions. By amalgamating distributional representations of linguistic units, the ICDS function offers a means to encapsulate the semantic essence of song lyrics into dense vector spaces, composing texts longer than one word while adhering to the principles of compositionality and contextuality. This process, however, introduces several considerations that significantly impact the quality and applicability of the generated embeddings, notably, the operational dynamics of the ICDS function make it uniquely sensitive to various factors in the composition process. These include the directional flow in which the song lyrics are processed, whether sequentially from start to end or vice versa, and the level of granularity at which the lyrics are segmented for analysis—be it at the level of entire songs, individual stanzas, or discrete sentences. Each choice in this compositional framework affects the resultant vector representations, influencing both their semantic depth and their alignment with the intricate structures of musical lyrics.

Moreover, the employment of embeddings from distinct Transformer layers further enriches our methodology. By extracting embeddings from both the initial and final layers of Transformer models, we aim to investigate the performance implications of leveraging different levels of linguistic abstraction and contextualization inherent in these layers. This exploration is crucial for understanding how varying depths of semantic processing contribute to the effectiveness of our models in capturing the nuanced fabric of song lyrics.

4.3.1 Composition Function

Following our exploration of the ICDS framework in Section 2.2 (Related Work: Text Representation), where we delved into embedding functions and their properties, as well as the foundational principles guiding our semantic analysis of song lyrics, we now turn our attention to the generalized composition function. As introduced in Section 2.2.5 (Text Representation: Generalized Composition Function), the norm of the composite vector is adjusted according to a combination of the individual vectors' norms and their dot product, balanced by the parameters λ and μ. These parameters are pivotal in modulating the composition's sensitivity to the magnitude and orientation of the constituent vectors, embodying a flexible approach to semantic composition.

The intricate interplay between the aforementioned parameters delineate distinct methods that provide unique strategies to vector composition:

- F_{sum}: The summation of the basic unit vectors.
- F_{avg}: The average of the basic unit vectors.
- F_{ind}: Operates under the presumption that the combined linguistic forms are statistically independent.

- F_{joint}: Represents the conjunction of the ICs.
- F_{inf}: Added as it fits within the theoretical framework satisfying the earlier described properties.

For each of these methods, utilized in this work, the values of λ and μ are as given in Table 4.2:

Method	λ	μ
F_{sum}	1	-2
F_{avg}	$\frac{1}{4}$	$-\frac{1}{2}$
F_{ind}	1	0
F_{joint}	1	1
F_{inf}	1	$\frac{\min(\|\vec{v}_1\|,\|\vec{v}_2\|)}{\max(\|\vec{v}_1\|,\|\vec{v}_2\|)}$

Table 4.2: ICMβ Composition Functions

4.3.2 Composition Direction

The ICDS function employed for semantic composition is sensitive to the direction in which it is applied. Specifically, two directions are considered: left-to-right and right-to-left. The directionality of composition affects the weightage of various parts of the song during embedding generation. For instance, a right-to-left composition may emphasize the beginning of the song, while a left-to-right composition could highlight its conclusion. Understanding the impact of composition direction offers insights into how different aspects of a song contribute to its overall semantic representation.

4.3.3 Granularity

Granularity refers to the level of segmentation applied to songs before they undergo semantic composition through the ICDS method. We consider three primary units of granularity: song level (entire lyrics), sentences, and stanzas. Each unit offers a distinct lens through which the semantic landscape of a song is explored, influencing the model's capacity to discern and represent both global thematic elements and localized semantic nuances.

Lyrics At the song level, the focus is on capturing the overarching thematic and emotional constructs that define the entirety of a song. This global perspective aims to synthesize the song's comprehensive narrative, offering insights into the dominant themes and sentiments that pervade the lyrics as a whole.

Stanzas Moving to a finer granularity, analyzing songs at the stanza level allows the model to hone in on specific sections of a song, each potentially encapsulating distinct thematic or narrative shifts within the broader context of the song. This level of analysis is particularly adept at uncovering the local semantic nuances that contribute to the song's overall meaning, enabling a more nuanced interpretation of its lyrical content.

Sentences Analyzing the song at the sentence level introduces the finest granularity, where each sentence is examined for its semantic significance within the song. This detailed approach is crucial for detecting subtle variations in tone, perspective, or thematic elements that sentences uniquely convey. It offers an in-depth comprehension of the song's dynamic semantic architecture, revealing the nuanced interplay of language and meaning that shapes the overall narrative.

This systematic exploration of granularity enables us to investigate the trade-offs between local interpretability and global coherence in the context of song similarity detection. By varying the level of granularity, we gain insights into how different segments of a song contribute to its semantic identity, thereby informing our approach to modeling song similarity in nuanced and contextually aware manners.

4.3.4 Transformer Layer

The selection of specific transformer layers for extracting embeddings is informed by the recognition of the representation degradation problem, where contextual models tend to cluster word representations into hypercones within the multidimensional space, thereby limiting the isometry essential for maintaining semantic similarity across word utterances. Works on this problem have suggested the enhanced performance achieved by combining outputs from various layers, rather than solely relying on the last hidden state (Li et al., 2020).

Transformers, particularly through their self-attention mechanism, multi-head attention, and multi-layer architecture, learn diverse representations that evolve across the model's layers. Initial layers tend to capture more about the general meaning of words (less contextual), whereas later layers incorporate progressively more context into these representations. This layered learning process underlines the importance of carefully selecting which layers to utilize for embedding extraction in our methodology. This insight has motivated us to explore the potential of both the initial pre-trained embedding layer and the output of the last layer, aiming to capture a richer, more nuanced representation of text that integrates both foundational linguistic properties and advanced contextual subtleties and facilitates a detailed comparative analysis.

The pre-trained embedding layer offers a broad, general representation of linguistic properties, capturing the foundational semantics of words before contextual influences are applied. This layer is instrumental for establishing a solid baseline of semantic understanding that is not overly specific to any particular context. Conversely, the output of the last layer, refined through successive transformations within the model, embodies a highly contextualized understanding of the text. It integrates the cumulative

insights gained from the model's deep processing, making it invaluable for capturing the nuanced, context-specific semantics essential for tasks like song lyric composition and similarity detection.

4.3.5 Considerations on Special Tokens

As delineated in Section 4.1.2 (Word Embeddings: Contextual Word Embeddings), our study employs encoder transformer blocks in the case of contextual models. These models inherently utilize a set of special tokens during their training phase, each serving distinct functions. Notably, the CLS and $< s >$ tokens are employed to mark the commencement of a text sequence and act as a classification embedding, encapsulating the entirety of the sequence's information. Similarly, SEP and $< /s >$ tokens function as separators or denote the end of a sequence. These tokens are present across all contexts, fulfilling a structural role and are invariably fed into the model.

In our investigative efforts, we conducted experiments to ascertain the impact of semantic composition with ICDS under various configurations: one that includes all special tokens in the sequence, another that omits special tokens during inference to solely obtain word or subword representations, and a third approach that provides the model with all tokens but removes the special tokens prior to executing operations for composite vector derivation. The experimental outcomes revealed negligible differences across these configurations, with any variations falling well within the margin of error. For illustrative clarity, a figure detailing these findings is presented in Appendix A.4.

Based on these observations, we opted for the third configuration. This decision stems from our recognition of the importance of presenting the model with input structured as it has been trained to expect, inclusive of the special tokens, yet removing them post-inference to extract an embedding solely composed of vectors representing words or subwords. It is pertinent to note that this decision does not affect non-contextual models.

4.4 Sentence Similarity Metrics

In the scope of this project, where sets of phrases are transformed into vectors in an n-dimensional space, semantic similarity is a metric in which the distance between them is based on how similar their semantic content is. This section introduces the two metrics utilized in this development.

4.4.1 Cosine Similarity

Cosine similarity is a fundamental metric in the field of NLP, offering a measure of similarity between two non-zero vectors in an inner product space. This metric quantifies the cosine of the angle between two vectors, thus capturing their directional alignment rather than their magnitude. The value of cosine similarity ranges within $[-1, 1]$, with a value of 1 indicating identical orientation, 0 denoting orthogonality, and -1 suggesting diametrically opposed directions.

Mathematically, cosine similarity is defined as:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B are the vector representations of the text units being compared, $\cdot$ denotes the dot product, and $\|A\|$ and $\|B\|$ are the Euclidean norms (magnitudes) of the vectors.

In the realm of NLP, cosine similarity is particularly valued for its ability to effectively measure the semantic proximity between documents, sentences, or words represented as vectors in high-dimensional spaces. This is especially relevant in applications such as document clustering, information retrieval, and similarity-based recommendation systems, where the semantic relatedness of textual entities plays a crucial role.

One of the particularities of cosine similarity is its insensitivity to the overall magnitude of the vectors, focusing solely on their orientation. This characteristic makes it exceptionally useful in comparing documents of varying lengths. Moreover, cosine similarity is a core component in the operational framework of many vector space models, including TF-IDF (Term Frequency-Inverse Document Frequency) and word embeddings like Word2Vec and GloVe.

In summary, cosine similarity provides a robust and intuitive metric for assessing semantic relationships in text data, enabling researchers and practitioners to uncover meaningful insights into the structure and dynamics of natural language.

4.4.2 Vector-Based Information Contrast Model

Building on our exploration of semantic similarity metrics within the domain of natural language processing, we utilize ICM, as delineated in Section 2.2.4 (Text Representation: Vector-Based Information Contrast Model). In contrast with cosine similarity, this metrics takes into account both, the magnitude of the vectors, as well as their angle. The ICM represents a significant evolution from the traditional Point-Wise Mutual Information (PMI) model by introducing three parameters: α_1, α_2, and β. These parameters enhance the model's flexibility, allowing for a more nuanced representation of semantic relationships between linguistic units.

ICM is characterized by its ability to adjust the weighting of the probabilistic events x and y, thus offering a more refined understanding of their informational content. When $\beta = 1$, ICM functions equivalently to PMI, capturing the mutual dependence between x and y. At $\beta = 2$, it approximates the product of conditional probabilities, further extending its applicability to a wider range of semantic analysis tasks. This versatility makes ICM a powerful tool for probing into the intricacies of language, supporting both theoretical inquiries and practical applications in computational linguistics.

A noteworthy aspect of ICM is its generalization of the Linear Contrast Model (Tversky, 1977) under certain conditions, which underscores its theoretical depth and potential for capturing complex semantic phenomena. Furthermore, the vector-based implementation of ICM, $ICM_{V\beta}$, adapts this model to vector space representations, making it particularly suited for contemporary NLP methodologies.

The Information Contrast Model stands out for its robust mathematical foundation and its adaptability to various semantic similarity assessment scenarios. By integrating ICM into our analysis, we aim to leverage its comprehensive approach to better understand and quantify the nuanced semantic relationships that underpin natural language.

4.5 Classifiers

The task of distinguishing between similar and dissimilar pairs of song lyrics necessitates an approach to classification capable of interpreting and quantifying the convoluted semantic relationships embedded within textual data. In this study, we employ two distinct classifiers for the three architectures: the Binary Logistic Classifier and the Cross-Encoder. Each classifier serves a unique purpose in our analysis, chosen for their specific advantages in handling the complexity of natural language data and their suitability for binary classification tasks.

The Binary Logistic Classifier is a foundational tool in our methodology, prized for its simplicity and effectiveness in leveraging distance metrics, such as cosine similarity, to predict binary outcomes. This approach aligns well with the need to quantify the degree of similarity between song lyrics, providing a probabilistic assessment based on the semantic distance between pairs.

Conversely, the Cross-Encoder represents a more sophisticated classification model, designed to directly process and analyze pairs of texts. This model's architecture, incorporating dense layers and non-linear activations, allows for a deeper understanding of the contextual relationships between song lyrics, making it adept at capturing the nuanced differences that distinguish similar from dissimilar pairs.

In our study, we implemented 5-Fold cross-validation to train all classifiers. After completing the training and validation across all folds, we calculated the classifiers' final performance metric by averaging the results from each fold. This approach smoothed out any anomalies specific to a single fold, leading to a more stable and generalizable performance metric

Transformer base model training and supervised fine tuning were implemented using HuggingFace[6] transformer and Sentence Transformers[7] libraries (Reimers and Gurevych, 2019a), employing two architectures: siamese networks as a bi-encoder, and a cross-encoder. In our experiments, we trained on AWS cloud service, and a NVIDIA RTX 4090, performing a grid search for hyperparameter selection. The search space included learning rates of 2e-5, 3e-5, 5e-5, with linear decay and 10% warm-up of the total step count. Epoch counts were 2, 3, 4, with batch sizes of 16 and 32. This training followed best practices proposed in the BERT original paper (Devlin et al., 2019).

[6]http://huggingface.co
[7]https://sbert.net

4.5.1 Binary Logistic Classifier

The Binary Logistic Classifier, in this context, is utilized for binary classification tasks based on the cosine distance between pairs of items. This classifier computes the probability of each pair belonging to a positive class, with the cosine distance serving as the primary feature. The logistic regression model, central to this classifier, employs the logistic function, but the input feature is the cosine distance between two vector representations. The cosine distance, denoted as $d_{\cos}$, is defined as:

$$d_{\cos}(\mathbf{v}_1, \mathbf{v}_2) = 1 - \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}$$

Where $\mathbf{v}_1$ and $\mathbf{v}_2$ represent the vectors of the two items being compared.

The logistic regression model then employs this cosine distance as the input feature ($\mathbf{x}$):

$$p(y = 1|d_{\cos}) = \sigma(\mathbf{w}^\top d_{\cos} + b)$$

In this equation, σ represents the logistic function, $\mathbf{w}$ is the weight vector associated with the feature, and b is the bias term. The objective of the model is to minimize the binary cross-entropy loss during the training process. The loss function for binary classification is defined as:

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log(p(y_i|d_{\cos,i})) + (1 - y_i) \log(1 - p(y_i|d_{\cos,i}))]$$

Here, N is the number of training examples, y_i is the binary ground truth label for the i-th example, and $p(y_i|d_{\cos,i})$ is the predicted probability for the i-th example being in the positive class, as determined by the cosine distance. This model is adept at classifying pairs of items based on their similarity, as quantified by the cosine distance, making it particularly suitable for tasks where relational dynamics between items are pivotal.

4.5.2 Cross-Encoder

The Cross-Encoder is a more advanced classification model for assessing pairwise similarity that directly ingests the pair of songs separated by a special token. Unlike the Logistic Classifier, the Cross-Encoder employs a series of dense layers and a non-linear activation, to capture more complex relationships between the song pairs.

The architecture employed, shown in Figure 4.1, is described as follows:

$$\text{Cross-Encoder} : (\mathbf{x}_1, \mathbf{x}_2) \mapsto y$$

The Cross-Encoder takes the pair $(\mathbf{x}_1, \mathbf{x}_2)$ as input and passes it through the transformer layers. Post this step, the architecture includes a sequence of dense layers and activation functions. The first dense layer, denoted as $\mathbf{D}_1$, is a fully connected layer that

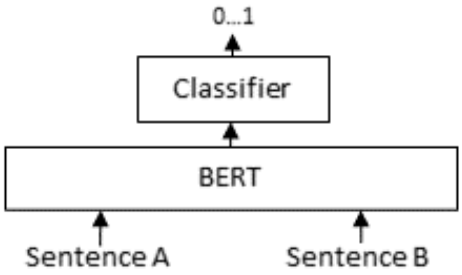


Figure 4.1: Cross-Encoder architecture, (Reimers and Gurevych, 2019a)

transforms the embedding into a different dimensional space. This is followed by a tanh activation function, providing the necessary non-linearity to the model:

$$\mathbf{z} = \tanh(\mathbf{D}_1(\mathbf{v}))$$

Here, $\mathbf{v}$ denotes the embedding obtained from the final hidden state, and $\mathbf{z}$ is the transformed feature vector.

Subsequent to the tanh activation, a second dense layer $\mathbf{D}_2$ is employed. This layer further transforms the feature vector, preparing it for the final classification step:

$$\mathbf{u} = \mathbf{D}_2(\mathbf{z})$$

Where $\mathbf{u}$ represents the output from the second dense layer.

For assessing the class, a softmax function is applied to the output of the final dense layer, converting the raw logits into probabilities, expressed as:

$$\text{softmax}(\mathbf{u})_i = \frac{e^{\mathbf{u}_i}}{\sum_{j=1}^{K} e^{\mathbf{u}_j}}$$

Here, K represents the number of classes, and i indexes a specific class. This results in a probability distribution over K classes for each song pair.

$$p(c|h) = softmax(Wh) \tag{4.1}$$

Similarly to the Binary Logistic Classifier, the loss function to optimize was the cross-entropy loss.

The input song pairs are tokenized, and the classification and separation tokens are added to the input sequences, which are then padded to meet the criteria of a fixed maximum sequence length of 512 tokens. Next, the pre-trained language representations are fine-tuned on the labeled dataset of Spanish song pairs, allowing them to effectively capture the nuances of the Spanish language and learn to identify similarities and dissimilarities between song pairs. For text classification, the final hidden state of the classification token is used as the representation, and a softmax classifier is added on top of the language representations to predict the probability of the label c (either 0 or

1, indicating dissimilar or similar song pairs, respectively), with the trainable parameter matrix *W*. A critical aspect of this architecture is the use of the [CLS] (classification) token, a special token used in transformers like BERT and RoBERTa. The final hidden state of this [CLS] token, after passing through the transformer layers, encapsulates a comprehensive representation of the combined input pair. In contrast, non-contextual models like GloVe lack this mechanism for aggregating contextual information across tokens. Static models generate embeddings for individual words without considering the broader sentence or document context, resulting in representations that are the same regardless of where or how a word is used within the text. This absence of the CLS token and the lack of inherent contextual awareness present a challenge when adapting static models to tasks traditionally suited for contextual models, especially in architectures that rely on aggregated representations for decision-making. Therefore, while it's theoretically possible to adapt static embeddings for use in a cross-encoder setting, this method was primarily employed with contextual models based on the transformer architecture. The parameters of the language representations are updated using an optimization algorithm, AdamW, with the goal of minimizing the loss function over the training dataset.

4.5.3 Siamese Network: Bi-Encoder

The Bi-Encoder fine tuning configuration consists of two parallel neural networks that encode each element of a song pair into separate embeddings. These embeddings capture the linguistic and semantic characteristics of each song, and their similarity is quantified using a cosine similarity metric, as depicted in figure 4.2.

Incorporating the methodology from the Sentence-BERT paper by Reimers et al., our training objective utilizes cosine similarity within a logistic loss function. This setup allows the model to fine-tune the embeddings, enhancing the differentiation between similar and dissimilar song pairs. By optimizing the embeddings to maximize cosine similarity for similar pairs and minimize it for dissimilar ones, the model becomes adept at discerning nuanced semantic relationships.

Our model's adaptation to Spanish song lyrics is a crucial aspect of our approach. The fine-tuning process on a dataset comprising Spanish song lyrics enables the Bi-Encoder to align with the unique linguistic and semantic idiosyncrasies of the Spanish language. This is particularly important given the distinct stylistic and cultural elements prevalent in Spanish music, which might differ significantly from those in more generic language datasets.

A key advantage of our approach is the application of supervised learning, as opposed to unsupervised methods. Leveraging annotated data for song similarity allows the model to have a clear understanding of its performance on specific instances. This direct feedback loop during training fosters a more refined and effective learning process, leading to superior performance in identifying both subtle similarities and differences between songs.

In our study, we employed a logistic classifier to evaluate the performance of models trained using the bi-encoder architecture. The bi-encoder is designed to independently

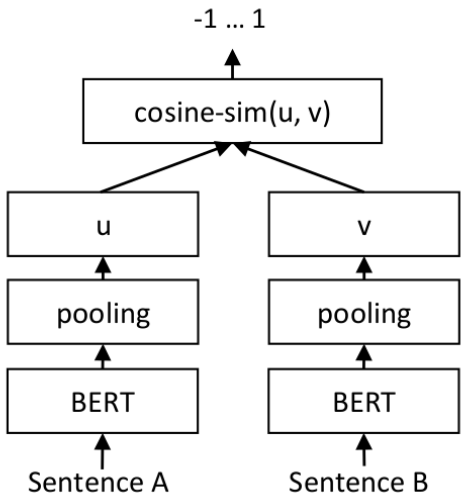


Figure 4.2: Bi-encoder training (Reimers and Gurevych, 2019a)

encode each element of a pair, generating embeddings that capture the semantic nuances of the content. The logistic classifier then uses these embeddings to classify pairs as similar or dissimilar. This process involves calculating the cosine similarity between embeddings and applying logistic regression to predict the likelihood of similarity.

4.6 Datasets

The necessity for models to perform efficiently across different but related domains — as mentioned at the beginning of this section — has become a fulcrum of research and application. At its core, domain adaptation addresses the distributional shift that occurs when the distribution of the data in the target domain differs from that in the source domain, thus leading to a decline in the performance of models when applied to the target domain.

In machine learning, it is conventionally presupposed that training and test datasets adhere to the principle of being independent and identically distributed (i.i.d.), implying that each sample is mutually independent and collectively sourced from a uniform distribution. This foundational assumption, however, may not invariably align with the conditions encountered in practical applications.

To further align our models with the domain of song lyrics, we engage in an extended training utilizing the MLM task. This approach involves selectively masking out tokens in the lyrics and prompting the model to predict the masked words, based on the surrounding context. By customizing pre-trained embeddings for the nuances of song lyrics, this task enables us to explore the potential for improved performance in domain-related tasks. It aims to bridge the gap between the broad linguistic knowledge

of pre-trained models and the unique vernacular of song lyrics, enhancing the model's ability to generate and interpret complex constructs within the target domain.

In the context of our research, domain adaptation emerges as an imperative process, offering a scaffold to build models that are not only proficient in understanding the intricacies of the source domain but are also adept at generalizing this understanding to analyze song lyrics in the target domain with a high level of expertise. By leveraging mathematical formulations and strategies of domain adaptation, our research stands on a robust foundation, paving the way for an exploration that is both deep and mathematically rigorous.

Work on cross-lingual embedding models, indicates that the actual choice of data used for the model to learn a cross-lingual representation space contributes more decisively to performance than the actual underlying architecture (Levy et al., 2017). This insight is particularly relevant to the process of supervised fine tuning and domain adaptation, where the goal is to tailor models to perform well on specific subsets of data, such as adapting from general Spanish text to the specialized domain of Spanish song lyrics. In this section, we elucidate the rationale behind our dataset selection, tailored to meet distinct research objectives, including domain adaptation and classification tasks. Our focus is on outlining the characteristics of each dataset and how the methodology ensures that the data not only provides a robust foundation for training but also aligns with the nuanced requirements of the respective tasks.

4.6.1 Domain Adaptation Dataset

Initially, a comprehensive corpus containing 147,912 songs was curated from a variety of publicly accessible online sources. This dataset spans multiple genres and covers an extensive time range, from 1936 to the present. Distinct from traditional prose, a song lyric exhibits a unique structural form that more closely aligns with poetic compositions. In this structure, lyrics are partitioned into smaller units termed stanzas, which serve as autonomous semantic and prosodical entities. Each stanza is further subdivided into lines that could either be complete sentences or fragments of an elongated sentence. Within our corpus, these stanzas are encoded as arrays of string elements, which are subsequently nested within a higher-dimensional array to represent a complete song.

To achieve a high-quality and non-redundant dataset, we executed a series of data cleansing steps. This included the removal of duplicate entries, empty strings, and single-word songs. Special characters, artist names, song titles, and blank lines were also eliminated. The resulting dataset was segmented into stanzas to better facilitate model training. We also incorporated a strategy to preserve line break characters, aimed at supporting models that depend on such delineations. This was achieved without compromising the tokenization process for models employing different types of tokenizers, such as BertTokenizers.

To enhance the quality of the input data, especially in the context of stanzas, we conducted an additional round of deduplication on the segmented dataset. This was to manage recurring elements like choruses that might otherwise be over-represented in the training data.

4.6.1.1 Data Preparation Through Fixed Sized Chunks

In the chunking approach, a non-overlapping sliding window technique was applied to create training samples that could cross document boundaries. This method involved selecting chunks of text with a length of 128 tokens for 90% of the training data, and 512 tokens for the remaining 10% of the data, so the model could learn long distance relations.

4.6.1.2 Preserving Lyric Integrity

The second approach, stanzas/lyrics, involved feeding the models with stanzas for 90% of the training data and entire song lyrics for the remaining 10%. This strategy aimed to preserve the document boundaries and inherent structure of the songs, as stanzas and lyrics often exhibit unique semantic and syntactic relationships. By maintaining these boundaries during the training process, the models were expected to better learn long-distance relations and capture the distinct characteristics of Spanish songs.

4.6.2 Fine-tuning Annotated Dataset

For our experimentation, we will utilize LyricSIM (Benito-Santos et al., 2023), a dataset tailor made for this task and composed of 676 pairs of songs annotated according to varying degrees of semantic similarity, which are demarcated into six distinct levels on the Likert scale. These levels range from 'Completely different (0)', denoting a complete dissimilarity in lyrics, to 'Outstanding similarity (5)', where lyrics share the same message, emotions, intentions, and lyrical situation, differing only in lexicon and genre. Between these two extremes, four levels capture an increasingly nuanced range of similarity, acknowledging minor aspects without semantic importance at level 1, thematic relationships at level 2, basic similarity in message and feelings at level 3, and substantial similarity with minor variations in lyrical situations and literal meaning at level 4. Emulating the structure utilized in original SemEval tasks (Agirre et al., 2012), the scale dedicates one level to total dissimilarity (level 0), while the remaining five levels encapsulate a spectrum of semantic similarity (levels 1-5) of increasing intensity. However, we have modified the category descriptions to accommodate the broader context of similarity among song lyrics.

- **Completely different (0)**: the lyrics are entirely dissimilar.
- **Barely any similarity (1)**: the lyrics share minor aspects without semantic importance, such as language style or sociocultural context.
- **Little similarity (2)**: there is no semantic similarity (lyrical situation, message, feelings), but the lyrics can be considered thematically (literal meaning) related.
- **Basic similarity (3)**: the lyrics resemble each other in message, feelings of the protagonist/singer, lyrical situation, or literal meaning.

- **Notable similarity / missing details (4)**: the lyrics share the same message and feelings but differ in lyrical situations and/or literal meaning.

- **Outstanding similarity (5)**: the lyrics share the same message, emotions, intentions, and lyrical situation, differing only in lexicon and genre.

In order to generate a thorough dataset of similarity annotations for Spanish song lyrics that covers various dimensions of lyrical content, we employed a crowdsourcing platform. From a total pool of 63 annotators, selected participants engaged in the annotation task. Following data collection, we garnered over 8,325 pair-wise similarity values that represented the evaluation of 2,775 pairs by three different participants each, detailed in Table 4.3.

dataset	rating	count	percent
Original	0	3058	36.73%
	1	3014	36.20%
	2	1058	12.71%
	3	746	8.96%
	4	347	4.17%
	5	102	1.23%
Filtered	0	837	41.27%
	1	705	34.76%
	2	360	17.75%
	3	88	4.34%
	4	34	1.68%
	5	4	0.20%

Table 4.3: Number of ratings in the dataset after refinement.

Aware of the need for reliable data, we set up strict filtering criteria to lower rating variability and improve the dataset's accuracy. This careful refinement process was crucial for making the dataset more useful, leading to a significant increase in agreement between annotators. We employed Krippendorff's reliability alpha as the metric for assessing annotator concordance, achieving a notable score of 0.90. This marked improvement from the preliminary score of 0.27 prior to refinement underscores the efficacy of our dataset curation efforts.

For identifying dissimilar pairs, we focused exclusively on those instances where all three annotators unanimously determined a pair to be dissimilar, assigning it a score of 0. This process yielded a total of 837 pairwise dissimilarities. Regarding similarities, our criteria targeted pairs that garnered exact agreement from two out of three annotators on a specific score. To ensure the inclusion of only the most reliable cases, we opted to omit any pairs where the third annotator's score deviated significantly (by a difference

of 2 or more points from the most common score), leading to a selection of 676 similarity pairs, which represents 24.36% of the initial dataset. Consequently, the curated dataset comprises 2,028 annotations reflecting both similarity and dissimilarity judgments across 75 unique song lyrics.

Such a high degree of reliability not only attests to the dataset's quality but also validates its potential as a foundational resource for advancing research into the semantic nuances of song lyrics. Through this refined dataset, our study aims to delve into the multifaceted semantic landscapes of musical texts, offering novel insights into how similarity in lyrical content is perceived and quantified.

4.7 Evaluation Metric

For evaluating the performance of our models, we utilize the F1 'macro' score as our primary metric which provides a balanced measure of a model's precision and recall by taking their harmonic mean. This score reaches its best value at 1 (perfect precision and recall) and its worst at 0. The formula used for the F1 score is:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Particularly, we compute the F1 score independently for each class and then take the average to obtain F1 'macro'. This treats all classes equally, giving equal weight to the performance on each class, regardless of its frequency. This is particularly important in datasets with class imbalances, as it ensures that the performance on less frequent classes contributes equally to the overall metric. The 'macro' F1 score is calculated as:

$$\text{F1 'Macro'} = \frac{1}{N} \sum_{i=1}^{N} \text{F1}_i$$

where N is the number of classes, and F1_i is the F1 score for the i-th class.

Chapter 5

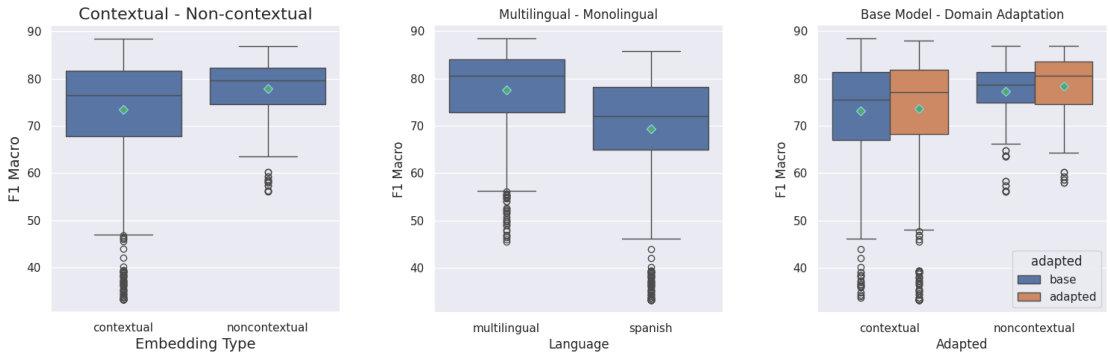
Experimental Results

In the preceding chapters, we have delineated the theoretical underpinnings of our research, outlined our methodology, and discussed the rationale behind our design choices, including the development of models that underwent domain adaptation training. Reflecting the specialized preprocessing methodologies employed, these domain-adapted models are prefixed with "L", symbolizing their tailored training to the analysis of lyrical content. Following this prefix, the base model name is appended to clearly indicate the foundational architecture upon which each domain-adapted model is built. Furthermore, to denote the specific preprocessing technique applied during domain adaptation, the identification 's' (stanzas) or 'c' (chunking) is included in the model name, signifying the methodological nuances that distinguish our approach to leveraging semantic similarity in song lyrics. Conversely, in the case of static models, they are identified by the "SPL" suffix.

With these foundational elements and methodological nuances in place, we now turn our attention to the empirical heart of our study—the experimental results. This chapter presents the outcomes of our investigation, evaluating the effectiveness of various semantic composition methods and supervised training strategies, including cross-encoders and bi-encoders, which have been instrumental in setting new benchmarks in tasks requiring a nuanced understanding of text pairs. Through a comprehensive analysis, we aim to elucidate how these approaches perform in the specific context of identifying semantically similar song lyrics, thereby contributing to the enhancement of music recommendation systems. The findings discussed herein are pivotal, as they not only test our initial hypothesis but also provide insights that could influence future directions in the application of NLP techniques within the realm of music analytics.

5.1 ICDS and Binary Logistic Classifier

In the upcoming analysis, we delve into the practical application of our outlined methodology, employing an unsupervised approach to generate embeddings through the ICDS framework. Upon generating these embeddings and quantifying similarity through the chosen metrics, we leverage the Binary Logistic Classifier to classify pairs of song lyrics.



(a) Contextual and Static models.

(b) Multilingual and Monolingual models.

(c) Base model and Domain Adapted models.

Figure 5.1: Comparative analysis of performance score across various scenarios.

5.1.1 Contextual vs Static Embeddings

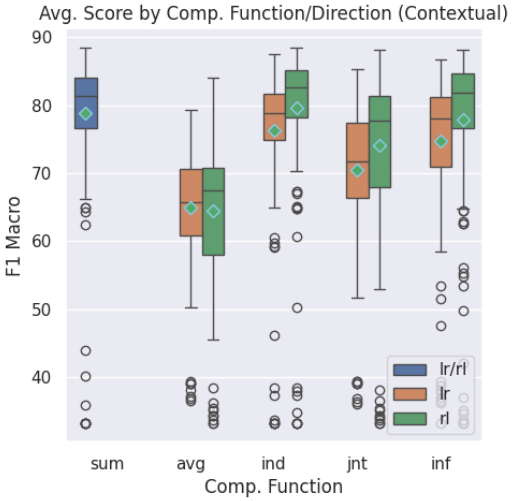
As depicted in Figure 5.1a, we conducted a comparative analysis between contextual and non-contextual embeddings. In this specific setting, non-contextual models such as Word2Vec and GloVe outperformed the transformer-based, contextual models, achieving higher F1 scores, on average, when used in conjunction with a logistic classifier. This suggests that non-contextual embeddings should not be overlooked, especially given their lower computational requirements. In specific scenarios and tasks, they may even offer performance advantages over their contextual counterparts.

Further analysis revealed that the performance disparity between the foundational models and their domain-adapted iterations was marginal (Figure 5.1c), suggesting the absence of statistical significance in the observed differences. This observation intimates that the current dataset may not provide sufficient variance to conclusively assess the impact of domain adaptation on model performance. Therefore, it posits that a more extensive dataset would be requisite for a robust evaluation of the potential advantages conferred by domain-specific model tuning. This insight prompts a reevaluation of the scale and diversity of data necessary for future studies, aiming to delineate the conditions under which domain adaptation might yield statistically significant performance enhancements.

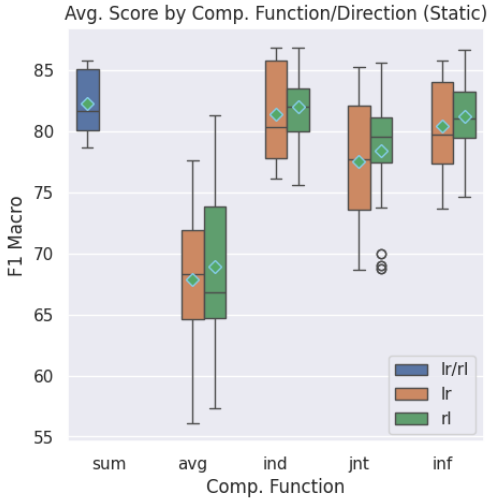
5.1.2 Multilingual and Monolingual models

Our findings indicate that multilingual models have an edge when it comes to assessing song similarity. As illustrated in Figure 5.1b, multilingual models consistently outperformed their monolingual counterparts across various evaluation metrics. This suggests that the capability to understand and encode linguistic nuances from multiple languages into the embeddings is advantageous for the task of song similarity assessment.

5.1.3 Composition Function and Direction



(a) Impact of composition function and direction on performance of contextual models.



(b) Composition function and direction on performance of static models.

Figure 5.2: Comparative performance analysis of the composition function and direction.

The exploration of composition functions and their respective directionalities has revealed distinct patterns in their efficacy for semantic vector composition. Notably, the sum function (F_{sum}), which is direction-agnostic, demonstrated robustness in capturing semantic content, as reflected by the consistently high F1 macro scores across various granularities and models. This function's direction neutrality suggests its strong capacity for holistic semantic aggregation.

For directional composition functions, the impact of the processing sequence on performance is evident in the independent (F_{ind}) and information (F_{inf}) functions, which are sensitive to vector magnitude and alignment. Here, the right-to-left variants consistently outperformed the left-to-right, particularly with methods, as highlighted by the improved F1 macro scores in the right-to-left composition

Further insights are gleaned when examining static models. While the performance results were comparable to those of contextual models, the average composition applied in a left-to-right direction (avg_{lr}) resulted in an F1 macro score of 66.51, while a right-to-left orientation (avg_{rl}) decreased slightly to 62.35. This indicates that even subtle shifts in directional processing can affect the outcome, highlighting the complexity inherent in capturing semantic nuances. Such results underscore the nuanced influence of composition order on the quality of semantic representations, especially when dealing with the layered meanings embedded within song lyrics.

In summary, while the sum function remains unaffected by directionality and excels in its application, the performance of other compositional functions notably varies

with the direction, suggesting that the sequential ordering of information during vector composition is a crucial consideration in modeling semantic relationships within NLP frameworks.

5.1.4 Layer-wise Performance

In contrast with non-contextual models like GloVe and Word2Vec, which are based on a single-layer architecture, contextual models based on the transformer architecture utilizes a multi-layered framework, as explained in Section 4.3.4 (Experimental Setup: Transformer Layer). Figure 5.3 provides an insightful look into how performance varied across different layers of contextual models and how they perform with the available composition functions. Interestingly, we found that the layer 0 of the transformer, which contains positional embedding representations, achieves better results than the more contextual, last layer of the architecture. This suggests that for the specific task of song similarity assessment, the initial, less contextual layers may capture features that are more salient, challenging the notion that deeper layers of the model may capture a more useful representation.

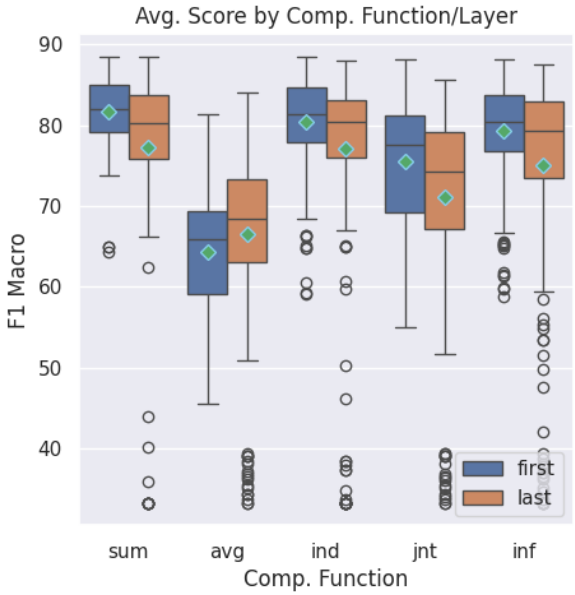
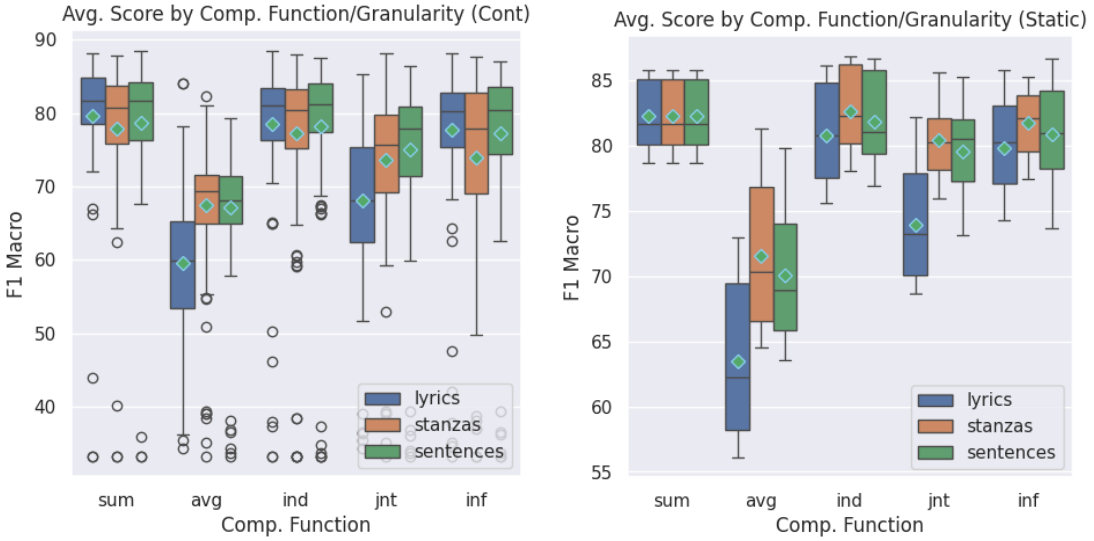


Figure 5.3: Composition function and Layer

5.1.5 Granularity

When the entire corpus of song lyrics is considered, contextual models demonstrate enhanced performance in the F_{sum}, F_{ind}, and F_{inf} composition methods. This is likely attributable to their ability to integrate extensive contextual information, thereby leveraging the rich narrative and thematic layers present in full-length song lyrics. The F_{sum}



(a) Composition function and granularity on avg. F1 macro scores of contextual models.

(b) Composition function and granularity on avg. F1 macro scores of static models.

Figure 5.4: Performance impact of the granularity.

method, aggregating semantic content, the F_{ind} method, assuming independence between linguistic forms, and the F_{inf} method, which considers informational content, all benefit from the broader context that full lyrics provide, as contextual models capture the diverse semantic signals present within the text.

However, when applying the F_{avg} and F_{jnt} composition methods, contextual models do not maintain their lead. Instead, they perform better with stanzas and sentences, with sentences yielding the most consistent performance across these composition methods. The F_{avg} method, averaging semantic vectors, and the F_{jnt} method, representing a joint probability distribution of meanings, appear to be more suitably applied at these lower levels of granularity. This shift suggests that the averaging and joint distribution processes align more closely with the concise and focused semantic information encapsulated within stanzas and sentences.

Static models, on the other hand, lack the ability to leverage larger contextual scopes effectively. However, they exhibit a consistent performance across different compositional methods and granularities, with a marked advantage in stanzas and sentences. This could be due to their invariant representations, which, while not capturing the dynamic contextual shifts within a song, efficiently encode the semantic properties that are central to the shorter, more bounded contexts of stanzas and sentences.

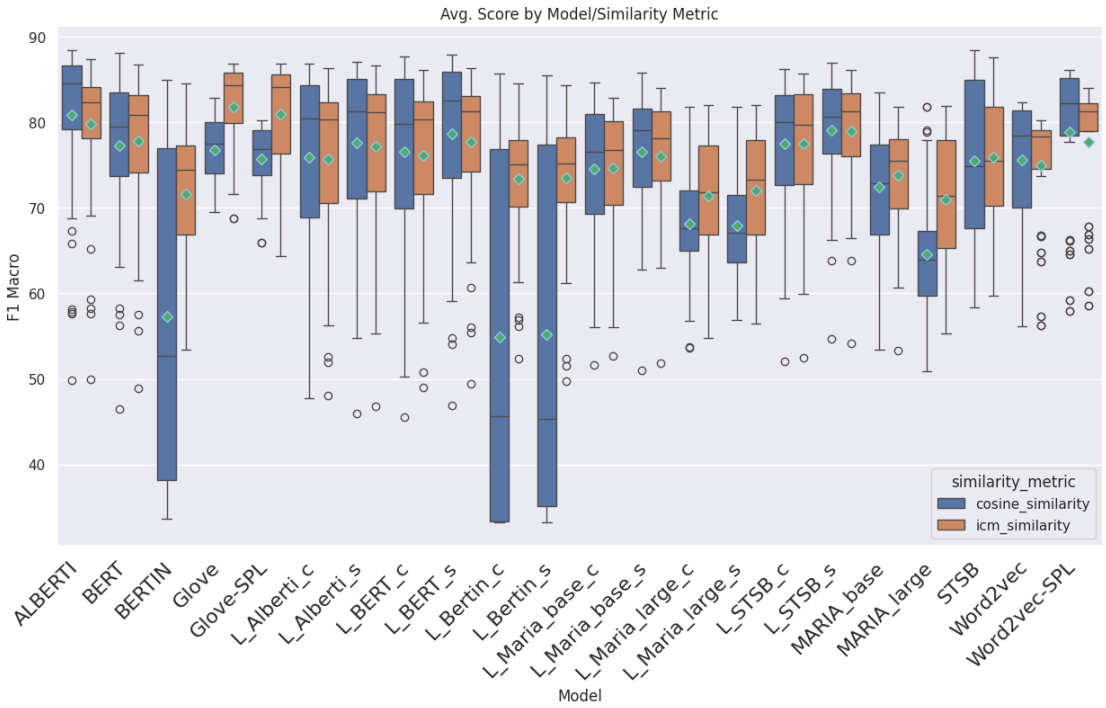


Figure 5.5: Performance comparative by model and similarity metric.

5.1.6 Model

A detailed examination of the F1 macro scores across a variety of models (Figure 5.5, with detail in Table 5.1), considering their respective configurations and the specific similarity metrics employed, sheds light on significant trends in model performance. The ALBERTI model showcases a wide performance range with an average F1 macro score of 80.8 ± 8.5 for cosine similarity and 79.8 ± 8.0 for ICM similarity, illustrating its adaptability across diverse semantic contexts, ranking the highest among contextual models. This variability, with a peak at 88.49 and a minimum at 49.85, highlights the model's flexibility but also its potential sensitivity to task-specific parameters and dataset characteristics.

Conversely, the GloVe model presents a more consistent and slightly superior performance for ICM similarity, with an average F1 macro score of 81.8 ± 5.4, peaking at 86.89. This indicates a higher level of stability across various scenarios compared to the ALBERTI model, positioning it as a reliable option for tasks that demand consistent semantic interpretation.

The L_STSB_s model (adapted from STSB with stanzas preprocessing), with F1 macro scores of 79.1 ± 6.7 for cosine similarity and 79.0 ± 6.5 for ICM similarity, along with the Glove-SPL and Word2vec-SPL models, which display similar average performances around 78.3 to 80.9, reveal a competitive landscape among both static and contextual models. These findings, especially pertinent to the domain of song lyrics

Model	Similarity Metric	F1 Mean Score
Glove	icm_similarity	81.82
Glove-SPL	icm_similarity	80.96
ALBERTI	cosine_similarity	80.83
ALBERTI	icm_similarity	79.78
L_STSB_s	cosine_similarity	79.14

Table 5.1: Model performance on descending order (highest 5 scores).

analysis, demonstrate that while contextual models like ALBERTI can reach high levels of performance, static models such as Glove and Word2vec-SPL offer notable robustness and computational efficiency, making them particularly valuable.

This analysis underscores the criticality of selecting the appropriate model and metric based on the specific demands of the semantic task at hand. While contextual models may afford advantages in capturing deeper semantic nuances within complex datasets, static models argue convincingly for their use in applications where consistency, computational efficiency, and a robust performance profile are paramount.

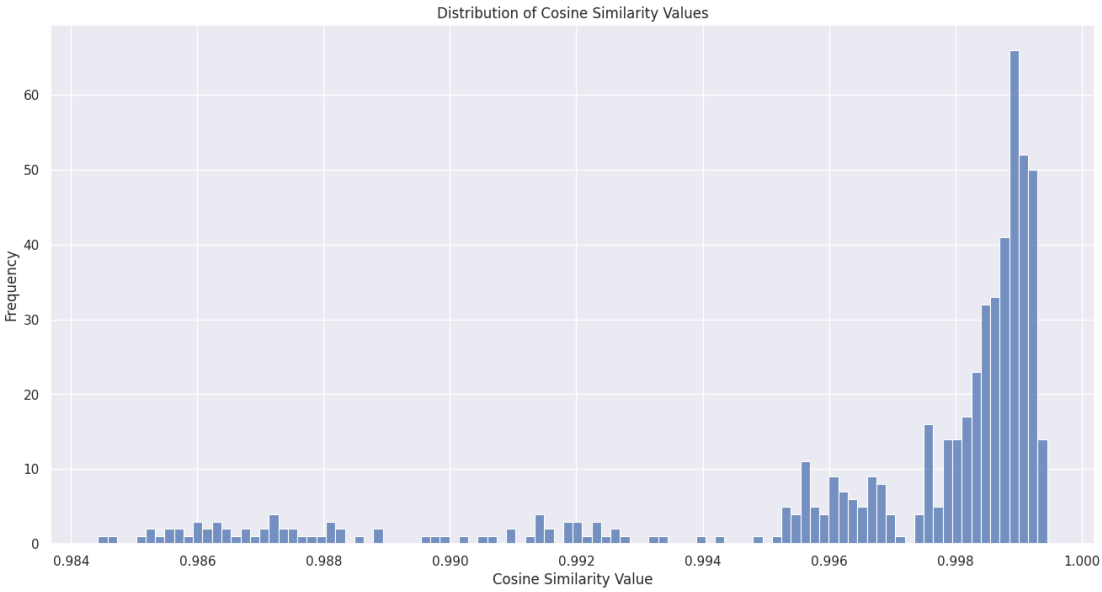


Figure 5.6: Histogram of pair-wise cosine similarity values for BERTIN model. The values cluster in the [0.98, 1] range.

In our research, an intriguing effect was observed concerning the behavior of cosine similarity when applied to certain models, particularly BERTIN and, to a lesser extent, MARIA, both of which are built on the RoBERTa architecture and are monolingual models. It was noted that their performance was substantially lower compared to that of ICM metric. This behavior does not show signs of improvement after additional in-

domain training, as seen in the scores of domain adapted models. Our experiments revealed that the cosine similarity of vectors from these models tends to cluster in a narrow range around 0.98, as highlighted in Figure 5.6. This clustering could have a pronounced impact on the performance of cosine similarity metrics, which rely solely on the angle between vectors. In contrast, ICM not only considers the angle but also incorporates the magnitude of vectors in its calculation of similarity, potentially offering a more discriminative measure that could account for the observed differences in performance. This behavior underscores the importance of considering both direction and magnitude when assessing the semantic similarity of embeddings, especially in models based on RoBERTa architecture.

5.2 Cross-Encoder

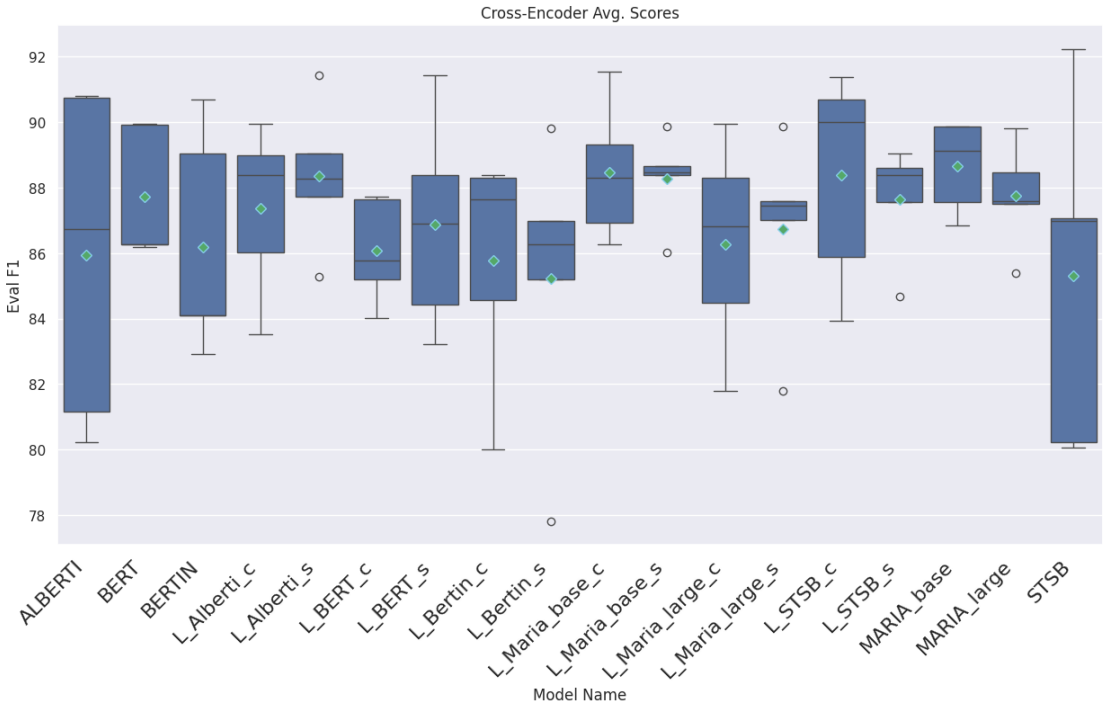


Figure 5.7: Cross-Encoder performance scores.

In our second experimental phase, we scrutinized the efficacy of a cross-encoder model, a sophisticated approach that has gained traction in a variety of NLP tasks for its capacity to capture complex semantic relationships.

As seen in Table 5.2, the MARIA base model emerged as the top-performing model, achieving an average F1 score of 88.65, indicative of its superior capability in accurately classifying song similarity. This model exemplifies the strength of the cross-encoder architecture in capturing intricate semantic relationships within the data.

Conversely, the L_Bertin_s model displayed the least effective performance, with an average F1 score of 85.21, suggesting that not all models benefit equally from the cross-encoder design. This variance underscores the importance of model selection based on the specific characteristics of the task at hand.

This bifurcation in performance, with contextual monolingual models achieving both the highest and lowest scores, adds a fascinating dimension to our analysis. The discrepancy in performance between two models, theoretically suited for the task and that also share architecture, raises intriguing questions about the variability in how different models capitalize on the cross-encoder's capabilities.

Overall, the cross-encoder approach demonstrates the potential for high accuracy in semantic classification tasks. However, the computational demands of such models necessitate careful consideration, especially in scenarios where resources are constrained.

Model Name	**F1 macro**
ALBERTI	85.93
BERT	87.72
BERTIN	86.17
MARIA_base	**88.65**
MARIA_large	87.74
STSB	85.32
L_Alberti_c	87.36
L_Alberti_s	88.36
L_BERT_c	86.07
L_BERT_s	86.87
L_Bertin_c	85.78
L_Bertin_s	85.21
L_Maria_base_c	88.47
L_Maria_base_s	88.27
L_Maria_large_c	86.26
L_Maria_large_s	86.73
L_STSB_c	88.37
L_STSB_s	87.65

Table 5.2: Model Comparison: Evaluated F1 Scores

5.3 Bi-Encoder

Building on the findings from our unsupervised analysis, the second experiment harnessed a supervised bi-encoder framework, which was calibrated using the F_{sum} composition function and lyric-level granularity, as these demonstrated robust performance. This approach capitalized on the supervised setting to refine the representation space and embeddings' semantic composition for song similarity detection.

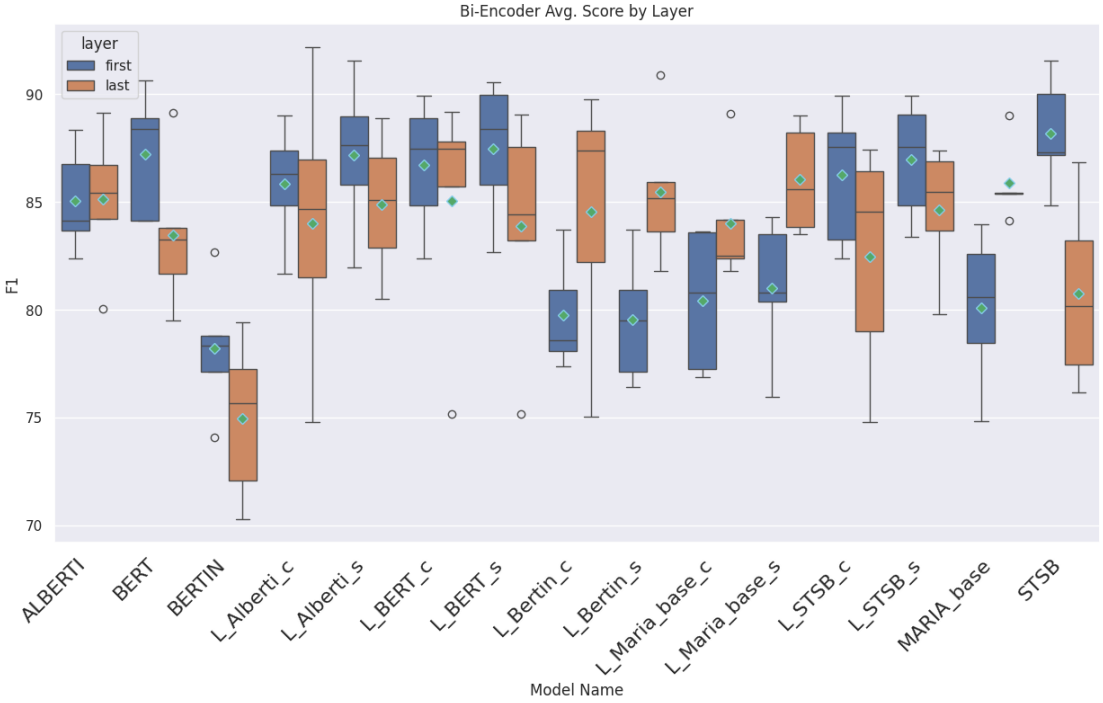


Figure 5.8: Bi-Encoder performance by transformer layer.

In our study, the performance analysis of bi-encoder models across various configurations unveiled noteworthy differences between the first and last layers, as illustrated in Table 5.3. A particularly striking observation is the consistent high performance of the first layer across several models, with the STSB model showcasing the highest first-layer performance at 88.18. Conversely, the last layer performance varies more significantly across models, with the domain-adapted L_Maria_base_s model achieving the highest last-layer performance at 86.04, indicating that domain adaptation, combined with specific data preprocessing techniques (denoted by the suffixes 's' and 'c'), might enhance a model's ability to leverage contextual information for improved semantic analysis.

In comparison to the unsupervised models (Figure 6.1, the supervised bi-encoder demonstrated enhanced capability in discerning song similarity in most models, with the exception of the two based on the BERT architecture (namely, BERT, ALBERTI and their adapter counterparts). This was evident in the improved F1 scores, which signifies a more refined understanding of the semantic nuances in the lyrics. The bi-encoder's performance underscores the value of supervision in embedding generation and semantic composition for complex natural language processing tasks such as this.

The observed performance difference after training within our experiment presents a notable and somewhat unexpected outcome, particularly when examining the efficacy of bi-encoder models in adjusting the representation space of the model's layers. Bi-encoders are designed to fine-tune the representation space of the model last layer.

However, our results indicated a consistent pattern where the first layer of the model, as observed in the unsupervised approach, outperformed the last layer in terms of capturing semantic similarities effectively. This was observed across all models, with the exception of the two monolingual contextual models, as highlighted in Figure 5.8. For the two monolingual contextual models, the exception to this trend could be attributed to their specialized training and architectural nuances, which might enable their last layers to capture and utilize the deep contextual information more effectively for same language data.

model_name	**layer**	
	first	**last**
ALBERTI	85.06	85.12
BERT	87.23	83.48
BERTIN	78.20	74.95
L_Alberti_c	85.85	84.02
L_Alberti_s	87.19	84.89
L_BERT_c	86.70	85.06
L_BERT_s	87.48	83.89
L_Bertin_c	79.74	84.55
L_Bertin_s	79.54	85.48
L_Maria_base_c	80.42	84
L_Maria_base_s	80.99	**86.04**
L_STSB_c	86.27	82.45
L_STSB_s	86.96	84.65
MARIA_base	80.10	85.87
STSB	**88.18**	80.77

Table 5.3: Bi-Encoder Performance by Layer

Furthermore, an interesting development was observed in the alleviation of the clustering issue of cosine similarity values through the bi-encoder training approach. Previously, models such as BERTIN and MARIA exhibited a clustering of cosine similarity scores around 0.98, which posed a challenge for distinguishing semantic differences effectively. The bi-encoder training, by using consine loss as objective function, diversified the distribution of cosine similarity scores. This improvement speaks to the efficacy of bi-encoder training in not just refining embedding quality but also in expanding the models' capacity to capture and differentiate subtle semantic nuances, marking a significant advancement in the application of NLP models to complex tasks such as song similarity detection.

Chapter 6

Discussion

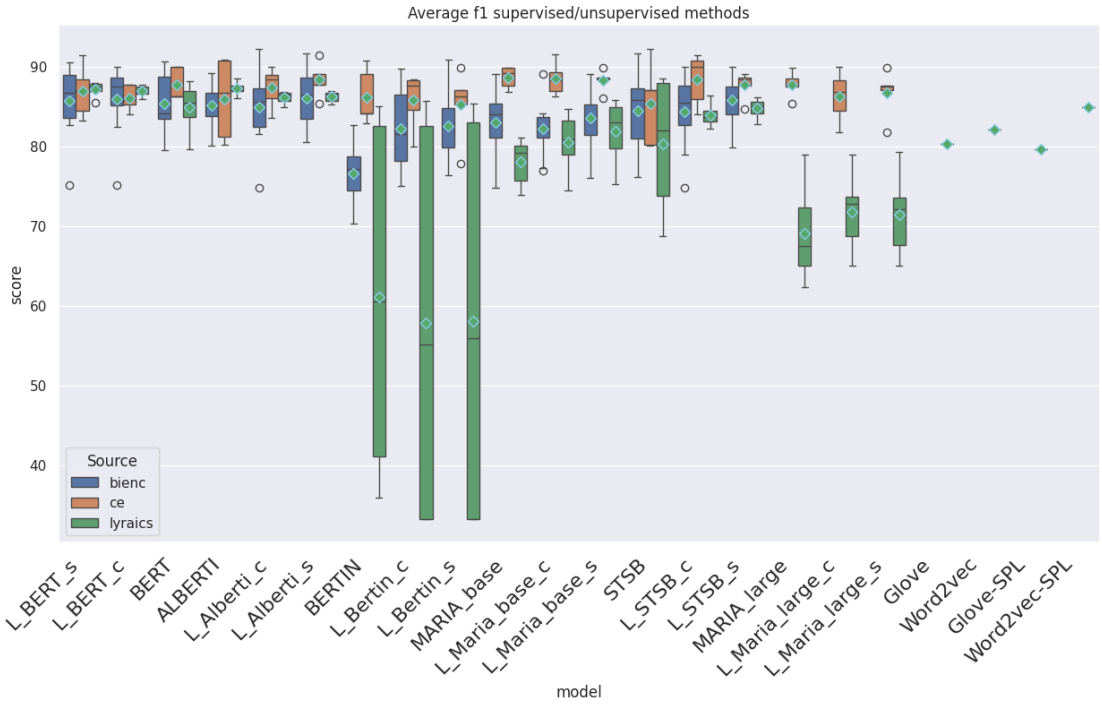


Figure 6.1: Comparative analysis of supervised and unsupervised methods' score across different models, for the "sum" function, cosine similarity metric, lyrics (contextual) and stanzas (static) granularity.

This study has unearthed intriguing performance disparities across various embedding models and compositional strategies in the task of song similarity assessment. Notably, the unexpected superior performance of multilingual models over their monolingual counterparts emerges as a significant finding, challenging the conventional expectation that monolingual models, being domain-specific, would inherently exhibit superior

performance due to their focused training data. This phenomenon suggests that multilingual models may possess a unique advantage in capturing and leveraging the nuanced semantic relationships across languages, thereby providing a more holistic and robust representation of song semantics. Such capability is particularly beneficial in the context of song lyrics, which often embody a rich tapestry of cultural and linguistic nuances that multilingual models are perhaps better equipped to interpret and encode.

During the examination of the comparative performance (Figure 6.1) between cross-encoder and bi-encoder, cross-encoders demonstrated a notable increase in performance metrics, showcasing their potential in capturing complex semantic relationships with a high degree of accuracy. However, this increased performance was accompanied by higher variability, particularly when compared to the unsupervised approach with static models (GloVe, Word2Vec and adapted variants), a phenomenon that can be partially attributed to the use of and additional untrained classifier head with random initialized weights (cross-encoder's classification layer). Such initialization can introduce a level of unpredictability in performance outcomes, as it affects the model's starting position in the gradient descent process. Furthermore, the performance advantage of both, cross-encoders and bi-encoders, suggests that they may benefit substantially from larger datasets. This characteristic underscores a potential limitation in contexts where data availability is constrained, highlighting the importance of dataset size and quality in leveraging the full capabilities of models for complex NLP tasks.

In terms of granularity, contextual models seemed to prefer larger contexts, indicating their design to capitalize on longer sequences for better contextual understanding. Conversely, non-contextual models favored shorter contexts, likely due to their inherent limitations in integrating broader semantic landscapes. This distinction underscores the inherent design differences between these model types, with contextual models requiring more extensive textual inputs to effectively infer semantic relations.

Moreover, the comparison between cosine similarity and ICM metrics revealed an interesting dynamic. While cosine similarity generally performed better by focusing on the angular difference between vectors, it encountered strong difficulties with certain models, such as BERTIN, due to its sole reliance on angular measures. ICM, by considering both angle and magnitude (module) of vectors, provided a more nuanced assessment of similarity, that consistently worked well across all configurations, highlighting the importance of selecting appropriate metrics based on model characteristics and the specific nature of the task. This effect on the cosine similarity was nonetheless alleviated when models were trained in a bi-encoder setting. This training architecture introduced a broader distribution in the cosine similarity values, effectively mitigating the clustering of scores around a narrow range, as previously observed.

In an intriguing departure from expectations, the analysis across all models revealed that the highest performance in terms of composition function direction was consistently achieved with a right-to-left orientation. This outcome is particularly surprising given the natural left-to-right progression of the writing and reading processes in many languages. One possible explanation for this phenomenon lies in the inherent structure and processing dynamics of neural networks, which do not necessarily mimic human cognitive

processes for language understanding, specially for transformer models based on the encoder block, since the self-attention is bidirectional in this architecture. The right-to-left superiority suggests that the models might be capturing and prioritizing semantic information differently, potentially due to the way terminal elements in sequences influence context comprehension. In right-to-left processing, the model encounters the conclusion or outcome of a sentence first, which could allow for a different form of semantic priming, enabling more effective integration of subsequent (or, in this orientation, preceding) elements. This reverse processing could inadvertently align with certain linguistic structures where the resolution or key information appears toward the start, thus offering an unexpected advantage in understanding and predicting the semantic totality of the input. This finding invites a deeper exploration into the cognitive and computational mechanisms underpinning semantic composition, challenging existing assumptions and opening new avenues for model optimization.

Another aspect that merits discussion is the performance of domain-adapted models, which did not exhibit the anticipated improvement over their non-adapted counterparts. This outcome was somewhat surprising, given the prevailing notion that domain adaptation should inherently enhance model performance by aligning the model's knowledge base more closely with the specific characteristics of the target domain. A plausible explanation for this observation could be the limitation posed by the dataset's size. In the realm of deep learning and particularly in tasks involving semantic understanding, the volume and diversity of training data play a crucial role in enabling the model to learn and generalize effectively. The lack of a sufficiently large and varied dataset for domain adaptation could hinder the model's ability to fully exploit the potential benefits of domain-specific tuning, thereby resulting in a performance that does not noticeably surpass that of non-adapted models.

Furthermore, the comparative analysis also sheds light on the nuanced dependencies of model performance on factors such as architecture, training regimen, and compositional functions. For instance, the efficacy of static versus contextual embeddings in unsupervised semantic composition, that could be attributed to a more isometric representation space, underscores the complexity of choosing the optimal approach for a given task. Conversely, the variable performance of contextual embeddings, influenced by model architecture and training, highlights the importance of careful model selection and optimization in leveraging the full potential of these more sophisticated approaches. In the analysis of contextual models, it was observed that the embedding layer consistently delivered the highest performance, a finding that might initially seem counterintuitive. This phenomenon can be attributed to the intrinsic nature of the embedding layer as being the least contextual among the model's layers. Unlike deeper layers, which progressively integrate more context and complex semantic relationships as information propagates through the network, the embedding layer seems to retain a closer alignment with the raw semantic inputs. This characteristic makes the embedding layer particularly effective for tasks where the nuanced understanding of context derived from subsequent layers does not necessarily translate to improved performance.

The findings of this study underscore the multifaceted nature of semantic composi-

tion and model performance in NLP tasks, revealing that the path to optimizing song similarity assessment is not straightforward. The unexpected efficacy of multilingual models points to the value of cross-linguistic semantic understanding, while the nuanced performance of domain-adapted models emphasizes the critical role of dataset size and diversity in achieving effective domain adaptation. In addition, the consistency in performance of static models in this task highlights their value, indicating they should not be overlooked despite the allure of more complex systems. These insights contribute to a deeper understanding of the challenges and opportunities in leveraging NLP techniques for music information retrieval, paving the way for further research into optimizing model selection and training strategies for enhanced performance in song similarity and recommendation systems.

Chapter 7

Conclusions and Future Work

In this work, we have explored the viability of unsupervised and supervised semantic composition methods for song similarity assessment, leveraging both static and contextual word embeddings. Employing a logistic classifier, we demonstrated that unsupervised semantic composition, especially when utilizing word embeddings, presents a competitive approach for enhancing song recommendation systems, which enabled us to answer RQ1. This method benefits from computational advantages over supervised approaches and is particularly effective for datasets too small to train a classifier robustly, thanks to the transfer learning capabilities inherent in pre-trained embeddings.

Our findings in regards of RQ2 suggest that both static and contextual embeddings are effective in unsupervised semantic composition of sentences, including sentences, stanzas, or complete lyrics, for encapsulating sentence semantics through an ICDS approach. Lyrics granularity performed the highest when paired with the sum composition function for contextual models, while it was stanzas in the case of static models. These static models also consistently delivered solid performance across various architectural, and training configurations, offering computational efficiency and a degree of semantic isometry conducive to the classification of song similarity levels (RQ2.1). In contrast, contextual embeddings' effectiveness, is highly dependent on model architecture and training, with the embedding layer (the least contextual of the transformer model) typically yielding the best results despite not preserving semantic isometry as static embeddings do (RQ2.2).

We answered RQ3 about the application of domain-specific transfer learning to both static and contextual embeddings by extending the training of base model with in-domain data. The results revealed mixed outcomes. While some models experienced performance degradation or stagnation, others showed notable improvement. However, the average benefit was marginally better, prompting the need for further data to ascertain the significance of these improvements.

Regarding supervised approaches, our study compared the effectiveness of cross-encoders, logistic classifiers, and bi-encoders in song similarity classification. This approach allowed us to answer RQ4: cross-encoders exhibited the highest potential, achieving superior performance scores but with a dependency on model choice, training data

volume, and the computational challenges posed by combinatorial explosion, where all possible pairs need to go through the transformer machinery. Bi-encoders, nonetheless, emerged as an attractive middle ground, balancing efficiency and performance, suggesting their suitability. While computationally more efficient than the cross-encoder, this method still requires a large amount of annotated data to shine.

As explained, certain research questions could not be conclusively answered within the scope of this study due to limitations in data and computational resources. Future work could further investigate these unresolved areas by employing larger datasets, exploring additional model architectures, and refining unsupervised semantic composition techniques. Additionally, the exploration of novel approaches to reduce the computational demands of cross-encoder models without compromising their accuracy presents an interesting avenue for research.

A pivotal discovery of our study was the pronounced competitiveness of static models in the realm of unsupervised semantic composition of sentences for encapsulating sentence semantics via ICDS. Static models distinguished themselves by delivering consistently robust performance, irrespective of variations in model architecture, layer configurations, or training paradigms. This consistency, coupled with their inherent computational advantages and a degree of semantic isometry, underscores the competitive edge of static embeddings. Their lower dependence on the intricacies of model architecture and the diminished computational demands they impose render them particularly advantageous for scalable and efficient semantic analysis applications.

Another significant finding from our analysis is the robustness of the ICM metric. Unlike traditional cosine similarity, which showed vulnerabilities by clustering values too narrowly for certain models, thereby affecting its discriminative power, ICM demonstrated a remarkable resilience to this issue. The inclusivity of both angle and magnitude in its calculations allowed the metric to maintain consistent performance across a variety of model outputs. Even in scenarios where it did not outperform other metrics, its decline in performance was not as pronounced or detrimental as that observed with cosine similarity. This stability makes ICM a more reliable metric for assessing semantic similarity. This finding not only challenges the conventional reliance on cosine similarity for all model outputs but also paves the way for more sophisticated metrics like ICM, which can better account for the complex semantic landscapes modeled by advanced neural architectures.

Potential future lines of work include investigating the impact of different linguistic features on the performance of embedding models in song similarity tasks and exploring the scalability of these models to other domains within NLP. Moreover, further analysis on the threshold of data size and domain specificity required for effective transfer learning could provide valuable insights for optimizing the performance of both static and contextual embeddings in unsupervised and supervised settings.

Lastly, for a more detailed exploration of our findings, the Appendix A.3 contains a comprehensive table with the full results from our unsupervised experiment. This additional data provides a clearer picture of how different models and configurations performed throughout our study. Readers interested in a deeper dive into the specifics

of our research will find this table particularly useful. It not only supplements the discussions in this thesis but also offers a straightforward look at the data behind our conclusions. We recommend checking out this appendix for anyone looking for a more granular understanding of the effectiveness of semantic composition methods in assessing song similarity.

7.1 Limitations

This study, while comprehensive in its approach to understanding song similarity, encounters limitations that must be acknowledged for a complete appraisal of its findings.

The first limitation pertains to the size of the dataset. Although substantial effort was made to curate a diverse and representative collection of song lyrics, the dataset's size can influence the generalizability of the study's results. Larger datasets may contain more nuanced variations and subtleties in language use, which could affect the performance and robustness of the models. Therefore, the findings presented here should be considered with the caveat that they may not fully extend to larger or more varied corpora.

Computational constraints also represent a significant limitation. The training and fine-tuning of sophisticated models such as the cross-encoder are resource-intensive processes. The models' complexity necessitates considerable computational power, which can be prohibitive, especially when scaling to extensive datasets or implementing the models in real-time applications. These constraints can limit the practicality of employing the most accurate models in commercial or resource-limited settings.

Furthermore, the study's reliance on computational resources such as GPUs and cloud services, while enabling the handling of complex tasks, raises concerns about the accessibility, reproducibility, and replicability of the research. Not all researchers or practitioners in the field have equal access to such computational power, which could hinder the broader adoption and further exploration of the findings.

Lastly, while the study endeavored to optimize model performance, it did not exhaustively explore all hyperparameter configurations or model architectures. The field of NLP is rapidly evolving, with new models and approaches being developed continuously. Therefore, the models and methods employed in this study represent a snapshot of the current state of the art, which may be superseded by more advanced techniques in the future.

In conclusion, while the study provides valuable insights into the applicability of NLP models to song lyrics similarity, these insights must be contextualized within the scope of the dataset size and computational capabilities employed in this research. Future work should aim to address these limitations by incorporating larger datasets, exploring more diverse computational strategies, and continuously adapting to the evolving landscape of NLP methodologies.

Bibliography

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*.

Agerri, R. and Agirre, E. (2023). Lessons learned from the evaluation of Spanish Language Models. *Procesamiento del Lenguaje Natural*, 70(0):157–170.

Agerri, R., San Vicente, I., Campos, J. A., Barrena, A., Saralegi, X., Soroa, A., and Agirre, E. (2020). Give your text representation models some love: the case for Basque. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jindi, J., Naumann, T., and McDermott, M. B. A. (2019). Proceedings of the 2nd clinical natural language processing workshop. In *ACL Anthology*.

Amigó, E., Ariza-Casabona, A., Fresno, V., and Martí, M. A. (2022). Information Theory–based Compositional Distributional Semantics. *Computational Linguistics*, 48(4):907–948.

Amigó, E., Ariza-Casabona, A., Fresno-Fernández, V., and Martí, M. A. (2022). Information theory-based compositional distributional semantics. *Comput. Linguistics*, 48(4):907–948.

Amigó, E., Giner, F., Gonzalo, J., and Verdejo, F. (2020). On the foundations of similarity in information access. *Information Retrieval Journal*, 23(3):216–254.

Andreas, J., Vlachos, A., and Clark, S. (2013). Semantic parsing as machine translation. In *Proceedings of the 51st ACL (Vo. 2: Short Papers)*, pages 47–52, Bulgaria.

Armengol-Estapé, J., Carrino, C. P., Rodriguez-Penagos, C., de Gibert Bonet, O., Armentano-Oller, C., Gonzalez-Agirre, A., Melero, M., and Villegas, M. (2021). Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4933–4946, Online. Association for Computational Linguistics.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *The International Conference on Learning Representations*.

Aucouturier, J.-J., Pachet, F., Roy, P., and Beurivé, A. (2007). Signal + context = better classification. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*.

Baumann, S. and Hummel, O. (2003). Using cultural metadata for artist recommendation. In *Proceedings of the 3rd International Conference on Web Delivering of Music (WEDELMUSIC'03)*.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Benito-Santos, A., Ghajari, A., Hernández, P., Fresno, V., Ros, S., and González-Blanco, E. (2023). LyricSIM: A Novel Dataset and Benchmark for Similarity Detection in spanish song lyrics. *CoRR*, abs/2306.01325.

Biswas, A., Wennekes, E., Wieczorkowska, A., and Laskar, R. H. (2023). *Advances in Speech and Music Technology: Computational Aspects and Applications*. Springer.

Boleda, G. and Erk, K. (2015). Distributional semantic features as semantic primitives — or not. In *AAAI Spring Symposium Series*, pages 2–5.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cai, X., Huang, J., Bian, Y., and Church, K. (2021). Isotropy in the contextual embedding space: Clusters and manifolds. In *Intl. Conference on Learning Representations*.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., and Kurzweil, R. (2018). Universal sentence encoder. *CoRR*, abs/1803.11175.

Coecke, B., Sadrzadeh, M., and Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *CoRR*, abs/1003.4394.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *Machine Learning, Proceedings of the (ICML), 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale.

de la Rosa, J., Ponferrada, E. G., Romero, M., Villegas, P., Salas, P. G. d. P., and Grandury, M. (2022). BERTIN: Efficient Pre-Training of a Spanish Language Model using Perplexity Sampling. *Procesamiento del Lenguaje Natural*, 68(0):13–23.

de la Rosa, J., Álvaro Pérez Pozo, Ros, S., and González-Blanco, E. (2023). Alberti, a multilingual domain specific language model for poetry analysis.

Deldjoo, Y., Schedl, M., Cremonesi, P., and Pasi, G. (2020). Recommender systems leveraging multimedia content. *ACM Comput. Surv.*, 53(5).

Deldjoo, Y., Schedl, M., Hidasi, B., Wei, Y., and He, X. (2022). Multimedia recommender systems: Algorithms and challenges. In *Recommender Systems Handbook*, pages 973–1014. Springer.

Deldjoo, Y., Schedl, M., and Knees, P. (2024). Content-driven music recommendation: Evolution, state of the art, and challenges. 51.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies, Volume 1*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dong, Y., Guo, X., and Gu, Y. (2020). Music recommendation system based on fusion deep learning models. In *Journal of Physics: Conference Series*. IOP Publishing.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 EMNLP-IJCNLP*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Gao, J., He, D., Tan, X., Qin, T., Wang, L., and Liu, T. (2019). Representation degeneration problem in training natural language generation models. In *ICLR*.

Goodwin, E., Sinha, K., and O'Donnell, T. J. (2020). Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the ACL*, pages 1958–1969. ACL.

Gururangan, S., Marasović, A., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. *arXiv preprint arXiv:2004.10964*.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Armentano-Oller, C., Rodriguez-Penagos, C., Gonzalez-Agirre, A., and Villegas, M. (2022). MarIA: Spanish Language Models. *Procesamiento del Lenguaje Natural*, 68(0):39–60.

Han, H., Luo, X., Yang, T., and Shi, Y. (2018). Music recommendation based on feature similarity. In *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*. IEEE.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. (2020). Compositionality decomposed: How do neural networks generalise? *JAIR*, 67:757–795.

Kenter, T., Borisov, A., and de Rijke, M. (2016). Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 941–951.

Kim, N., Chae, W.-Y., and Lee, Y.-J. (2018). Music recommendation with temporal dynamics in multiple types of user feedback. In *Proceedings of the 7th International Conference on Emerging Databases*. Springer, Singapore.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Adv. in Neural Information Processing Systems 28: Annual Conf. on Neural Information Processing Systems*, pages 3294–3302.

Knees, P., Pohle, T., Schedl, M., and Widmer, G. (2006). Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*, Santa Barbara, CA.

Knees, P. and Schedl, M. (2013). A survey of music similarity and recommendation from music context data. *ACM Trans. Multimed. Comput. Commun. Appl.*, 10(1).

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems 27*, pages 2177–2185.

Levy, O., Søgaard, A., and Goldberg, Y. (2017). A strong baseline for learning cross-lingual word embeddings from sentence alignments. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (2020). On the sentence embeddings from pre-trained language models. In *2020 Conference on EMNLP*, pages 9119–9130, Online. Association for Computational Linguistics.

Li, G. and Zhang, J. (2018). Music personalized recommendation system based on improved knn algorithm. In *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, E., Seddah, D., and Sagot, B. (2020). Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Maruyama, Y. (2019). Compositionality and contextuality: The symbolic and statistical theories of meaning. In Bella, G. and Bouquet, P., editors, *Modeling and Using Context - 11th International and Interdisciplinary Conference, CONTEXT 2019, Italy, 2019*, pages 161–174. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mitchell, J. and Lapata, M. (2010). Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.

Moscato, V., Picariello, A., and Sperli, G. (2020). An emotional recommender system for music. *IEEE Intelligent Systems*.

Nanopoulos, A., Rafailidis, D., Symeonidis, P., and Manolopoulos, Y. (2010). Musicbox: Personalized music recommendation based on cubic analysis of social tags. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(2):407–412.

Nowak, S. and Rüger, S. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566. ACM.

O'Dair, M. and Fry, A. (2020). Beyond the black box in music streaming: the impact of recommendation systems upon artists. *Popular Communication*.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543.

Pimentel, T., Valvoda, J., Hall Maudslay, R., Zmigrod, R., Williams, A., and Cotterell, R. (2020). Information-theoretic probing for linguistic structure. In *58th Annual Meeting of the ACL*, pages 4609–4622, Online. ACL.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Reimers, N. and Gurevych, I. (2019a). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.

Reimers, N. and Gurevych, I. (2019b). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 EMNLP-IJCNLP*, pages 3982–3992.

Ruder, S. (2019). *Neural Transfer Learning for Natural Language Processing*. Thesis, NUI Galway.

Salton, G. and Lesk, M. E. (1965). The SMART automatic document retrieval systems - an illustration. *Commun. ACM*, 8(6):391–398.

Schedl, M., Gómez, E., and Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Found. Trends Inf. Retr.*, 8(2–3):127–261.

Schedl, M., Knees, P., and Gouyon, F. (2017). New paths in music recommender systems research. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*.

Sordo, M., Laurier, C., and Celma, O. (2007). Annotating music collections: How content-based similarity helps to propagate labels. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, pages 531–534.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to Fine-Tune BERT for Text Classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Swearingen, K. and Sinha, R. (2001). Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 workshop on recommender systems*. Citeseer.

Talmor, A., Elazar, Y., Goldberg, Y., and Berant, J. (2020). olmpics-on what language model pre-training captures. *Transactions of the ACL*, 8:743–758.

Thorat, P. B., Goudar, R., and Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017a). Attention is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017b). Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Velankar, M., Deshpande, A., and Kulkarni, P. (2020). *3 application of machine learning in music analytics*. De Gruyter.

Wilks, Y. (1968). On-line semantic analysis of english texts. *[Mechanical Translation and Computational Linguistics*, 11(2):59–72.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Wu, J., Belinkov, Y., Sajjad, H., Durrani, N., Dalvi, F., and Glass, J. (2020). Similarity analysis of contextual word representation models. pages 4638–4655.

Yogatama, D., de Masson d'Autume, C., Connor, J., Kociský, T., Chrzanowski, M., Kong, L., Lazaridou, A., Ling, W., Yu, L., Dyer, C., and Blunsom, P. (2019). Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373.

Glossary

AI Artificial Intelligence. 1, 26

CBOW Continuous Bag of Words. 19, 21

GloVe Global Vectors. 8, 20–22, 31, 42, 44, 46, 54

IC Information Content. 11, 28

ICDS Information-Theoretic Compositional Distributional Semantics. 10–13, 27, 28, 30, 41, 57, 58

ICM Vector-Based Information Contrast Model. 12, 28, 31, 32, 46–48, 54, 58

LLMs Large Language Models. 3, 15

MLM Masked Language Modeling. 8, 24–26, 36, 71, 72

NLP Natural Language Processing. 1–4, 8, 10, 15, 17, 19–21, 23, 24, 26, 30, 31, 41, 44, 48, 54, 56, 58, 59

NSP next sentence prediction. 24, 25

PMI Point-Wise Mutual Information. 31

RIAA Recording Industry Association of America. 1

RSs Recommender Systems. 1–3, 5, 6, 17

Seq2Seq Sequence to Sequence. 8

SGNS Skip-Gram with Negative Sampling. 8

SOTA state-of-the-art. 4, 9, 16, 19

VSM Vector Space Model. 8

Appendix A

APPENDIX

A.1 Training Costs

In this study, a total of 18 representations were trained. The cost for each specific representation is divided based on the expenses of the parameter optimization and representation training phases. The expense associated with the fine tuning process can fluctuate based on the instances utilized in SageMaker and EC2 (Amazon Elastic Compute Cloud) from AWS (Amazon Web Service), and is contingent on the particular tasks involved in the training process.

For domain adaptation, 6 models were trained using the two techniques described. The parameter optimization task involves finding the optimal values for the hyperparameters of the model, which can significantly impact its performance. This task was performed using EC2 instances and cost approximately $1,797. The instance used was p4d.24xlarge, which provides 8 A100 GPUs with 40GB of memory per GPU, 320 cores of 3rd generation NVIDIA Tensor Core with up to 250 TOPS, 192 vCPU, 768 GB of RAM memory, and 3.8 TB of local NVMe-based SSD storage. The model training task involves training the machine learning model using the optimized hyperparameters. This task was also performed using the p4d.24xlarge instance. It is important to note that the cost may vary depending on the size of the model and the length of the sequence. In total, the cost for the pre-training MLM task was approximately $4,700 using the p4d.24xlarge instance.

The fine tuning task for the cross-encoder was executed using EC2 instances for approximately 20 hours at a cost of $250. The instance employed was a p3.8xlarge, featuring 4 V100 GPUs with 16GB of memory per GPU, 32 vCPU, and 244 GB of RAM memory.

The training task was also conducted using the same instance for roughly 8 hours at a cost of $100. It should be noted that costs may vary depending on the representation's size and the sequence length.

In the case of the bi-encoder, a consumer grade machine was utilized, consisting of a single NVIDIA RTX 4090 with 24GB of VRAM and 64GB of RAM. The training took about 48 hours, with an estimated associated cost of $2. It is crucial to emphasize

that these costs are approximate and may differ depending on the specific resources employed in the training process. Nonetheless, by leveraging AWS and Hugging Face, cost-effective and scalable training of machine learning representations was achieved, allowing for comprehensive analysis and experimentation.

A.2 Domain Adaptation Training

Whole Word Mask (Devlin et al., 2019), where the whole word is masked if any of its constituents tokens is selected for masking, was used for the MLM task, aiming to improve the performance of the language models on the downstream task. To this end, the implementation of HuggingFace was extended to accommodate other models with different special tokens (e.g. Roberta and XLMRoberta tokenizers). This allowed us to better capture the relationship between words within a sentence, and the model to better asses the probability distribution of words, as entire words were masked at once instead of just individual tokens or subwords.

BERT models have been found to improve during fine-tuning, achieving best performance after 100k training steps (Sun et al., 2019). In this work, all models were trained for 150k steps, which is comparable to he amount employed to train a model with an unannotated dataset of 2M elements in biomedical domain adaptation (Alsentzer et al., 2019). In the conducted experiments, up to 100 epochs (Gururangan et al., 2020) and 300k steps were explored; however, no additional enhancements were observed. Our pre-training approach prevented the model from losing its generalization capacity despite the risk of overfitting and catastrophic forgetting.

To train with 512 tokens, the warm-up value was reduced to 500 steps (de la Rosa et al., 2022), with Adam optimizer and the same values of epsilon and beta as the original BERT paper (Devlin et al., 2019). During training, the learning rate decayed linearly. The training subset was composed of 80% of the entire original dataset, while the validation subset was composed of the remaining 20%.

A.2.1 Hyper-parameters

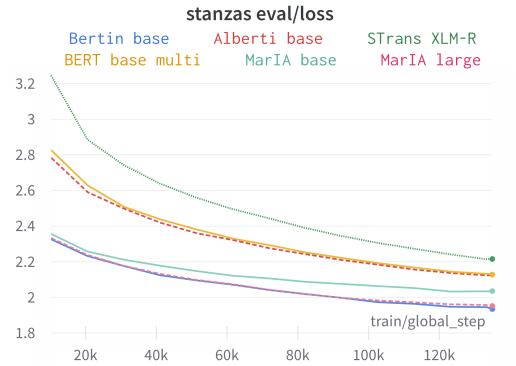
For hyper-parameter selection a grid search was perform for 20k steps, to asses the best set over the search space illustrated in Table A.1.

Parameters	**Values**
Learning Scheduler	Linear
Total steps	20k
Batch size range	32, 64, 1024
Learning rate range	1e-5, 2e-5, 3e-5, 5e-5, 1e-4
Weight decay range	0.01, 0.1
Granularity range	stanzas/lyrics, chunks of 128/512 tokens

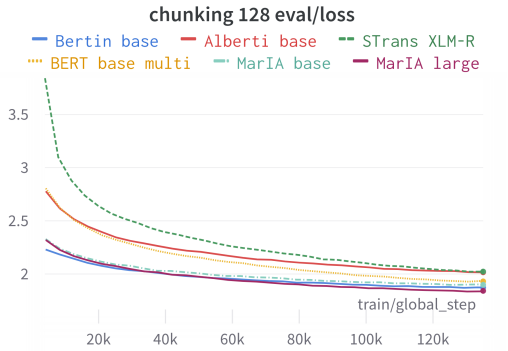
Table A.1: Hyperparameter ranges for model adaptation

A.2.2 Training Evaluation Loss

When compared to the chunking approach, the utilization of stanzas or lyrics leads to a reduced loss value during training across all models. This indicates that maintaining document boundaries may offer certain benefits in terms of the model's capacity to learn the structure and semantic content of the text. In addition, as depicted in Figure A.1a and Figure A.1b, a faster convergence rate was identified for monolingual models as opposed to multilingual ones, a trend that aligns with the former being pre-trained more prominently on a dataset with a greater degree of vocabulary overlap in relation to Spanish songs. Likewise, monolingual models exhibit lower loss values. It is worth noting that Alberti failed to outperform BERT, particularly when employing the chunking technique, despite both models sharing a common architecture and Alberti being trained on poetry.



(a) Evaluation loss for the first stage of training with stanzas/lyrics



(b) Evaluation loss for the first stage of training with chunking 128/512

Figure A.1: Evaluation loss for stanzas and chunking preprocessing techniques.

Model	Loss	F1	Precision	Recall	Accuracy
Alberti base chunking	1.576	0.684	0.694	0.689	0.689
BERT base chunking	1.527	0.691	0.701	0.696	0.696
Bertin base chunking	**1.428**	**0.718**	**0.725**	**0.728**	**0.728**
MarIA base chunking	1.460	0.715	0.722	0.726	0.726
MarIA large chunking	1.443	0.718	0.725	0.728	0.728
STSB chunking	1.602	0.681	0.692	0.687	0.687
Alberti base stanzas	1.530	0.700	0.711	0.705	0.705
BERT base stanzas	1.498	0.703	0.714	0.708	0.708
Bertin base stanzas	**1.357**	**0.736**	**0.743**	**0.745**	**0.745**
MarIA base stanzas	1.39	0.732	0.739	0.742	0.742
MarIA large stanzas	1.368	0.736	0.743	0.745	0.745
STSB stanzas	1.576	0.691	0.704	0.698	0.698

A.3 Unsupervised Composition Full Results

model	layer	dir	op sim	avg lyr	avg sen	avg sta	ind lyr	ind sen	ind sta	inf lyr	inf sen	inf sta	jnt lyr	jnt sen	jnt sta	sum lyr	sum sen	sum sta
ALBERTI	first	lr	ICM	58.29	59.31	57.64	78.04	78.79	79.35	77.64	79.91	81.15	73.23	78.3	77.9	81.18	82.87	82.28
			cos	57.64	57.86	58.19	80.22	81.08	80.81	79.3	80.04	79.17	68.79	78.1	74.5	87.79	85.98	87.08
		rl	ICM	49.9	65.19	69.13	84.15	82.14	83.77	84.14	85.83	86.38	81.32	84.19	86.01	81.18	82.87	82.28
			cos	49.85	65.87	67.3	88.49	85.97	87.26	88.12	86.5	87.27	81.61	84.85	86.73	87.79	85.98	87.08
	last	lr	ICM	78.03	76.95	75.39	85.66	82.35	82.44	86.73	85.28	86.74	83.3	83.14	82.23	81.97	83.99	82.85
			cos	78.25	75.83	75.17	85.77	85.67	87.43	86.33	84.76	86.71	84.38	84.06	84.21	87.04	88.45	86.85
		rl	ICM	84.04	75.38	80.98	86.17	84.7	82.97	87.43	86.72	85.11	85.1	81.32	83.13	81.97	83.99	82.85
			cos	84.03	74.58	80.94	87.02	86.07	86.68	86.85	84.96	85.19	85.24	82.38	83.77	87.04	88.45	86.85
BERT	first	lr	ICM	57.54	61.49	55.64	79.58	80.92	82.31	78.28	82.96	81.33	74.17	81.31	79.55	83.23	83.81	83.79
			cos	57.54	58.29	56.29	83.26	84.71	82.41	83.08	83.48	81.51	73.28	81.55	77.75	88.11	86.97	86.84
		rl	ICM	48.94	68.56	71.46	85.81	84.01	85.25	85.99	85.99	85.64	82.37	84.88	86.73	83.23	83.81	83.79
			cos	46.46	68.94	70.04	86.43	86.98	87.58	86.05	86.45	86.67	82.57	84.58	85.57	88.11	86.97	86.84
	last	lr	ICM	71.48	73.83	74.19	78.76	83.69	83.99	80.43	83.07	81.33	74.15	79.84	80.08	80.97	82.94	81.52
			cos	70.95	74.54	75.97	80.22	77.92	82.99	79.51	78.01	80.45	73.73	76.64	78.3	83.51	79.56	84.04
		rl	ICM	62.53	67.97	68.61	80.89	84.93	81.47	79.37	82.67	77.72	65.04	79.03	78.91	80.97	82.94	81.52
			cos	63.05	68.2	68.48	79.56	78.57	79.84	75.78	75.43	76.71	65.91	70.34	73.84	83.51	79.56	84.04
BERTIN	first	lr	ICM	64.18	70.26	67.23	78.24	81.67	77.78	77.3	78.06	70.75	69.15	74.05	69.14	81.7	82.41	80.81
			cos	64.76	71.02	67.76	81.44	78.6	71.0	81.09	77.97	71.06	68.96	76.73	68.71	84.98	84.35	77.13
		rl	ICM	54.06	68.59	72.73	84.22	84.57	83.15	82.6	82.72	77.36	62.48	79.74	76.3	81.7	82.41	80.81
			cos	53.11	68.86	71.25	81.24	84.94	81.26	80.15	82.25	79.46	60.48	77.54	75.17	84.98	84.35	77.13
	last	lr	ICM	67.95	70.38	63.31	75.62	75.07	75.11	77.0	66.87	53.42	68.03	70.56	63.48	75.94	75.21	75.42
			cos	52.24	36.46	38.99	46.22	37.32	38.42	47.55	36.22	37.07	51.71	36.13	38.99	43.89	35.91	40.11
		rl	ICM	54.89	60.15	62.87	76.17	75.24	74.93	77.34	67.0	53.4	55.59	59.95	64.34	75.94	75.21	75.42
			cos	36.24	34.34	38.37	50.26	34.8	38.44	42.0	33.64	37.0	36.57	33.95	38.16	43.89	35.91	40.11
Glove	first	lr	ICM	68.73	73.13	76.73	85.64	86.17	86.71	82.94	84.01	82.94	72.72	81.07	80.19	85.83	85.83	85.83
			cos	69.47	71.48	75.59	76.18	76.97	78.24	75.06	73.7	77.47	70.11	73.14	75.97	80.26	80.26	80.26
		rl	ICM	71.66	79.88	81.3	84.71	86.53	86.89	84.34	86.71	84.56	73.74	84.72	85.64	85.83	85.83	85.83
			cos	72.98	77.48	80.03	78.93	80.08	81.23	77.18	79.59	82.83	69.96	77.54	81.63	80.26	80.26	80.26
	last	lr	ICM	68.73	73.13	76.73	85.64	86.17	86.71	82.94	84.01	82.94	72.72	81.07	80.19	85.83	85.83	85.83
			cos	69.47	71.48	75.59	76.18	76.97	78.24	75.06	73.7	77.47	70.11	73.14	75.97	80.26	80.26	80.26
		rl	ICM	71.66	79.88	81.3	84.71	86.53	86.89	84.34	86.71	84.56	73.74	84.72	85.64	85.83	85.83	85.83
			cos	72.98	77.48	80.03	78.93	80.08	81.23	77.18	79.59	82.83	69.96	77.54	81.63	80.26	80.26	80.26
Glove-SPL	first	lr	ICM	69.53	74.0	77.29	85.44	86.7	86.9	84.19	85.07	84.73	72.21	82.19	82.74	85.64	85.64	85.64
			cos	69.8	70.08	77.64	77.65	77.33	79.87	76.8	75.08	79.3	70.52	73.75	76.99	79.67	79.67	79.67
		rl	ICM	64.34	76.04	73.79	85.25	86.35	86.53	83.61	85.46	85.28	69.02	82.01	80.38	85.64	85.64	85.64
			cos	65.9	74.24	73.09	75.66	79.07	80.27	74.64	76.93	79.1	68.75	74.26	77.61	79.67	79.67	79.67
	last	lr	ICM	69.53	74.0	77.29	85.44	86.7	86.9	84.19	85.07	84.73	72.21	82.19	82.74	85.64	85.64	85.64
			cos	69.8	70.08	77.64	77.65	77.33	79.87	76.8	75.08	79.3	70.52	73.75	76.99	79.67	79.67	79.67
		rl	ICM	64.34	76.04	73.79	85.25	86.35	86.53	83.61	85.46	85.28	69.02	82.01	80.38	85.64	85.64	85.64
			cos	65.9	74.24	73.09	75.66	79.07	80.27	74.64	76.93	79.1	68.75	74.26	77.61	79.67	79.67	79.67
L_Alberti_c	first	lr	ICM	52.0	64.06	56.31	71.76	80.37	80.69	70.96	80.41	79.34	61.91	76.63	71.8	82.9	81.99	82.11
			cos	51.68	61.13	57.31	74.7	81.73	79.89	70.77	80.17	78.1	54.98	77.76	68.44	86.86	85.49	84.87
		rl	ICM	48.07	66.59	70.38	85.1	83.82	84.72	85.28	86.36	83.12	79.18	83.67	84.23	82.9	81.99	82.11
			cos	47.71	66.25	70.21	85.72	86.07	85.58	85.53	85.71	84.44	77.64	82.53	81.37	86.86	85.49	84.87
	last	lr	ICM	58.08	72.78	67.33	79.61	80.51	81.3	79.55	83.67	80.62	65.14	82.25	77.03	82.34	81.61	82.15
			cos	57.4	71.13	66.73	82.17	84.49	83.15	81.12	83.21	79.56	63.38	80.76	72.4	86.69	86.33	86.52
		rl	ICM	52.57	65.11	68.37	81.54	81.73	83.73	81.68	84.19	82.96	62.59	76.51	79.16	82.34	81.61	82.15
			cos	51.55	63.34	67.48	83.05	84.15	86.32	81.44	83.06	83.29	61.58	77.9	77.93	86.69	86.33	86.52
L_Alberti_s	first	lr	ICM	55.3	64.23	55.29	76.29	81.09	80.14	75.37	81.32	79.16	67.44	78.82	71.61	82.33	83.08	81.93
			cos	55.51	62.64	54.74	78.65	83.04	80.25	76.68	81.47	78.64	60.6	78.82	68.06	86.55	86.8	85.25
		rl	ICM	46.81	65.81	71.63	84.91	83.62	85.08	84.36	86.72	84.57	81.33	84.39	86.01	82.33	83.08	81.93
			cos	45.98	65.15	69.84	86.46	86.45	87.07	86.46	85.9	86.52	81.18	82.71	84.47	86.55	86.8	85.25
	last	lr	ICM	60.69	73.09	71.44	79.43	81.96	83.58	80.61	82.6	83.86	66.76	81.89	80.62	82.11	81.75	82.83
			cos	59.3	72.44	66.78	83.45	84.68	86.53	79.15	84.48	84.73	66.45	80.91	77.73	86.53	85.4	86.88
		rl	ICM	63.5	69.74	72.87	83.43	82.65	85.43	84.02	82.96	84.22	68.4	77.95	79.71	82.11	81.75	82.83
			cos	63.5	70.67	72.91	83.97	84.48	85.8	82.53	83.03	83.97	68.61	77.86	80.07	86.53	85.4	86.88
L_BERT_c	first	lr	ICM	50.79	64.45	56.53	72.52	81.45	80.68	72.63	81.47	79.53	61.17	79.51	75.77	84.36	83.08	82.32
			cos	50.28	62.93	56.24	77.92	83.78	79.18	75.43	81.28	78.27	56.9	79.32	72.13	87.03	86.6	85.94
		rl	ICM	48.99	68.21	71.39	85.45	84.01	85.45	85.81	86.17	84.74	81.13	85.63	85.47	84.36	83.08	82.32
			cos	45.55	67.7	70.51	85.89	86.98	86.86	85.54	85.35	86.86	80.79	84.21	84.49	87.03	86.6	85.94
	last	lr	ICM	61.32	72.56	69.42	77.54	81.84	81.33	78.82	81.86	78.28	63.93	80.05	74.18	81.45	81.48	81.82
			cos	60.74	71.15	67.14	80.76	83.48	80.43	78.24	81.5	78.82	61.98	77.11	69.68	87.57	87.74	86.64
		rl	ICM	57.63	69.07	69.1	81.7	81.57	83.02	82.57	83.86	82.78	63.19	76.34	77.87	81.45	81.48	81.82
			cos	58.0	68.34	67.48	82.47	86.49	86.86	80.67	83.64	83.79	63.47	77.0	77.74	87.57	87.74	86.64
L_BERT_s	first	lr	ICM	55.38	63.64	56.03	77.84	80.7	81.03	76.37	82.39	79.69	68.05	79.69	76.67	83.25	84.17	82.3
			cos	54.03	59.91	54.81	82.05	83.68	80.62	80.79	81.7	78.8	66.79	80.5	73.95	87.24	86.8	85.41
		rl	ICM	49.41	66.55	71.08	84.71	84.53	86.16	84.9	86.34	84.2	82.05	85.07	86.37	83.25	84.17	82.3
			cos	46.9	67.43	70.71	86.63	86.43	86.52	86.26	85.69	86.69	83.18	84.22	86.34	87.24	86.8	85.41
	last	lr	ICM	60.73	73.3	71.02	79.96	81.48	82.42	80.18	83.66	82.6	69.11	82.4	76.68	82.51	82.52	82.89

			cos	59.08	71.58	69.87	83.93	84.97	86.35	81.97	83.72	84.19	69.23	81.78	75.18	87.94	87.77	87.75
		rl	ICM	66.1	73.61	77.76	82.61	81.56	83.17	82.93	84.58	84.76	70.16	80.46	80.61	82.51	82.52	82.89
			cos	65.74	73.42	77.05	85.17	85.95	87.96	83.04	84.36	85.79	70.32	79.31	80.08	87.94	87.77	87.75
L_Bertin_c	first	lr	ICM	63.82	70.45	67.06	78.41	81.32	77.74	77.29	78.05	70.16	68.43	74.23	68.97	81.7	82.77	81.16
			cos	64.6	71.39	67.05	81.44	78.6	71.19	81.27	77.82	70.34	68.79	76.74	67.98	85.72	84.35	77.0
		rl	ICM	52.35	67.3	70.77	84.4	84.57	82.79	82.78	82.55	77.74	62.15	79.56	75.59	81.7	82.77	81.16
			cos	51.95	67.74	70.54	81.64	85.13	81.27	80.34	82.97	79.82	59.7	77.02	74.47	85.72	84.35	77.0
	last	lr	ICM	70.23	79.24	74.46	75.42	74.68	75.1	74.64	76.56	61.34	70.41	79.94	73.93	75.76	75.04	75.77
			cos	37.17	38.05	39.1	33.25	33.64	33.25	33.25	39.42	38.78	36.44	39.3	39.1	33.25	33.25	33.25
		rl	ICM	57.19	67.7	73.82	75.98	74.66	75.1	74.29	75.71	56.11	56.85	67.67	75.08	75.76	75.04	75.77
			cos	34.39	33.25	33.25	37.96	33.25	33.25	34.67	33.25	33.25	34.39	33.25	33.25	33.25	33.25	33.25
L_Bertin_s	first	lr	ICM	64.71	70.45	67.04	79.14	81.86	77.96	77.11	78.23	71.3	68.61	74.58	69.5	81.7	82.95	81.34
			cos	64.43	71.01	67.92	81.61	78.79	71.57	81.27	78.37	71.26	68.97	77.51	68.89	85.35	84.52	78.6
		rl	ICM	52.38	68.22	71.3	84.4	84.39	83.5	82.97	82.36	78.27	62.46	79.74	76.32	81.7	82.95	81.34
			cos	51.22	67.6	70.0	81.64	85.5	81.48	80.35	82.99	80.19	60.43	77.22	75.0	85.35	84.52	78.6
	last	lr	ICM	65.26	78.37	77.5	75.42	74.68	75.1	73.54	74.24	51.49	65.26	79.65	77.49	75.76	75.4	75.77
			cos	39.1	36.91	39.32	33.25	33.25	33.25	37.73	36.56	38.11	39.1	36.91	39.32	33.25	33.25	33.25
		rl	ICM	61.16	73.9	75.26	75.98	75.05	75.1	73.9	71.82	49.72	62.05	74.65	76.33	75.76	75.4	75.77
			cos	35.44	33.64	35.16	37.26	33.25	33.25	35.06	33.25	33.25	35.44	33.64	35.16	33.25	33.25	33.25
L_Maria_base_c	first	lr	ICM	63.3	67.53	64.99	72.86	77.02	76.63	72.69	74.5	70.38	64.29	70.35	66.92	78.02	78.91	79.07
			cos	63.13	66.97	64.84	75.77	74.94	68.33	74.88	73.39	67.23	66.12	71.72	64.68	79.83	78.7	74.46
		rl	ICM	52.72	66.15	70.17	81.64	81.09	77.87	81.33	80.55	75.94	80.22	79.83	80.45	78.02	78.91	79.07
			cos	51.62	64.97	69.18	82.8	81.6	76.81	82.44	81.25	76.42	78.14	80.87	79.75	79.83	78.7	74.46
	last	lr	ICM	67.11	75.85	76.39	80.59	80.4	82.02	79.1	80.0	74.6	72.38	80.6	79.47	78.83	78.12	77.93
			cos	67.27	75.53	73.76	82.52	79.79	83.35	81.44	79.66	82.24	71.71	77.5	77.11	84.71	81.96	83.7
		rl	ICM	56.32	70.13	69.6	82.93	81.47	81.68	81.09	79.13	74.03	56.05	74.65	76.14	78.83	78.12	77.93
			cos	56.07	65.5	69.59	81.07	80.21	82.48	79.11	76.73	80.97	56.8	70.27	73.84	84.71	81.96	83.7
L_Maria_base_s	first	lr	ICM	62.95	67.19	64.99	74.56	77.2	76.13	73.65	74.68	69.24	66.19	70.53	67.84	80.57	79.07	79.25
			cos	62.77	66.63	64.28	77.37	75.92	69.6	77.2	74.14	67.96	67.99	72.24	65.82	81.08	79.31	75.2
		rl	ICM	51.87	65.98	69.79	82.9	81.45	77.37	82.74	81.11	77.75	80.78	80.2	81.52	80.57	79.07	79.25
			cos	51.02	64.41	68.81	85.01	80.91	78.07	83.53	81.66	78.06	78.93	81.06	80.3	81.08	79.31	75.2
	last	lr	ICM	63.6	79.3	72.08	80.76	81.31	81.28	78.53	81.66	73.09	67.18	82.75	74.82	78.83	78.12	78.12
			cos	63.37	78.47	70.09	81.32	82.92	82.45	80.0	82.78	80.38	67.12	81.35	75.51	84.85	84.86	85.8
		rl	ICM	75.18	78.05	82.24	84.03	82.51	82.54	81.48	82.41	74.96	73.61	79.7	82.78	78.83	78.12	78.12
			cos	73.28	76.83	80.46	82.89	84.68	82.62	81.09	84.01	82.81	74.11	79.21	80.97	84.85	84.86	85.8
L_Maria_large_c	first	lr	ICM	60.89	63.53	62.65	71.48	70.1	69.34	69.92	66.71	61.28	56.36	64.77	61.51	77.19	77.62	75.33
			cos	60.72	63.5	62.31	71.92	66.33	59.17	70.19	65.15	58.78	56.74	64.98	59.9	78.95	73.72	65.0
		rl	ICM	57.23	67.43	71.79	82.07	80.07	78.43	81.53	78.97	69.87	66.66	80.58	78.63	77.19	77.62	75.33
			cos	56.99	67.34	71.53	81.81	78.18	64.78	81.44	78.97	65.37	66.5	77.69	74.27	78.95	73.72	65.0
	last	lr	ICM	66.57	71.84	71.62	78.38	77.46	78.62	77.21	73.62	67.58	67.65	73.3	73.84	76.67	75.41	75.78
			cos	66.45	69.61	67.6	70.42	69.01	75.47	75.41	67.86	72.43	66.71	68.72	68.11	72.11	67.63	73.32
		rl	ICM	54.83	66.61	71.25	80.41	79.11	77.36	76.92	77.26	68.99	55.75	67.98	73.79	76.67	75.41	75.78
			cos	53.74	65.13	64.89	74.21	66.94	71.73	71.97	65.8	69.31	53.67	65.81	64.94	72.11	67.63	73.32
L_Maria_large_s	first	lr	ICM	60.72	63.53	62.83	71.82	69.93	69.38	69.75	67.06	61.46	56.73	64.77	61.69	78.09	77.82	75.13
			cos	60.72	63.33	62.5	71.56	66.34	59.23	68.9	65.15	59.77	56.92	64.82	60.47	79.33	73.73	65.0
		rl	ICM	56.5	67.61	71.07	82.07	80.07	78.24	81.53	78.92	70.26	66.83	80.41	78.28	78.09	77.82	75.13
			cos	57.74	67.06	70.64	81.81	77.81	64.81	81.63	79.16	65.25	67.06	77.84	74.47	79.33	73.73	65.0
	last	lr	ICM	65.26	71.88	71.27	78.06	77.55	78.46	77.96	74.83	69.05	65.6	73.15	71.32	76.31	75.4	75.96
			cos	64.62	71.84	70.5	65.17	70.45	74.92	68.3	68.22	74.41	64.38	70.5	71.02	66.24	71.47	72.81
		rl	ICM	65.41	73.56	75.44	79.7	78.24	76.84	78.19	76.09	65.26	64.45	73.91	76.33	76.31	75.4	75.96
			cos	60.32	60.7	62.99	72.7	70.99	70.29	69.11	64.4	64.81	60.39	60.86	62.51	66.24	71.47	72.81
L_STSB_c	first	lr	ICM	66.41	72.13	67.68	78.95	80.92	76.41	79.02	78.77	71.63	73.8	75.0	70.02	82.9	81.07	79.17
			cos	65.71	70.81	67.14	81.13	79.42	74.31	81.68	78.35	72.12	75.22	76.68	69.25	86.3	84.52	82.19
		rl	ICM	52.45	69.69	72.49	85.44	84.9	83.67	84.72	85.77	83.32	79.69	85.28	84.57	82.9	81.07	79.17
			cos	52.09	69.13	72.16	85.7	84.72	85.56	85.89	84.9	84.86	76.33	84.03	84.48	86.3	84.52	82.19
	last	lr	ICM	66.78	74.97	72.15	82.93	83.99	84.0	83.12	84.02	82.75	74.65	80.95	76.49	80.03	80.74	79.83
			cos	65.31	75.35	70.81	79.56	81.62	83.23	80.56	82.25	81.32	75.45	80.65	77.28	83.02	83.23	84.02
		rl	ICM	59.96	64.98	69.29	85.1	83.29	83.12	84.37	80.43	83.85	68.39	75.06	76.44	80.03	80.74	79.83
			cos	59.41	64.57	68.4	84.22	81.26	83.18	83.52	78.8	82.67	67.79	74.48	76.85	83.02	83.23	84.02
L_STSB_s	first	lr	ICM	66.42	70.24	68.19	81.12	82.35	78.61	80.21	82.72	76.49	76.5	75.71	73.42	83.82	81.97	80.47
			cos	66.25	70.38	68.75	82.37	82.73	76.46	82.38	80.33	76.11	77.58	79.29	72.68	86.14	84.73	83.97
		rl	ICM	54.11	68.79	73.22	85.81	85.82	85.28	85.45	85.95	83.67	81.85	84.55	86.18	83.82	81.97	80.47
			cos	54.64	68.93	73.41	85.67	84.91	86.46	86.05	84.91	86.65	81.33	85.08	87.02	86.14	84.73	83.97
	last	lr	ICM	63.83	73.14	70.2	83.66	82.59	84.93	84.56	83.68	83.14	74.15	80.97	79.2	81.56	79.75	81.19
			cos	63.83	72.51	69.23	83.16	80.29	82.83	83.72	79.76	81.33	75.49	78.52	78.62	85.56	82.79	85.39
		rl	ICM	76.34	71.41	78.48	82.94	82.95	83.48	82.59	83.13	82.96	75.97	80.26	81.34	81.56	79.75	81.19
			cos	76.34	70.78	78.48	85.37	80.98	83.74	83.94	79.18	82.13	77.17	77.03	80.41	85.56	82.79	85.39
MARIA_base	first	lr	ICM	62.75	68.05	65.0	73.9	77.17	74.9	72.63	74.33	67.99	60.7	70.72	67.14	79.7	78.94	78.75
			cos	62.41	67.48	65.2	77.03	75.34	69.04	75.42	74.86	67.78	63.24	72.23	66.36	81.06	79.07	73.87
		rl	ICM	53.35	65.94	70.36	81.64	81.86	78.07	81.85	80.22	76.82	78.21	80.21	80.07	79.7	78.94	78.75
			cos	53.39	65.11	69.34	83.52	80.79	77.48	83.15	81.41	77.98	75.52	80.3	80.43	81.06	79.07	73.87
	last	lr	ICM	67.41	74.46	77.94	77.93	77.69	77.35	77.98	73.64	65.74	69.13	70.53	75.6	77.56	75.95	76.68
			cos	63.57	70.51	70.76	78.62	74.24	80.64	76.04	72.06	77.07	67.24	70.28	70.94	80.3	74.51	79.17
		rl	ICM	73.18	69.76	77.67	80.15	79.17	77.54	80.36	71.94	62.92	71.87	68.63	75.34	77.56	75.95	76.68
			cos	66.06	62.81	65.81	75.25	71.19	77.34	72.0	65.7	73.58	66.41	63.48	66.84	80.3	74.51	79.17
MARIA_large	first	lr	ICM	59.63	63.53	63.38	71.3	70.48	69.15	69.04	67.79	61.83	55.84	65.28	61.87	78.64	77.83	75.03

			cos	59.81	63.67	62.86	71.2	66.15	60.53	69.09	64.79	59.81	55.67	65.35	59.93	78.93	73.74	64.37
		rl	ICM	57.09	66.89	71.44	81.89	80.07	78.24	81.89	78.76	71.16	65.38	80.76	77.92	78.64	77.83	75.03
			cos	57.37	66.59	70.62	81.81	77.98	65.12	81.82	79.14	65.59	67.01	76.73	74.05	78.93	73.74	64.37
	last	lr	ICM	59.26	69.28	62.65	78.58	78.59	76.28	78.49	72.57	58.52	60.88	70.26	65.09	76.49	76.49	76.32
			cos	56.49	62.42	58.66	64.95	68.74	59.72	64.28	67.0	59.41	57.42	63.52	59.33	67.05	67.89	62.36
		rl	ICM	69.94	74.75	72.09	79.51	78.77	77.0	78.7	71.22	55.32	69.56	74.6	71.93	76.49	76.49	76.32
			cos	54.74	59.1	50.93	65.01	67.38	60.65	62.58	62.56	54.78	54.7	60.48	52.99	67.05	67.89	62.36
STSB	first	lr	ICM	66.15	69.28	67.0	79.36	86.18	80.41	79.88	86.0	80.08	79.74	84.39	78.45	81.97	83.99	81.39
			cos	66.51	68.56	65.16	84.16	87.52	80.8	84.17	85.86	79.9	79.38	85.3	76.64	88.12	88.45	87.39
		rl	ICM	62.0	73.98	75.21	86.17	86.36	86.35	86.36	86.89	87.62	82.59	86.35	87.08	81.97	83.99	81.39
			cos	62.35	75.21	75.58	87.74	87.53	87.75	86.82	86.99	87.21	83.02	86.42	88.13	88.12	88.45	87.39
	last	lr	ICM	75.39	59.76	65.7	76.01	73.7	75.28	74.84	69.6	73.04	74.48	65.51	70.36	77.37	75.65	76.96
			cos	74.31	58.33	66.27	76.11	67.44	72.52	74.48	65.88	70.58	74.27	64.27	70.02	76.45	68.69	72.9
		rl	ICM	74.85	60.23	64.93	76.3	72.7	75.77	74.13	68.79	70.19	75.21	65.72	67.35	77.37	75.65	76.96
			cos	75.02	60.08	64.78	75.38	67.32	70.93	73.57	64.39	68.04	74.84	62.91	67.12	76.45	68.69	72.9
Word2vec	first	lr	ICM	56.25	63.75	66.78	77.18	77.89	78.06	77.17	77.78	78.88	73.77	77.74	78.38	78.69	78.69	78.69
			cos	56.12	63.55	66.23	76.63	79.5	81.31	74.26	78.43	79.68	68.71	76.4	77.71	82.04	82.04	82.04
		rl	ICM	57.32	64.83	66.67	78.78	79.94	79.56	79.18	79.25	79.75	78.39	80.28	80.16	78.69	78.69	78.69
			cos	58.29	66.95	67.62	82.21	81.86	82.39	81.48	80.76	80.77	77.17	78.42	80.04	82.04	82.04	82.04
	last	lr	ICM	56.25	63.75	66.78	77.18	77.89	78.06	77.17	77.78	78.88	73.77	77.74	78.38	78.69	78.69	78.69
			cos	56.12	63.55	66.23	76.63	79.5	81.31	74.26	78.43	79.68	68.71	76.4	77.71	82.04	82.04	82.04
		rl	ICM	57.32	64.83	66.67	78.78	79.94	79.56	79.18	79.25	79.75	78.39	80.28	80.16	78.69	78.69	78.69
			cos	58.29	66.95	67.62	82.21	81.86	82.39	81.48	80.76	80.77	77.17	78.42	80.04	82.04	82.04	82.04
Word2vec-SPL	first	lr	ICM	58.61	67.88	66.91	80.35	80.34	82.17	79.78	83.31	83.68	78.98	84.04	82.06	81.4	81.4	81.4
			cos	57.94	66.12	64.96	86.18	85.65	86.18	85.83	85.82	85.27	82.22	85.28	83.65	84.9	84.9	84.9
		rl	ICM	60.22	65.16	66.41	81.34	82.7	82.51	80.8	81.27	83.13	79.14	80.78	82.22	81.4	81.4	81.4
			cos	59.25	66.25	64.55	81.47	82.55	83.11	80.92	81.49	81.49	77.71	80.95	81.67	84.9	84.9	84.9
	last	lr	ICM	58.61	67.88	66.91	80.35	80.34	82.17	79.78	83.31	83.68	78.98	84.04	82.06	81.4	81.4	81.4
			cos	57.94	66.12	64.96	86.18	85.65	86.18	85.83	85.82	85.27	82.22	85.28	83.65	84.9	84.9	84.9
		rl	ICM	60.22	65.16	66.41	81.34	82.7	82.51	80.8	81.27	83.13	79.14	80.78	82.22	81.4	81.4	81.4
			cos	59.25	66.25	64.55	81.47	82.55	83.11	80.92	81.49	81.49	77.71	80.95	81.67	84.9	84.9	84.9

Table A.2: Unsupervised Composition F1 Scores

A.4 Special Tokens Experiment Results

The figure below presents the results from our analysis of the impact of special tokens on the semantic composition process within contextual models. This experiment was designed to compare three distinct configurations: the inclusion of all special tokens within the sequence, the exclusion of special tokens during inference, and the provision of all tokens to the model with subsequent removal of special tokens before performing semantic composition operations.

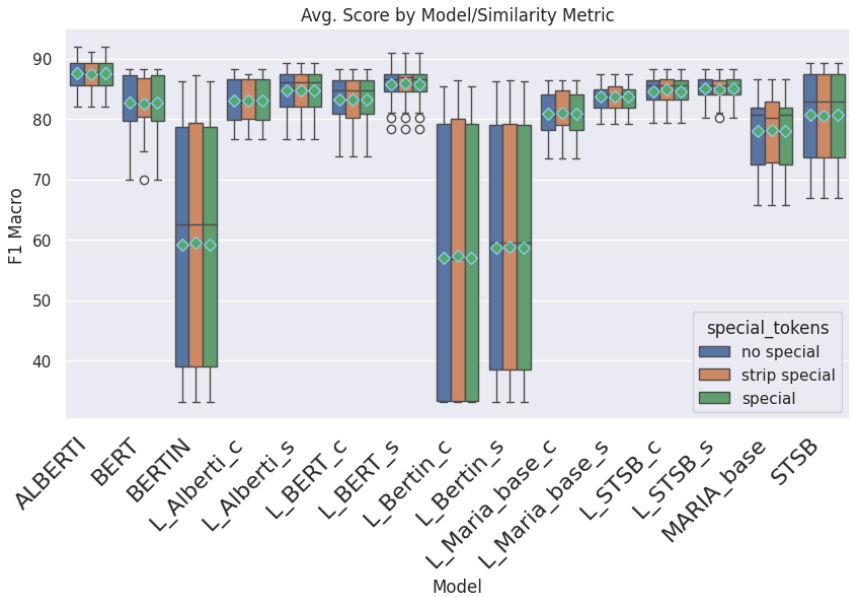


Figure A.2: Comparative analysis of composition with different special token configurations.

Remarkably, the data clearly demonstrate that the variations in performance across different configurations are minimal, with all observed differences falling within the established margin of error.