UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE POLITICHE,
ECONOMICHE E SOCIALI

Master in Data Science and Economics

Evaluating and Enhancing Figurative Language Understanding in Large Language Models:

A Study on Multi-Type Figurative Expressions Using the FLUTE Dataset

Supervisor: Dr. Alfio Ferrara

Co-Supervisor:

Master's Candidate:

Ashkan Samavatian (965235)

To my beloved Parents, Sasan and Hengameh, whose unconditional love, support, and sacrifices have guided me throughout my life.

To my wonderful Wife, Farzan, for her unwavering support, patience, and love during this challenging journey.

To my dear Siblings, Arman and Hasti, for always believing in me and encouraging me to pursue my dreams.

And to the cherished memory of Amir, who, although no longer with us, continues to inspire me for perseverance and patience.

This work is a testament to your love, support, and belief in me.

Table of Contents

Index of Tables and Figures

1) Abstract

This thesis investigates the figurative language understanding of transformer-based large language models (LLMs), including ROBERTA, BERT, and T5, using the FLUTE dataset, which covers sarcasm, idioms, similes, and metaphors. Despite advancements in NLP, figurative language remains a challenging domain due to its reliance on implicit meaning, context, and cultural knowledge. This study explores whether fine-tuning on multi-type figurative datasets enhances the performance of transformer-based models in understanding and classifying figurative language.

The research introduces a probing task inspired by Holmes' contrastive analysis, assessing model preferences between figurative and literal meanings through log-likelihood scoring and statistical tests. Fine-tuning experiments were conducted on ROBERTA and BERT with optimized hyperparameters, and their performance was compared with T5's generative text-to-text approach. The evaluation incorporated multi-metric analysis, including accuracy, precision, recall, BLEURT, and BERTScore, along with comprehensive error analysis and bias investigation.

Results indicate that fine-tuning improves models' performance in general. This research contributes to NLP by offering a deeper understanding of figurative language processing and highlighting the strengths and limitations of transformer-based models.

Keywords:

Figurative Language, Transformer-Based Large Language Models, BERT, ROBERTA, T5, FLUTE Dataset, Probing Task, NLP, Fine-Tuning, Sarcasm Detection, Idiom, Simile, Metaphor, Multi-Metric Evaluation.

2) Introduction

2-1) Background and Motivation

Figurative language is an essential component of human communication, enriching discourse with abstract, non-literal meanings that go beyond conventional word definitions. Sarcasm, idioms, similes, and metaphors among other figurative expressions, are pervasive in both spoken and written communication, serving various cognitive and social functions such as persuasion, humor, and conceptual understanding. However, these forms of language pose significant challenges for computational models, as they often require deep semantic interpretation, contextual inference, and cultural knowledge elements that traditional natural language processing (NLP) systems struggle to capture effectively.

With the advent of transformer-based large language models (LLMs) such as ROBERTA, BERT, and T5, NLP systems have achieved unprecedented success in tasks such as sentiment analysis, text generation, and language translation. However, despite their remarkable progress in understanding and processing natural language, these models still exhibit substantial limitations when dealing with figurative expressions. Unlike literal language, figurative language often relies on implicit meanings, context shifts, and speaker intent, which can cause models to misinterpret or fail to classify such expressions correctly.

Given the increasing reliance on AI-driven communication tools, chatbots, and automated text processing systems, ensuring that these models can effectively process figurative language is crucial. Misinterpretations of figurative expressions can lead to significant errors in real-world applications, from misinformation in automated content moderation to breakdowns in user interactions with AI assistants. Therefore, there is a growing need for systematic evaluation and enhancement of figurative language comprehension in transformer-based models to improve their interpretative accuracy and generalization capabilities.

2-2) Research Problem and Objectives

Despite the improvements in NLP models, figurative language understanding remains an open challenge. Prior studies have primarily focused on figurative language classification, assessing models based on their ability to categorize expressions into predefined figurative types. However, classification alone does not necessarily reflect whether a model truly comprehends figurative meaning. The primary research question that this thesis seeks to address is:

"Can fine-tuning on multi-type figurative datasets like FLUTE improve the performance of transformer-based large language models in understanding and classifying figurative language?"

To answer this question, the study focuses on three main objectives:

1) **Evaluating Figurative Language Understanding Beyond Classification:** This research introduces a probing task inspired by Holmes' contrastive approach to examine whether models prefer figurative or literal meanings when both options are available. By leveraging sentence likelihood scoring, statistical hypothesis testing, and log-likelihood difference analysis, the study investigates whether transformer-based models truly grasp figurative meaning rather than merely classifying expressions correctly.
2) **Enhancing Model Performance Through Fine-Tuning:** The study systematically fine-tunes ROBERTA and BERT using optimized hyperparameters and compares their performance against the generative T5 model. This evaluation quantifies the impact of fine-tuning on figurative language classification and assesses whether additional training on multi-type figurative datasets like FLUTE improves model accuracy.

3) **Comprehensive Error Analysis and Bias Investigation:** Unlike prior research that mainly reports accuracy scores, this study conducts a detailed error analysis to examine misclassification patterns across different figurative types. Additionally, dataset biases and their impact on model performance are analyzed to identify potential generalization issues.

By addressing these objectives, this study contributes to the field of NLP by offering a more nuanced understanding of how transformer-based models process figurative language and exploring novel ways to enhance their interpretative abilities.

2-3) Scope and Contributions

This thesis specifically focuses on transformer-based large language models (LLMs), distinguishing them from other AI-based linguistic models. While the term large language models (LLMs) is sometimes used broadly, in this study, it refers explicitly to transformer-based architectures, including ROBERTA, BERT, and T5. These models were chosen due to their widespread adoption in NLP research and their state-of-the-art performance in various language tasks. The research contributes to the field of NLP and figurative language processing in the following ways:

1) **Probing Figurative Understanding Beyond Classification:** Unlike conventional classification-based evaluations, this study employs a contrastive probing task to assess whether models exhibit a genuine preference for figurative over literal meanings.
2) **Comprehensive Model Benchmarking:** By systematically comparing ROBERTA, BERT, and T5 on the FLUTE dataset, this research provides an in-depth analysis of their strengths, weaknesses, and generalization capabilities across different figurative language types.

3) **Fine-Tuning and Hyperparameter Optimization:** The study explores the impact of hyperparameter tuning on figurative language classification, demonstrating improvements in model accuracy and generalization.
4) **Application of T5 for Figurative Language Processing:** This research uses T5 for figurative language classification, treating it as a text-to-text generation task rather than a conventional classification problem.
5) **Multi-Metric Evaluation for Linguistic Understanding:** Beyond traditional accuracy and F1-score metrics, the study incorporates BLEURT and BERTScore to assess semantic similarity, ensuring a more linguistically meaningful evaluation.

By addressing these contributions, the research advances figurative language understanding in transformer-based NLP models, paving the way for more sophisticated AI-driven language comprehension.

2-4) Methodology Overview

To investigate the research question, this study employs a systematic evaluation of transformer-based LLMs using the FLUTE dataset, which consists of figurative expressions across multiple categories: sarcasm, idioms, similes, and metaphors. The study follows a three-step methodology:

1) **Probing Task for Figurative Understanding:**
    - Inspired by Holmes' approach, a contrastive sentence-pair evaluation is implemented.
    - Models are tested on their preference for figurative vs. literal interpretations using log-likelihood scoring and statistical tests (T-test, Wilcoxon signed-rank).

2) **Fine-Tuning Transformer Models for Figurative Language Classification:**
    - ROBERTA and BERT are fine-tuned on the FLUTE dataset using optimized hyperparameters such as learning rate adjustments, gradient checkpointing, and label smoothing.
    - The T5 model is adapted to treat figurative classification as a text-to-text generation problem.
3) **Evaluation and Comparative Analysis:**
    - Performance is assessed across multiple metrics, including accuracy, precision, recall, BLEURT, and BERTScore.
    - Error analysis and dataset bias evaluations are conducted to examine misclassification trends.

This methodology ensures a comprehensive evaluation of transformer-based models, offering insights into both their classification performance and deeper linguistic competence in figurative language processing.

This study underscores the need for refined model architectures and training methodologies, aiming to advance AI-driven communication tools with nuanced figurative language understanding.

3) Literature Review

3-1) Overview of Figurative Language in NLP

Figurative language includes expressions that go beyond their literal meanings, often requiring cultural, contextual, or cognitive understanding to interpret. Common forms include metaphors, idioms, similes, sarcasm, and hyperboles. These constructs play a crucial role in communication, allowing for creativity, persuasion, and emotional expression. However, their inherent complexity makes them a challenging area for computational models to process (Gibbs, 1994). Figurative language plays a significant role in enhancing the depth and richness of both written and spoken communication (Florman, 2017). From a cognitive science perspective, understanding figurative language is critical as it involves abstract thought processes and interpretive adjustments to individual words. This adjustment allows individuals to derive meanings that go beyond the literal sense of words (Michaeli et al., 2008). Michaeli and colleagues highlight the intricate cognitive mechanisms involved in interpreting figurative expressions, emphasizing the relevance of figurative language in both linguistic and psychological studies (Michaeli et al., 2008).

Furthermore, in lexicographical studies, figurative language is viewed as a transference or extension of meaning from its literal sense. This perspective underscores its prevalence in everyday communication, highlighting its role in shaping effective and nuanced language use (Deignan, A., 2015). Springer’s Lexicographical Encyclopedia describes figurative language as an essential component of human interaction, illustrating its integration into both casual and formal language contexts (Hanks, P., de Schryver, GM., 2015). Figurative language enriches human interactions by allowing speakers to express nuanced ideas and emotions concisely. Various studies emphasize its prevalence in daily communication, literature, and digital media (Lal, Y., & Bastan, M., 2022). For instance, idiomatic expressions can convey cultural meanings that literal language cannot capture, posing challenges for Large Language Models (LLMs) that primarily rely on surface-level text comprehension (Imran, M.M. et al., 2023).

This indicates a need for LLMs to be equipped with deeper semantic understanding capabilities to effectively process and generate figurative language.

3-2) Definitions of Figurative Language Forms

Figurative language encompasses various linguistic constructs that go beyond their literal meanings, playing a critical role in enriching communication. Each form of figurative expression contributes uniquely to the depth and nuance of language, enabling the conveyance of abstract ideas, emotions, and cultural nuances. Among these, metaphors stand out as conceptual mappings between domains, allowing one concept to be understood in terms of another. For example, the metaphor “Time is a thief” compares the intangible concept of time to the concrete image of a thief, illustrating the power of metaphor to encapsulate complex relationships (Lakoff & Johnson, 1980). Idioms further illustrate the intricacy of figurative language. These fixed expressions, such as “spill the beans” (reveal a secret) or “kick the bucket” (to die), defy interpretations based on the meanings of their words. Their context-dependent nature makes them particularly challenging for computational models to decode accurately (Katz & Fodor, 1963). Similarly, similes employ explicit comparisons to convey meaning, using connectors like “like” or “as.” Phrases such as “She is as fierce as a lion” directly liken one concept to another, offering a vivid, relatable depiction of bravery (Veale, 2012). Likewise, sarcasm, a more complex form of figurative language, uses irony to mock or convey contempt, often requiring an understanding of tone and intent. Expressions like “Oh great, another rainy day” may mean the opposite of their literal wording, posing interpretive challenges even for advanced language models (Ghosh et al., 2018). Hyperboles, by contrast, achieve emphasis through intentional exaggeration. Statements such as “I’ve told you a million times” exemplify this form, where the exaggeration underscores the speaker’s frustration or urgency (Colston & O'Brien, 2000).

Other forms, such as personification, metonymy, and synecdoche, add layers of richness to figurative language by imbuing non-human entities with human characteristics, substituting closely associated terms, or allowing parts to represent wholes, respectively. For instance, “The wind whispered through the trees” anthropomorphizes nature to create an evocative image while referring to the White House as the “executive branch” demonstrates metonymy’s utility in communication (Long, 2018; Sun, 2024). Synecdoche operates similarly by focusing on a specific aspect, as seen when the term “sails” is used to represent an entire ship (Brown, 1979). Irony, onomatopoeia, and alliteration further illustrate the versatility of figurative language. Irony conveys meanings opposite to their literal interpretation, often creating unexpected or contradictory outcomes, such as in the scenario of a traffic cop penalized for unpaid parking tickets (Nurdinova et al., 2022). Onomatopoeia, by mimicking sounds, enriches narratives with auditory characteristics, as demonstrated in “The bees buzzed busily among the blooming flowers” (Bredin, 1996). Meanwhile, alliteration enhances the aesthetic appeal of language through the repetition of initial consonant sounds, exemplified in “The swift, silent serpent slithered seamlessly” (Semaeva, 2024).

Puns, or paronomasia, add a playful dimension by exploiting multiple meanings of a term or similar-sounding words for rhetorical or humorous effect. For example, the phrase “I like kids, but I don’t think I could eat a whole one” humorously juxtaposes the meanings of “kid” as both a young child and a young goat, showcasing the interplay of meanings (Giorgadze, 2014).

Together, these forms of figurative language highlight the intricacies of human communication and underline the challenges they present to natural language understanding systems. The ability to process and interpret these constructs is essential for developing effective language models and advancing applications in NLP.

3-3) Importance in NLP

Processing figurative language is crucial in Natural Language Processing (NLP) due to its diverse applications and the complexities it introduces. (Hyewon Jang., et al., 2023) Figurative language includes varieties such as metaphors, similes, idioms, and puns, which convey meanings that extend beyond their literal interpretation. (Bisikalo, O.V., et al., 2019) Understanding these forms of language is fundamental for several key applications in NLP, including sentiment analysis, dialogue systems, and creative text generation.

In sentiment analysis, accurately interpreting figurative language is essential to gauge the true emotional tone behind textual data. (Nguyen, H. L., et al., 2015) For example, a statement like “I'm on cloud nine” suggests extreme happiness rather than a literal interpretation, which could mislead sentiment classification models. (Karamouzas, D., et al., 2022) The presence of figurative expressions can significantly affect the sentiment expressed in a sentence, necessitating models that can recognize and appropriately interpret these nuances. Effective sentiment analysis enhances applications such as customer feedback systems and social media monitoring, allowing for a more accurate understanding of public sentiment. (Tomáš Hercig., et al., 2017)

Dialogue systems, including chatbots and virtual assistants, require a robust understanding of figurative language to engage users effectively. (Karamouzas, D., et al., 2022) In human conversations, figurative expressions are often employed and failing to recognize these can lead to misunderstandings or inappropriate responses. For instance, if a user states, “It's raining cats and dogs”, the system must interpret this phrase correctly to provide relevant responses instead of a literal interpretation, which could confuse the conversation flow. (Tummala, P., et al., 2024) Developing NLP systems that can process and respond to figurative language helps improve user experience and communication efficacy. (Jhamtani, H., et al., 2021)

Creative text generation, an emerging area within NLP, often incorporates figurative language to produce engaging narratives or generate artistic expressions. Figurative language enriches storytelling by adding depth and resonance, allowing for more expressive and relatable content. (Repeko, A.P., et al., 2001) For instance, systems designed to generate poetry, or narrative prose must leverage onomatopoeia, metaphors, and other figures of speech to connect with audiences emotionally. (Vulchanova, M.D., et al., 2018) The ability to understand and generate figurative expressions enhances the quality of the texts produced and broadens the potential applications of NLP technology in fields like marketing and entertainment. (Mason, Z.J. 2004)

Despite the importance of processing figurative language, its inherently nonliteral nature poses significant challenges for NLP systems. (Nagels, A., et al., 2013) The disparity between literal and figurative meanings often confounds algorithms, leading to misinterpretations. (Reyes, A. 2013) Many NLP models operate on statistical or machine learning principles that rely heavily on context, but figurative language frequently subverts straightforward contextual cues. This complexity necessitates advanced model architectures, such as those leveraging deep learning techniques, to better capture the subtleties of language use. (Potamias, R.A., et al., 2020)

A significant barrier is the scarcity of annotated datasets tailored to figurative language. Existing datasets often emphasize literal language, creating a training deficit for models handling figurative forms. Without sufficient diversity in training data, models struggle to generalize across different types of figurative expressions. To address these gaps, ongoing research seeks to develop diverse datasets that reflect the complexity of figurative language, enabling models to distinguish effectively between literal and figurative meanings. This work is critical for advancing NLP systems to handle the nuanced and context-dependent nature of figurative communication. (Huiyuan Lai., et al., 2024)

3-4) Review of Previous Work on Figurative Language Understanding

Research into the understanding of figurative language in NLP has progressed significantly over the years, evolving through four primary paradigms: rule-based systems, statistical models, neural models, and transformer-based models. Each of these approaches has contributed unique insights and advancements to the field, addressing various challenges posed by the nonliteral and context-dependent nature of figurative expressions.

Early research in figurative language understanding focused on constructing models to decode individual linguistic phenomena. However, as the complexity of figurative expressions became apparent, the need for unified approaches to evaluate linguistic phenomena, including those underlying figurative constructs, gained importance.

Rule-based systems, which rely on manually crafted rules or linguistic patterns, represent some of the earliest efforts in this area. Katz and Fodor (1963) introduced one of the first semantic theories for idioms, highlighting the necessity of non-compositional interpretations. While these systems are interpretable and effective for predefined expressions, they are limited by their inability to scale and adapt to novel or unseen figurative constructs.

The advent of statistical models marked a shift towards probabilistic approaches for figurative language detection. Techniques such as Latent Semantic Analysis (LSA) enabled researchers to model semantic similarities between words and phrases, aiding in the identification of figurative relationships. For instance, Turney et al. (2001) utilized statistical patterns to uncover metaphorical connections by comparing word embeddings within their contextual usage. These models offered improved flexibility over rule-based approaches but still faced challenges in handling complex or deeply nuanced expressions.

The introduction of neural models, particularly recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, represented another significant step forward. These models demonstrated promise in detecting figurative language by capturing sequential dependencies within text, a feature critical for understanding idiomatic and metaphorical phrases. As Rei et al. (2017) noted, this ability to process sequential patterns brought greater sophistication to figurative language analysis. However, the limitations of RNNs and LSTMs in capturing long-range dependencies hindered their overall effectiveness, leaving room for further innovation.

Transformer-based architectures, such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), have revolutionized the field, leveraging self-attention mechanisms to model global contextual relationships. These advancements have significantly improved the performance of figurative language understanding tasks, including metaphor detection (Liu, J., et al., 2020) and idiom classification (Feldmann et al., 2021). Despite their success, transformer models are not without limitations. They continue to struggle with non-compositionality and the nuanced contextual cues inherent in figurative language, particularly when interpreting sarcasm. Moreover, as transformer-based language models (TLMs) are not intrinsically explainable, one way of inspecting the reasons behind their decisions is through post-hoc feature analysis (Hyewon Jang et al., 2023). This lack of inherent interpretability further complicates their ability to handle complex figurative expressions, emphasizing the need for further refinement and innovation to address these challenges fully.

To address these issues, more comprehensive benchmarks have been introduced to evaluate linguistic phenomena systematically. The Holmes benchmark, for instance, offers structured evaluations by isolating linguistic competence from linguistic performance. This methodology enables better assessment of abstract constructs like metaphors and sarcasm, complementing existing approaches in figurative language understanding (Waldis et al., 2024).

The evolution of approaches to figurative language understanding underscores the growing complexity and sophistication of NLP systems, as well as the persistent challenges posed by the abstract and context-dependent nature of figurative expressions. Each paradigm has contributed valuable advancements, paving the way for continued research and development in this critical area.

3-5) Datasets for Figurative Language Understanding

Datasets are integral to training and evaluating models for figurative language understanding, providing the foundational material that enables computational systems to interpret and process nonliteral expressions. Among the most widely utilized resources is the FLUTE dataset, which is specifically designed for multi-type figurative language understanding. This dataset provides natural language inference (NLI) pairs annotated for entailment and contradiction, encompassing a range of figurative forms such as idioms, metaphors, sarcasm, and similes (Chakrabarty et al., 2022). Its comprehensive scope makes it a valuable resource for evaluating the performance of models across various figurative types.

The VUA Metaphor Corpus offers another significant contribution by focusing on metaphorical language, particularly verbs. It provides detailed annotations that distinguish between metaphorical and literal meanings, making it an essential tool for understanding the nuanced use of metaphors (Steen et al., 2010). For idiomatic expressions, resources like the Cambridge Idiom Corpus and idiomatic annotations from Wiktionary enable models to capture the contextual subtleties that define idioms. These datasets contribute to the development of systems capable of recognizing and interpreting idiomatic language, which often resists straightforward compositional analysis.

Sarcasm, with its reliance on tone, context, and intent, presents unique challenges for computational models. The Sarcasm Corpus, composed of annotated tweets and comments, facilitates the training of systems to detect sarcasm and understand its often contradictory nature (Ghosh et al., 2018). These datasets collectively address specific facets of figurative language, providing valuable resources for targeted model development.

However, despite their importance, these datasets exhibit notable limitations. Most focus on a single type of figurative expression, such as metaphors, idioms, or sarcasm, which restricts their utility for generalization across multiple figurative forms. This narrow focus often results in models that excel in one domain but fail to adapt effectively to others. The FLUTE dataset stands out in this regard, as its multi-type coverage provides a broader and more balanced foundation for evaluating and enhancing models' comprehensive figurative understanding capabilities. Addressing these limitations through the development of more diverse and inclusive datasets remains a critical avenue for advancing the field of figurative language understanding.

3-6) Evaluation Methods

Evaluating figurative language understanding in NLP involves a combination of automated and human metrics to comprehensively assess model performance. Automated metrics provide an efficient means of evaluation, with accuracy and F1-score serving as standard measures for classification tasks such as metaphor detection. These metrics assess the precision and recall of models, offering a clear picture of their ability to classify figurative language accurately.

Metrics like BLEU and ROUGE are frequently employed for text similarity tasks, particularly in paraphrasing applications. BLEU (Papineni et al., 2002) evaluates the correspondence between generated text and reference text using n-gram overlap, while ROUGE focuses on recall-oriented measures for the same purpose. Both metrics provide valuable insights into the quality of model-generated text but often fall short in capturing the subtleties of figurative expressions. BERTScore, a more recent contextual embedding-based metric, evaluates semantic similarity by leveraging the embeddings from pre-trained language models like BERT (Zhang et al., 2020). This metric offers improved sensitivity to the context and meaning of figurative expressions, making it particularly suitable for evaluating nuanced language use.

Human evaluation, however, remains indispensable for assessing aspects of figurative language that automated metrics often overlook. Human evaluators typically use Likert scales to rate fluency, figurative accuracy, and contextual appropriateness, providing a nuanced understanding of model performance. This approach is especially valuable for evaluating complex constructs like sarcasm and idiomaticity, where contextual understanding and intent play a significant role. For instance, as Ghosh et al. (2018) emphasize, the detection of sarcasm often depends on subtle contextual cues and tone, which are difficult for automated metrics to capture reliably.

By integrating automated metrics with human evaluation, researchers can achieve a more holistic assessment of figurative language understanding. This combination allows for both scalability and depth, ensuring that models are rigorously evaluated across a spectrum of linguistic challenges. Such comprehensive evaluation frameworks are essential for advancing NLP systems capable of interpreting and generating figurative language with accuracy and contextual sensitivity.

3-7) Research Gaps

Despite the progress made in figurative language understanding, several significant gaps and challenges persist, limiting the effectiveness and generalizability of current models. One prominent issue lies in the difficulty of generalization across different types of figurative language. Models trained in specific types, such as metaphors or sarcasm, often fail to adapt to other forms of figurative expressions. This specialization limits the broader applicability of these models, underscoring the need for more versatile and comprehensive approaches.

Furthermore, recent studies highlight that large-scale models often struggle with complex phenomena such as reasoning and discourse, which are critical for understanding figurative expressions. These deficiencies are evident even in state-of-the-art benchmarks, revealing gaps in evaluating models' ability to generalize across multiple linguistic phenomena (Waldis et al., 2024). This limitation parallels challenges in figurative language understanding, particularly in capturing contextual subtleties like sarcasm and idiomatic expressions.

Another critical challenge is the requirement for deep contextual reasoning, particularly in understanding sarcasm and idioms. Sarcasm detection often hinges on subtle contextual cues, such as tone and intent, which are not always explicitly available in textual data. Similarly, idiom interpretation demands a nuanced understanding of cultural and linguistic context, further complicating computational processing. Current models frequently struggle to grasp these intricacies, resulting in misinterpretations and reduced performance in real-world applications.

Evaluation methods also present a notable limitation in advancing figurative language understanding. Automated metrics, while efficient and widely used, often fail to capture the creativity and subtlety inherent in figurative expressions. These metrics are typically designed for tasks like classification or text similarity and may not fully reflect the complexities of figurative language. Consequently, there is a pressing need for improved evaluation frameworks that integrate nuanced human judgments with automated methods. Such frameworks would provide a more accurate and holistic assessment of model performance, facilitating the development of systems capable of addressing the multifaceted nature of figurative language.

3-8) Addressing the Gaps

This thesis aims to address the existing gaps in figurative language understanding by leveraging and extending the capabilities of transformer-based large language models (LLMs) such as ROBERTA, BERT, and T5. These models will be evaluated on the FLUTE dataset, a resource specifically designed for multi-type figurative language understanding. The evaluation will provide insights into the strengths and limitations of current transformer-based LLMs in handling diverse figurative expressions, including idioms, metaphors, sarcasm, and similes.

To enhance contextual understanding, this research proposes the implementation of advanced techniques such as fine-tuning and innovative preprocessing strategies. These enhancements are intended to equip models with a deeper ability to discern the subtleties of figurative language, particularly in challenging contexts like sarcasm detection and idiom interpretation. In addition to these techniques, recent advancements in evaluation methodologies, such as classifier-based probing as implemented in the Holmes benchmark, provide a structured approach to isolating linguistic competence from linguistic performance (Waldis et al., 2024). Adapting these methods to figurative language processing can enable a more targeted understanding of nuanced constructs like metaphors and sarcasm.

By addressing these challenges, this study seeks to advance the field of figurative language processing, contributing novel insights into the capabilities and limitations of transformer-based LLMs.

3-9) Research Question

Based on the identified gaps: “Can fine-tuning on multi-type figurative datasets like FLUTE improve the performance of transformer-based large language models in understanding and classifying figurative language?”

4) Methodology

Before delving into the detailed description of the methods employed in this research, it is essential to provide an overview of the key components that form the foundation of this study.

4-1) Dataset Overview

The FLUTE dataset (Figurative Language Understanding through Textual Explanations) is a benchmark dataset designed to evaluate the comprehension of figurative language in natural language processing (NLP) models. It consists of pairs of premises and hypotheses, where the hypothesis contains a figurative expression, and the premise provides supporting context. The dataset covers four primary types of figurative language: Sarcasm, Idiom, Simile, and Metaphor (Chakrabarty et al., 2022). Each sample in the dataset also includes a literal explanation, providing an alternative non-figurative interpretation of the hypothesis.

The FLUTE dataset is structured to support textual entailment tasks, where the model is required to determine whether the premise entails the hypothesis. Its comprehensive coverage of figurative language types and annotated explanations makes it an ideal resource for training and evaluating transformer-based language models in this domain (Chakrabarty et al., 2022). In this study, the evaluation focused on four types of figurative language:

- **Sarcasm:** Sarcasm involves using irony to mock or convey contempt, often requiring an understanding of tone, context, and intent. For instance, "Oh great, another rainy day." might mean the opposite of what is stated (Ghosh et al., 2018).
- **Idiom:** Idioms are fixed expressions whose meanings cannot be derived directly from individual words. For example, "spill the beans." means "reveal a secret" or "kick the bucket" means "to die", and its meaning is contextually dependent. (Katz & Fodor, 1963).
- **Simile:** Similes are explicit comparisons between two concepts using connectors like "like" or "as". For example, "She is as fierce as a lion." or "She is as brave as a lion." conveys bravery through a direct comparison (Veale, 2012).
- **Metaphor:** Metaphor represents conceptual mappings between two domains, where one concept (the source) is understood in terms of another

(the target). For example, “Time is a thief.” compares the abstract concept of time to the concrete idea of a thief (Lakoff & Johnson, 1980).

4-2) Probing Framework for Figurative Language Evaluation Overview

The evaluation of figurative language understanding in this research relies on a probing framework that leverages contrastive sentence pairs. This section outlines the conceptual foundation and implementation of the Holmes Probing Task framework, highlighting its significance in assessing the nuanced comprehension capabilities of language models.

4-2-1) Holmes Probing Task

The Holmes Probing Task is inspired by the conceptual approach of using contrastive pairs to evaluate language understanding, particularly in figurative contexts (Waldis et al., 2024). In this study, pre-trained transformer models were evaluated on their ability to differentiate between figurative and literal sentences.

This probing task involved presenting the models with pairs of sentences: one containing a figurative expression and the other providing a literal interpretation of the same expression. The model's preference for one sentence over the other, based on computed log-likelihood scores, was used as an indicator of its figurative language comprehension. The methodology aligns with the probing framework which emphasizes contrastive evaluation to unveil linguistic capabilities in language models (Waldis et al., 2024).

4-2-2) Metrics for Holmes Probing Task

The probing task utilized unique evaluation metrics tailored to the nature of contrastive language understanding:

- **Log-Likelihood Difference:** The difference in log-likelihood scores between the figurative and literal sentences. A positive value indicates a preference for the figurative sentence, while a negative value indicates a preference for the literal sentence. (Waldis et al., 2024)
- **Confidence Score:** Calculated as the absolute difference in log-likelihood scores. This metric measures the strength of the model's preference, providing insight into how confidently the model distinguishes between figurative and literal language. (Rogers et al., 2020)
- **Statistical Significance Tests:**
  - **T-Test:** Used to determine if the mean log-likelihood scores of figurative and literal sentences differ significantly. (Raschka, S., & Mirjalili, V. 2019)
  - **Wilcoxon Signed-Rank Test:** A non-parametric test used to verify the consistency of the model's preferences across samples. (Wilcoxon, F. 1945)

4-3) Models Overview

Three transformer-based large language models were utilized in this study: RoBERTa, BERT, and T5. Each model represents a state-of-the-art approach in NLP, with unique architectures and training methodologies.

4-3-1) BERT (Bidirectional Encoder Representations from Transformers)

BERT is a bidirectional transformer model designed to pre-train deep bidirectional representations by jointly conditioning on both left and right contexts. Introduced by Devlin et al. (2019), BERT uses a masked language modeling (MLM) objective, where random tokens are masked, and the model learns to predict them. This enables BERT to capture rich contextual representations and has become the foundation for numerous downstream NLP tasks.

4-3-2) RoBERTa (A Robustly Optimized BERT Pretraining Approach)

Roberta is an improved version of BERT, introduced by Liu et al. (2019), with optimizations in pre-training procedures. RoBERTa removes the next sentence prediction objective, uses dynamic masking, and is trained with larger batches and more data. These improvements make RoBERTa more robust and efficient in capturing linguistic nuances, including figurative language (Liu et al., 2019).

4-3-3) T5 (Text-To-Text Transfer Transformer)

T5 was introduced by Raffel et al. (2020) as a unified framework for all NLP tasks, where every problem is cast as a text-to-text task. T5 uses the transformer encoder-decoder architecture and is trained on a diverse corpus of tasks, making it versatile and highly effective. Its ability to generate text directly makes it well-suited for figurative language classification when trained to produce the correct label based on textual input (Raffel et al., 2020).

4-4) Evaluation Metrics Overview

A combination of traditional classification metrics and advanced semantic similarity metrics were employed to assess model performance.

4-4-1) Accuracy

Accuracy measures the proportion of correctly classified samples out of the total samples. It is a straightforward metric that indicates overall model performance but may not fully capture model behavior in imbalanced datasets (Sokolova et al., 2006).

4-4-2) Precision, Recall, and F1-Score

- **Precision** is the ratio of correctly predicted positive observations to the total predicted positives.
- **Recall** is the ratio of correctly predicted positives to all actual positives.
- **F1-Score** is the harmonic mean of precision and recall, balancing both metrics (Sasaki, 2007).

4-4-3) Confusion Matrix

A Confusion Matrix is a performance evaluation tool that provides a summary of classification outcomes on a dataset. It presents the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), offering a detailed view of model performance across different classes. In this study, the confusion matrix was used to analyze model errors for each figurative language type, highlighting which types were frequently misclassified and where the model struggled the most. This analysis is critical in understanding model behavior beyond aggregate metrics like accuracy and F1-score (Fawcett, T., 2006).

4-4-4) BLEURT (BERT-Based Learned Evaluation of Understudied Reference Texts)

BLEURT is a learned metric based on BERT embeddings, fine-tuned on human-evaluated data. It evaluates the semantic similarity between generated and reference text, making it ideal for assessing textual entailment and figurative language comprehension (Sellam et al., 2020).

4-4-5) BERTScore

BERTScore computes token-level similarity between predictions and references using BERT embeddings, providing a robust measure of semantic similarity (Zhang et al., 2020).

4-5) Environment Setup and Library Installation

The implementation and experimentation for this project were conducted using the “Google Colab” environment that supports “Python” and offers access to powerful GPUs. Specifically, an “A100 High-RAM GPU” was utilized to accelerate the training and evaluation of transformer-based models. The “A100 High-RAM GPU” significantly reduced model training time and ensured smooth data processing and model training without memory overflow issues.

Several Python libraries were imported and installed to facilitate the development of transformer-based large language models and ensure compatibility with state-of-the-art evaluation metrics. Hugging Face’s transformers and datasets libraries were used to access pre-trained language models and handle large datasets. The “torch” was installed to utilize the “PyTorch” deep learning framework, “evaluate” was included to compute various evaluation metrics, and the “tabulate” was included to format tables for descriptive analysis.

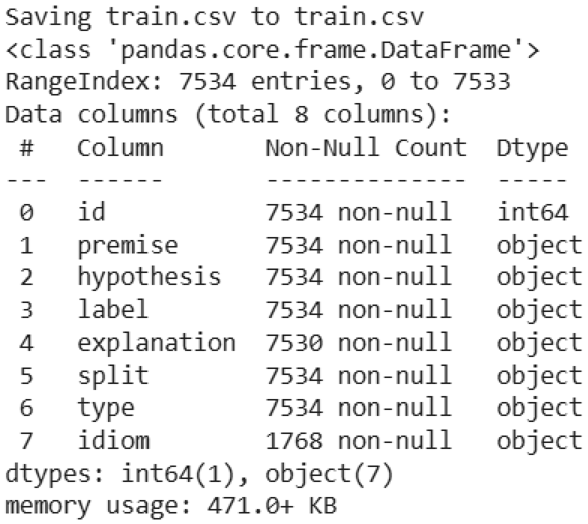
Environment configurations were adjusted to ensure efficient and deterministic model training. Specifically, the “WANDB_DISABLED” environment variable was set to "true" to disable automatic logging to Weights and Biases (wandb), and “PyTorch” anomaly detection was enabled to identify issues during backpropagation.

To maintain consistency across experiments, a random seed was set using a custom-defined “set_seed()” function. This function applied the same random seed to Python’s random module, “NumPy”, and “PyTorch”, including

GPU operations through “torch.cuda.manual_seed_all()”. Additionally, “PyTorch’s cudnn backend” was configured to disable automatic optimizations (torch.backends.cudnn.benchmark=False) and enforce deterministic behavior (torch.backends.cudnn.deterministic = True). By setting these parameters, the results obtained from the model remained consistent across multiple executions.

4-6) Dataset Acquisition and Preliminary Review

The FLUTE train set was uploaded into the environment via Google Colab’s file upload utility. It was then read using Pandas and stored as a DataFrame for further processing. Initial exploration of the dataset was performed using the info() and head() methods to gain insights into its structure, including column names, data types, and the presence of null values. This step was crucial for understanding the composition of the dataset and identifying potential issues such as incomplete records or incorrect data types.



```
Saving train.csv to train.csv
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7534 entries, 0 to 7533
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           7534 non-null   int64
 1   premise      7534 non-null   object
 2   hypothesis   7534 non-null   object
 3   label        7534 non-null   object
 4   explanation  7530 non-null   object
 5   split        7534 non-null   object
 6   type         7534 non-null   object
 7   idiom        1768 non-null   object
dtypes: int64(1), object(7)
memory usage: 471.0+ KB
```

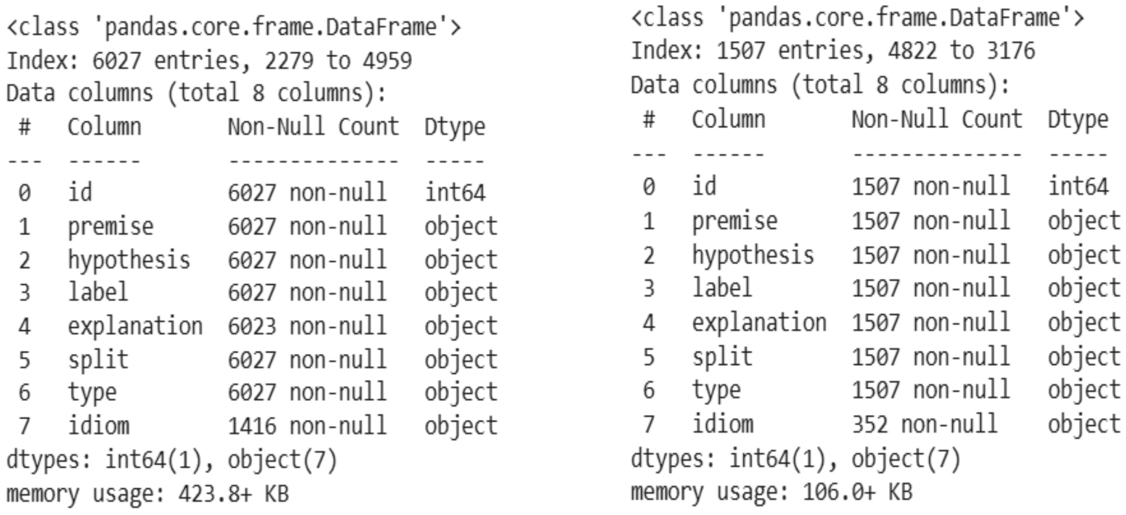
Figure 4-1: FLUTE Data Set Overview

4-6-1) Exploratory Data Analysis (EDA) of Key Columns Dataset Splitting for Model Training and Evaluation

To understand the distribution and characteristics of the dataset, an exploratory data analysis (EDA) procedure was conducted on specific columns: “type”, “label”, “premise”, “hypothesis”, and “explanation”. A custom function named “analyze_columns()” was developed to streamline this analysis. This function calculated and displayed the number of unique values and their respective frequencies for each specified column.

This analysis provided valuable insights into the dataset’s composition, such as the diversity of inference types and the balance of class labels. Such insights are essential for identifying class imbalances or data biases that could impact model performance. The results of this analysis informed subsequent preprocessing decisions, such as sampling techniques.

Then the dataset was partitioned into training and validation sets using the “train_test_split()” function from scikit-learn. An 80/20 split was applied, with 80% of the data allocated for training and 20% for validation. The random seed was set to 101 to ensure reproducibility. This division was critical for training the model on a portion of the data while using the unseen validation set to evaluate its performance and generalization capabilities.



```
<class 'pandas.core.frame.DataFrame'>
Index: 6027 entries, 2279 to 4959
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           6027 non-null   int64
 1   premise      6027 non-null   object
 2   hypothesis   6027 non-null   object
 3   label        6027 non-null   object
 4   explanation  6023 non-null   object
 5   split        6027 non-null   object
 6   type         6027 non-null   object
 7   idiom        1416 non-null   object
dtypes: int64(1), object(7)
memory usage: 423.8+ KB
```

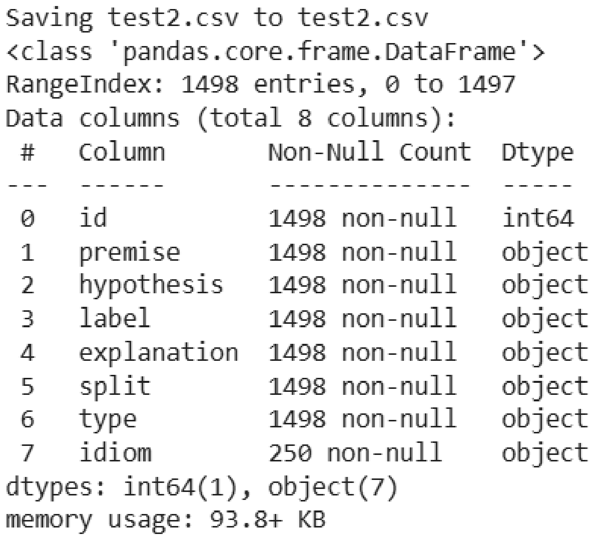
```
<class 'pandas.core.frame.DataFrame'>
Index: 1507 entries, 4822 to 3176
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           1507 non-null   int64
 1   premise      1507 non-null   object
 2   hypothesis   1507 non-null   object
 3   label        1507 non-null   object
 4   explanation  1507 non-null   object
 5   split        1507 non-null   object
 6   type         1507 non-null   object
 7   idiom        352 non-null    object
dtypes: int64(1), object(7)
memory usage: 106.0+ KB
```

Figure 4-2: FLUTE Training and Validation Sets Overview

Following the split, the training and validation datasets were independently analyzed to verify that they retained similar distributions of classes and types. The same “analyze_columns()” function was applied to both subsets, ensuring that any potential class imbalance was identified and could be addressed during model training.

4-6-2) Test Set Import and Review

The test set, provided separately, was uploaded and read into a Pandas data frame. Similar to the training and validation datasets, the test set underwent a preliminary review using info() and head() methods to check for completeness and structure. Additionally, the “analyze_columns()” function was employed to analyze the “type”, “label”, “premise”, “hypothesis”, and “explanation” columns. This ensured that the test set was appropriately formatted and that its class distribution aligned with the training and validation sets.



```
Saving test2.csv to test2.csv
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1498 entries, 0 to 1497
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   id           1498 non-null   int64 
 1   premise      1498 non-null   object
 2   hypothesis   1498 non-null   object
 3   label        1498 non-null   object
 4   explanation  1498 non-null   object
 5   split        1498 non-null   object
 6   type         1498 non-null   object
 7   idiom        250 non-null    object
dtypes: int64(1), object(7)
memory usage: 93.8+ KB
```

Figure 4-3: FLUTE Test Set Overview

4-7) Evaluating Figurative Language Understanding: Designing the Probing Task (Holmes' Approach)

This section presents the methodology employed to evaluate the ability of pre-trained transformer models to understand figurative language. The evaluation was conducted using contrastive pairs of figurative and literal expressions, a probing technique inspired by Holmes' approach (Waldis et al., 2024).

4-7-1) Pre-trained Models and Evaluation Framework for the Probing Task

Two transformer-based masked language models "RoBERTa (roberta-base)" and "BERT (bert-base-uncased)" were assessed on their ability to distinguish figurative from literal meanings across four figurative types: "sarcasm", "idiom", "simile", and "metaphor".

Both models were loaded using Hugging Face's "AutoTokenizer" and "AutoModelForMaskedLM" classes. The tokenizer segmented input sentences into subword tokens compatible with the models, while the masked language model (MLM) was used to compute token-level log probabilities necessary for log-likelihood estimation. The models were set to evaluation mode (model.eval()), disabling gradient computations and optimizing memory usage during inference.

The contrastive pairs were constructed from the FLUTE dataset, which provides both figurative sentences (hypothesis) and their literal explanations (explanation). The pairs were created as follows:

- **Figurative Sentence:** The hypothesis containing figurative expression.
- **Literal Sentence:** The explanation providing a literal interpretation.

The FLUTE dataset was then preprocessed using Pandas to ensure compatibility with Hugging Face models. Specifically:

- Missing values in the "hypothesis" and "explanation" columns were replaced with "N/A".

- All text fields were converted to string data types.
- The dataset was converted into a Hugging Face Dataset object for structured processing.

4-7-2) Sentence Log-Likelihood Estimation for the Probing Task

To evaluate model preferences for figurative or literal expressions, a custom function, “compute_sentence_likelihood()”, was implemented. This function calculated the log-likelihood of a sentence using the masked language model (MLM). The process involved the following steps:

1) **Tokenization:** The input sentence was tokenized.
2) **Model Inference:** The model’s output logits were obtained.
3) **Probability Computation:** Using softmax to convert logits into probabilities.

$$P(w_i) = \mathrm{softmax}(logits) = \frac{e^{logits_i}}{\sum_j e^{logits_j}}$$

4) **Log-Likelihood Calculation:** Summing the log probabilities for all tokens in the sentence.

$$\text{Log-Likelihood} = \sum_{i=1}^{n} \log(P(w_i))$$

The computed log-likelihood indicates how likely the model can find the sentence under its learned language distribution. A higher log-likelihood implies a better fit to the model’s internal representations.

4-7-3) Contrastive Pair Evaluation and Model Preference for the Probing Task

For each contrastive pair, the model's preference was determined by comparing log-likelihood scores for the figurative and literal sentences:

- **If Literal Score > Figurative Score:** The model preferred the literal interpretation.
- **If Figurative Score > Literal Score:** The model preferred the figurative interpretation.

The results for the first five pairs of each figurative type were displayed to provide qualitative insights into the model's interpretive tendencies.

4-7-4) Primary Evaluation Metrics Analysis for the Probing Task

To quantitatively assess model performance, three key metrics were computed:

1) **Accuracy:** Measures how often the model preferred the figurative sentence, assuming that figurative expressions are the intended meaning:

$$\text{Accuracy} = \frac{\text{Number of Figurative Preferences}}{\text{Total Pairs}}$$

2) **Mean Log-Likelihood Difference:** Reflects the average difference between the figurative and literal sentence scores. Positive values indicate a preference for figurative expressions:

$$\text{Mean Difference} = \frac{1}{N}\sum_{i=1}^{N}(\text{Figurative Score}_i - \text{Literal Score}_i)$$

3) **Confidence Score (Mean Absolute Difference):** Measures the average strength of the model’s preference, regardless of direction:

$$\text{Mean Confidence} = \frac{1}{N}\sum_{i=1}^{N} |\text{Figurative Score}_i - \text{Literal Score}_i|$$

4-7-5) Statistical Significance Tests for the Probing Task

To evaluate whether differences in model preferences were statistically significant, two statistical tests were conducted:

1) **Independent T-test:** Tests whether the mean log-likelihood scores for figurative and literal sentences are significantly different:
   - **Null Hypothesis:** No difference between figurative and literal scores.
   - **Alternative Hypothesis:** Significant difference between figurative and literal scores.

   A p-value < 0.05 indicates a statistically significant difference.

2) **Wilcoxon Signed-Rank Test:** A non-parametric test for paired samples, assessing the rank differences between figurative and literal scores:
   - **Null Hypothesis:** The distribution of differences is symmetric around zero.
   - **Alternative Hypothesis:** The distribution is not symmetric.

   A p-value < 0.05 indicates a significant difference in model preferences.

4-7-6) Model-Specific Evaluation Across Figurative Types for the Probing Task

The evaluation procedure was repeated for each figurative type (Sarcasm, Idiom, Simile, and Metaphor) for both models (roberta-base and bert-base-uncased). The results were stored in a structured dictionary (results_by_model) containing:

- Mean Log-Likelihood Difference
- Accuracy
- Mean Confidence
- Statistical Test Results (T-test and Wilcoxon)

This structured storage enabled the comparison of performance across models and figurative types.

4-8) Model Training and Evaluation Process for ROBERTA and BERT Models

This section presents the methodology employed to train and evaluate pre-trained language models for figurative language classification using the FLUTE dataset. Specifically, the models ROBERTA (roberta-base) and BERT (bert-base-uncased) were fine-tuned to classify sentences into the figurative language types. The evaluation included standard metrics such as accuracy, precision, recall, and F1-score, along with semantic similarity metrics BLEURT and BERTScore.

4-8-1) Conversion to Hugging Face Datasets and Label Encoding

The dataset was initially loaded as Pandas data frames for training (flute_train), validation (flute_val), and testing (test_df). To ensure compatibility with Hugging Face models and the Trainer API, these data frames were converted into “Hugging Face” dataset objects. Then the five figurative language types inside the training and validation sets were assigned numeric labels to facilitate supervised learning. (type_mapping = {"Sarcasm": 0, "Idiom": 1, "Simile": 2, "Metaphor": 3, "CreativeParaphrase": 4 }). This mapping ensured consistency during model training and evaluation.

4-8-2) Tokenization and Feature Extraction

The tokenizers for “roberta-base” and “bert-base-uncased” were loaded using Hugging Face’s AutoTokenizer. Each tokenizer segmented input sentences (premise and hypothesis) into subword tokens. A custom function was defined to tokenize input pairs and generate tokenized outputs (input_ids, attention_mask) with padding and truncation, and then the tokenization function was applied to all datasets, and numeric labels were assigned using the type_mapping dictionary. The same operations were performed for the validation and test datasets. Unnecessary columns were removed to retain only essential input fields (input_ids, attention_mask, labels).

4-8-3) ROBERTA and BERT Models’ Training

The “roberta-base” and “bert-base-uncased” models were initialized with classification heads for multi-class classification. The “Gradient checkpointing” was enabled to reduce memory consumption during backpropagation.

The model training was managed using Hugging Face’s Trainer API. The following training arguments were configured using TrainingArguments. Then the Trainer was initialized with the training and validation datasets, the model, the tokenizer, and an early stopping callback.

Hyperparameter	Value
Learning Rate	5e-5
Batch Size (Train/Eval)	16
Number of Epochs	5
Weight Decay	0.01
Evaluation Strategy	End of each epoch
Mixed Precision	fp16 (16-bit floating)
Early Stopping Patience	2 epochs

Table 4-1: Training Configuration Arguments for ROBERTA and BERT Models

The hyperparameters and training configurations were chosen based on best practices for fine-tuning large transformer models and ensuring efficient, stable convergence while minimizing overfitting. Each parameter was selected with consideration of the model architecture, dataset size, and available computational resources.

- **Learning Rate: 5e-5:** A learning rate of 5e-5 is a common default for fine-tuning large transformer models such as BERT and RoBERTa, as recommended by the Hugging Face and BERT research papers. This value ensures a balance between convergence speed and training stability, preventing overshooting the minima or slow convergence.
- **Batch Size: 16 (Train and Eval):** A batch size of 16 was selected to ensure a good balance between training speed and memory efficiency, especially on the A100 GPU with mixed precision (fp16). A batch size of 16 helps in maintaining gradient stability while enabling training with limited GPU memory.
- **Number of Epochs: 5:** The number of epochs was set to 5 to allow sufficient learning without overfitting. Combined with early stopping, the model stops training if performance on the validation set plateaus, ensuring that the model generalizes well.
- **Weight Decay: 0.01:** A weight decay of 0.01 was applied as an L2 regularization technique to prevent overfitting. This regularization forces the model to reduce complexity and generalize better, especially on the small FLUTE dataset.
- **Evaluation Strategy: End of Each Epoch:** Validation was performed after every epoch to monitor training progress and apply early stopping if the model’s performance stopped improving. This approach allows timely detection of overfitting and saves computation resources by preventing unnecessary epochs.
- **Mixed Precision Training (fp16): 16-bit Floating-Point Precision:** Mixed-precision training was enabled using fp16 (16-bit floating point) to leverage the hardware capabilities of the A100 High-RAM GPU, which is

optimized for mixed-precision operations. Reduces GPU memory consumption by nearly 50%, enabling larger batch sizes and faster training without compromising model accuracy.

- **Early Stopping: Patience of 2 Epochs:** Early stopping with patience of 2 epochs was implemented to halt training if validation performance did not improve for 2 consecutive epochs. This prevents overfitting and reduces training time without compromising performance.

In the initial phase of model training, the “CreativeParaphrase” label was intentionally retained within the training and validation datasets, despite its absence in the test set. This decision was driven primarily by the desire to leverage all available data to enhance the generalization capabilities of the transformer-based models. Figurative language classification is inherently complex and providing the model with a diverse range of linguistic examples, including creative paraphrases, enriched its exposure to various forms of semantic and syntactic variations. This additional label acted as a valuable source of training data, allowing the model to develop a more comprehensive understanding of language patterns, which could indirectly benefit its performance on the primary figurative language types of interest: sarcasm, idiom, simile, and metaphor.

Moreover, the inclusion of the “CreativeParaphrase” label functioned as an implicit form of regularization. By introducing an additional class during training, the model was prevented from overfitting to the limited set of figurative language types present in the dataset. Regularization is critical in deep learning, especially when dealing with relatively small datasets like FLUTE, as it encourages the model to learn more generalized features rather than memorizing specific examples. This broader learning during the initial training phase helped the model establish a solid foundation before the dataset was refined in the hyperparameter tuning stage.

4-8-4) Model Evaluation and Metrics

The trained models were evaluated on the test set using the "trainer.evaluate()" method. Predictions were generated, and labels were compared with ground truth values. The following performance metrics were computed:

1) **Accuracy:** Overall proportion of correct predictions.
2) **Precision, Recall, and F1-score:** Class-wise performance evaluation.
3) **Confusion Matrix:** Visualization of class-specific errors.
4) **BLEURT:** Semantic similarity between model predictions and references.
5) **BERTScore:** Token-level semantic similarity using BERT embeddings.

4-8-5) Misclassification Analysis for ROBERTA and BERT

To better understand model errors, misclassified examples were stored and analyzed to provide insight into which figurative types were most frequently confused by the models.

4-9) Error Patterns Analysis and Dataset Bias Inspection for ROBERTA and BERT Models

4-9-1) Error Patterns Analysis

Error pattern analysis is a crucial step in model evaluation, providing insights into how and why the models fail on certain predictions. This section outlines the methodology used to analyze the misclassified examples produced by the RoBERTa (roberta-base) and BERT (bert-base-uncased) models. Additionally, a bias inspection of the training dataset is performed to identify any imbalances that may contribute to systematic misclassification errors. The goal is to uncover patterns in errors, identify potential causes, and inform future model improvements.

Misclassified examples from both models were collected during the evaluation phase. The misclassified results were stored in dictionaries (misclassified_per_model), where each entry consisted of:

- **Index:** The sample's index in the test dataset.
- **Reference:** The original hypothesis (figurative sentence).
- **Prediction:** The predicted figurative type.
- **True Label:** The actual figurative type.
- **Predicted Label:** The model's prediction label.

The collected misclassifications were stored separately for ROBERTA and BERT, enabling comparative analysis, and a custom function, analyze_error_patterns(), was developed to process the collected misclassified examples for each model and summarize the types of errors made. The Error Pattern Analysis function performs the following steps:

1) Converts the misclassified examples into a Pandas data frame.
2) Maps numeric labels to their figurative type names.
3) Aggregates misclassifications by true and predicted labels.
4) Tags the results with the model's name.

For visualization of misclassification patterns, separate bar charts were plotted for ROBERTA and BERT models to analyze their error patterns individually.

4-9-2) Dataset Bias Inspection

Dataset imbalance is a common source of model bias, where underrepresented classes result in higher misclassification rates. To investigate this, the distribution of figurative types in the FLUTE training set was analyzed.

4-10) Models’ Fine-Tuning Process for ROBERTA and BERT Models

This section outlines the methodology used to fine-tune the ROBERTA (roberta-base) and BERT (bert-base-uncased) models on the FLUTE dataset. The process involves dataset refinement, hyperparameter tuning, and model evaluation. The fine-tuning was performed on the models by adjusting key parameters such as learning rate, batch size, and number of epochs.

The main goal of fine-tuning was to answer the research question: “Can fine-tuning on multi-type figurative datasets like FLUTE improve the performance of transformer-based large language models in understanding and classifying figurative language?”

4-10-1) Data Refinement and Preparation for Fine-Tuning

The “CreativeParaphrase” label was removed during the dataset refinement phase for fine-tuning to ensure alignment between the training and test sets. Since the test set did not contain this label, retaining it during fine-tuning could have misled the model and affected its ability to generalize to the target figurative language types. Eliminating the extra label focused the model’s learning on the four main figurative categories, enhancing its precision and performance on the final evaluation task.

Then the refined FLUTE dataset was split into training and validation sets using an 80/20 split with a fixed random seed for reproducibility. A numeric mapping was applied to the figurative types to prepare them for supervised learning at the next step. (Sarcasm: 0, Idiom: 1, Simile: 2, Metaphor: 3). Then the datasets were converted into Hugging Face dataset objects, enabling compatibility with the Hugging Face “Trainer API” and for the tokenization process, a custom tokenization function was created to tokenize the datasets using the Hugging Face tokenizer. Both “RoBERTa” and “BERT” use “WordPiece tokenization”, with padding and truncation applied to ensure consistent input lengths.

4-10-2) Fine-Tuning Configuration and Hyperparameters

The models were loaded from the Hugging Face Hub with pre-trained weights. A classification head was added to predict four figurative language types. The Key hyperparameters were adjusted based on experimentation to achieve optimal performance.

Hyperparameter	Value Before Fine-Tuning	Value After Fine-Tuning	Change Description
Learning Rate	5e-5	3e-5	Reduced to slow down learning for better convergence and stability.
Batch Size (Train/Eval)	16/16	8/16	Reduced training batch size to prevent memory overflow with Mixed-precision (fp16).
Number of Epochs	5	7	Increased to allow more iterations for learning.
Weight Decay	0.01	0.02	Increased to reduce overfitting with stronger regularization.
Evaluation Strategy	End of each epoch	End of Each Epoch	No change. Evaluation is still performed at the end of each epoch.
Mixed Precision	fp16	fp16	No change. Mixed-precision training retained.
Early Stopping Patience	2 epochs	Removed	Early stopping replaced with load_best_model_at_end = True.

Table 4-2: Comparison of Hyperparameters Before and After Fine-Tuning

The Lower learning rate (3e-5) improved convergence and reduced oscillations and made the training more stable. Adjusting batch size (8 for training) optimized memory use under fp16 and helped the efficiency of the training, and by increasing epochs from 5 to 7 and the weight decay from 0.01 to 0.02, the overfitting was reduced during the training process.

The hyperparameters were passed to the “TrainingArguments class” after tuning and a custom function (compute_metrics()) was defined to calculate accuracy during validation. Then the models were fine-tuned using the Hugging Face “Trainer” class with the prepared datasets and configurations.

4-10-3) Models’ Evaluation on Test Set After Fine-Tuning

Following fine-tuning, the trained models were evaluated on the test set using multiple metrics such as the training process.

1) **Accuracy:** Overall proportion of correct predictions.
2) **Precision, Recall, and F1-score:** Class-wise performance evaluation.
3) **Confusion Matrix:** Visualization of class-specific errors.
4) **BLEURT:** Semantic similarity between model predictions and references.
5) **BERTScore:** Token-level semantic similarity using BERT embeddings.

4-10-4) Misclassification Analysis After Fine-Tuning

Also, another Misclassification Analysis was done after fine-tuning the hyperparameters to understand model errors. Misclassified examples were stored and analyzed to provide insight into which figurative types were most frequently confused by the models.

4-11) Comparing Performance of ROBERTA and BERT Models Before and After Fine-Tuning

To assess the effectiveness of hyperparameter tuning, the performance before and after fine-tuning was compared and the percentage of improvement was calculated.

$$\text{Improvement } (\%) = \left(\frac{\text{After} - \text{Before}}{\text{Before}} \times 100 \right)$$

4-12) T5 Model Fine-Tuning and Evaluation Methodology

This section presents the methodology used to fine-tune the T5 (t5-base) model for the task of figurative language classification. The process involves preparing the dataset in a T5-compatible format, tokenizing the data, training the model with optimized hyperparameters, and evaluating its performance across accuracy, BLEURT, and BERTScore metrics.

The T5-base model and its tokenizer were loaded from the Hugging Face Hub and a label mapping was defined to convert figurative types into numeric labels and vice versa. Then each data sample was formatted into an input text and target text, compatible with T5's text-to-text paradigm:

- **Input:** Combined premise and hypothesis as a text string.
- **Target:** The correct figurative type as a text label.

Empty or corrupted samples were removed to ensure training data quality and the dataset was tokenized using the T5 tokenizer with a maximum input length of 128 tokens and a maximum target length of 16 tokens.

- **Padding and Truncation:** Ensured consistent input sizes for the transformer model.
- **Label Processing:** Padding tokens were replaced with -100 to be ignored during loss computation.

4-12-1) T5 Model Fine-Tuning Configuration and Training

The T5 model was fine-tuned using the Hugging Face “Seq2Seq Training Arguments” class. The hyperparameters were selected based on best practices for transformer fine-tuning, empirical experimentation, and the specific requirements of the figurative language classification task, and then the T5 model was trained using the “Seq2Seq Trainer API”.

Hyperparameter	Value
Learning Rate	2e-5
Batch Size (Train/Eval)	16
Number of Epochs	7
Weight Decay	0.02
Gradient Clipping	1
Gradient Accumulation	2 steps
Label Smoothing	0.1
Mixed Precision (fp16)	False

Table 4-3: Training Configuration Arguments for T5 Model

- **Learning Rate: 2e-5:** A lower learning rate was chosen to ensure stable training. T5 is a large transformer model, and a higher learning rate could cause the model weights to oscillate or diverge. Empirically, 2e-5 is a common choice for fine-tuning transformer-based models, as it strikes a balance between convergence speed and generalization performance.
- **Batch Size: 16 (Train and Eval):** The batch size was set to 16 to balance GPU memory usage and training efficiency. Given the use of an A100 GPU with high memory capacity, this batch size allowed for efficient parallel processing without running into memory overflow issues.
- **Number of Epochs: 7:** Training the model for 7 epochs provided sufficient learning cycles to allow the model to capture complex patterns in the figurative language dataset without overfitting. Initial trials with fewer

epochs showed suboptimal performance, while more epochs led to diminishing returns.

- **Weight Decay: 0.02:** Weight decay acts as a regularization term by penalizing large weights during training, thereby reducing the risk of overfitting. A value of 0.02 was selected through experimentation, as it provided better generalization on the validation set compared to the initial value of 0.01.
- **Gradient Clipping: 1:** Gradient clipping was set to 1.0 to prevent exploding gradients, a common issue in deep learning models with many parameters. This ensures that gradients are capped at 1.0 during backpropagation, leading to stable training even in complex scenarios.
- **Gradient Accumulation: 2 steps:** Gradient accumulation allowed the model to effectively use a larger batch size by accumulating gradients over 2 forward passes before updating the model weights. This was necessary to maintain training stability while utilizing the available GPU memory efficiently.
- **Label Smoothing: 0.1:** Label smoothing was used to prevent the model from becoming overconfident in its predictions by distributing some probability mass to incorrect classes. A smoothing factor of 0.1 ensures that the model assigns 90% of the probability to the correct class and distributes the remaining 10% among the other classes, thereby improving generalization.
- **Mixed Precision (fp16): False:** Mixed precision training (fp16) was disabled for the T5 model due to stability issues observed during initial trials. While fp16 generally improves training speed and reduces memory usage, it introduced numerical instability in T5's text-to-text architecture, leading to NaN losses. Therefore, fp32 precision was used for stable training.

4-12-2) Evaluation Metrics and Results for T5 Model

Such as the process for ROBERTA and BERT models, the trained T5 model was evaluated on the test set using multiple metrics.

1) **Accuracy:** Overall proportion of correct predictions.
2) **Precision, Recall, and F1-score:** Class-wise performance evaluation.
3) **Confusion Matrix:** Visualization of class-specific errors.
4) **BLEURT:** Semantic similarity between model predictions and references.
5) **BERTScore:** Token-level semantic similarity using BERT embeddings.

4-12-3) Misclassification Analysis for T5 Model

The Misclassification Analysis was done also for T5 model to understand the errors of the model. Misclassified examples were stored and analyzed to provide insight into which figurative types were most frequently confused by T5 model.

4-13) Comparing T5 with Tuned ROBERTA and BERT Models

In the last step, the T5 model's performance was compared to the fine-tuned ROBERTA (roberta-base) and BERT (bert-base-uncased) models to provide a comprehensive analysis of how well each model performed in the task of figurative language classification. This comparison was conducted across key evaluation metrics such as class-wise accuracy, precision, recall, and F1-score for each figurative type (Sarcasm, Idiom, Simile, and Metaphor). By analyzing these metrics side-by-side, the aim was to highlight the strengths and weaknesses of the masked language modeling and sequence classification strategies employed by ROBERTA and BERT in contrast to the T5 model's text-to-text generation approach.

5) Results

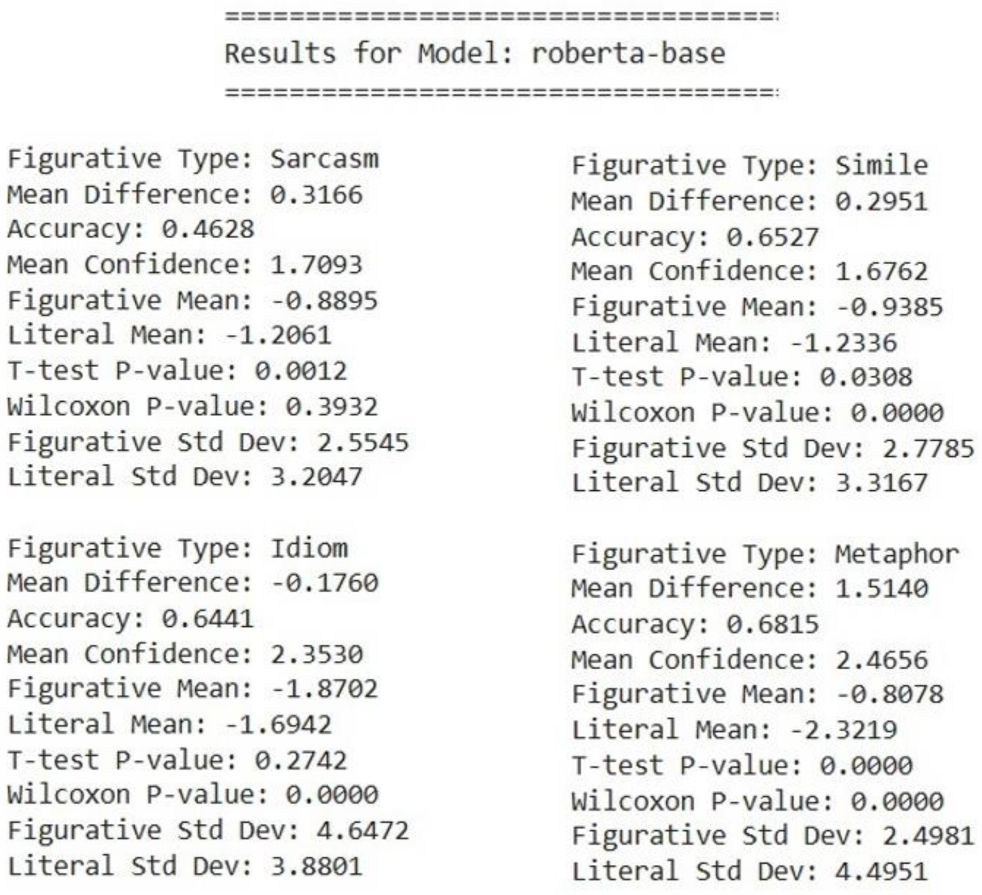
5-1) The Probing Task Results for ROBERTA and BERT Models

5-1-1) Metrics Clarification

This section presents the results obtained from the probing task designed to evaluate the ability of RoBERTa (roberta-base) and BERT (bert-base-uncased) to distinguish figurative language from literal expressions across four figurative language types: sarcasm, idiom, simile, and metaphor.The evaluation metrics include accuracy, mean log-likelihood difference, mean confidence, and statistical significance tests (t-test and Wilcoxon signed-rank test). For each figurative type, these metrics are analyzed:

- **Mean Difference:** Measures the difference between the Figurative Mean and the Literal Mean (Figurative Mean - Literal Mean) which are the average log-likelihood scores of figurative and literal sentences.
  (positive = preference for figurative, negative = preference for literal)
- **Accuracy:** Proportion of cases where the model prefers the figurative sentence over the literal sentence.
- **Mean Confidence:** Measures how confident the model is in distinguishing figurative from literal sentences. In this study, Mean Confidence was derived from absolute log-likelihood differences. The higher values mean stronger and more confident model preference.
- **T-test P-value:** Determines whether the difference between figurative and literal scores is statistically significant. A small p-value (< 0.05) indicates a significant preference.
- **Wilcoxon P-value:** A non-parametric test for the significance of paired samples. A small p-value (< 0.05) indicates a significant preference.
- **Standard Deviation:** For figurative and literal scores, this helps assess the variability in the model's predictions. A standard deviation close to 0 means the model’s predictions are very consistent and higher standard deviations indicate more variability in the model’s predictions.

5-1-2) The Probing Task Results for the ROBERTA Model



```
                    =====================================
                    Results for Model: roberta-base
                    =====================================

Figurative Type: Sarcasm                     Figurative Type: Simile
Mean Difference: 0.3166                      Mean Difference: 0.2951
Accuracy: 0.4628                             Accuracy: 0.6527
Mean Confidence: 1.7093                      Mean Confidence: 1.6762
Figurative Mean: -0.8895                     Figurative Mean: -0.9385
Literal Mean: -1.2061                        Literal Mean: -1.2336
T-test P-value: 0.0012                       T-test P-value: 0.0308
Wilcoxon P-value: 0.3932                     Wilcoxon P-value: 0.0000
Figurative Std Dev: 2.5545                   Figurative Std Dev: 2.7785
Literal Std Dev: 3.2047                      Literal Std Dev: 3.3167

Figurative Type: Idiom                       Figurative Type: Metaphor
Mean Difference: -0.1760                     Mean Difference: 1.5140
Accuracy: 0.6441                             Accuracy: 0.6815
Mean Confidence: 2.3530                      Mean Confidence: 2.4656
Figurative Mean: -1.8702                     Figurative Mean: -0.8078
Literal Mean: -1.6942                        Literal Mean: -2.3219
T-test P-value: 0.2742                       T-test P-value: 0.0000
Wilcoxon P-value: 0.0000                     Wilcoxon P-value: 0.0000
Figurative Std Dev: 4.6472                   Figurative Std Dev: 2.4981
Literal Std Dev: 3.8801                      Literal Std Dev: 4.4951
```

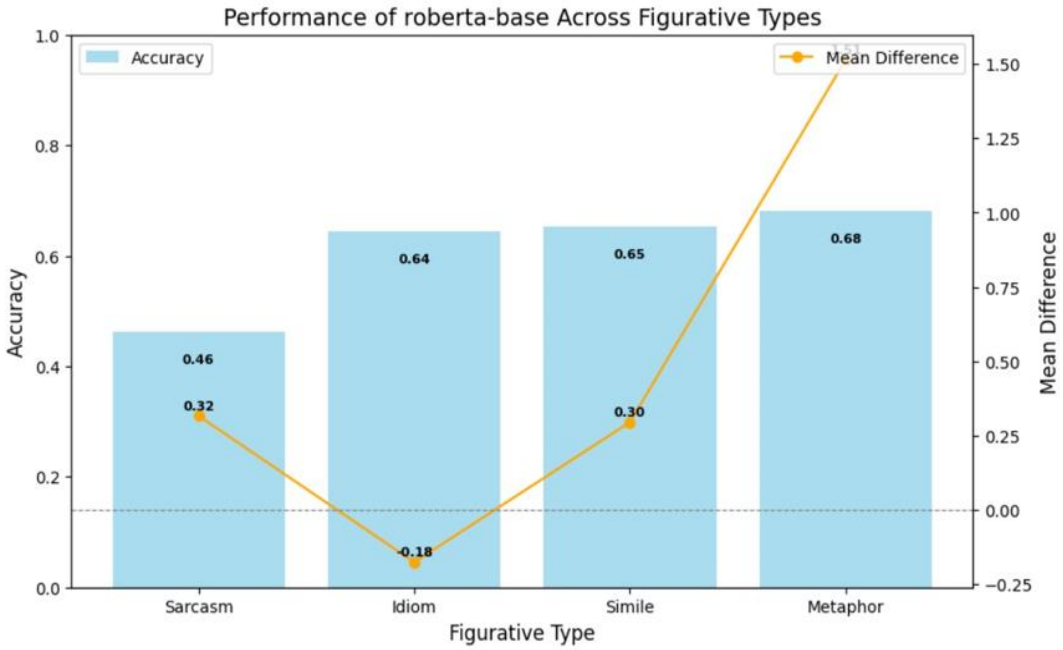


Figure 5-1: The Probing Task Results for the ROBERTA Model

- **Sarcasm:**

- **Mean Difference (0.3166):** A small positive value suggests a slight preference for figurative sarcasm over literal expressions, but this was weak.
- **Accuracy (0.4628):** Roberta-base's accuracy was relatively low (Below random guessing which is 50%), indicating difficulty in distinguishing sarcastic language from literal language. Sarcasm often relies on tone and context, which this model struggled with despite its contextual embeddings.
- **Mean Confidence (1.7093):** Low confidence further indicates uncertainty when dealing with sarcastic content.
- **T-test P-value (0.0012):** Statistically significant difference between figurative and literal scores, meaning the model differentiated between the two, albeit not reliably.
- **Wilcoxon P-value (0.3932):** Not statistically significant, indicating that while roberta-base exhibited slight preferences for figurative sarcasm, these preferences were not consistent enough to be considered reliable across all pairs.
- **Standard Deviations (Figurative: 2.5545, Literal: 3.2047):** High variability in predictions shows inconsistent performance across sarcastic examples.

- **Idiom:**

- **Mean Difference (-0.1760):** A slight negative value indicates a subtle preference for literal interpretations over idiomatic figurative language.
- **Accuracy (0.6441):** Higher accuracy compared to sarcasm, indicating that roberta-base was more adept at recognizing idiomatic expressions.
- **Mean Confidence (2.3530):** Moderate confidence suggests the model was somewhat assured in its classification of idiomatic text.

- **T-test P-value (0.2742):** No statistically significant difference between figurative and literal scores, indicating the model treated both types similarly.
- **Wilcoxon P-value (0.0000):** Highly significant, meaning roberta-base consistently differentiated between idiomatic expressions and their literal counterparts, despite a slight preference for literal language.
- **Standard Deviations (Figurative: 4.6472, Literal: 3.8801):** High variability, especially in figurative scores, highlights that while the model sometimes succeeded, it was inconsistent.

➢ **Simile:**

- **Mean Difference (0.2951):** Positive value shows a preference for figurative similes over literal alternatives.
- **Accuracy (0.6527):** The model performed well with similes, likely due to their structured format ("X is like Y"), making them easier to recognize.
- **Mean Confidence (1.6762):** Moderate confidence but lower than idioms, indicating the model was still cautious.
- **T-test P-value (0.0308):** Statistically significant, reinforcing that roberta-base distinguished similes from literal expressions.
- **Wilcoxon P-value (0.0000):** Statistically significant, reinforcing that roberta-base reliably distinguished similes from literal sentences across multiple instances.
- **Standard Deviations (Figurative: 2.7785, Literal: 3.3167):** Relatively lower variability compared to idioms and sarcasm suggests more consistent predictions.

➢ **Metaphor:**

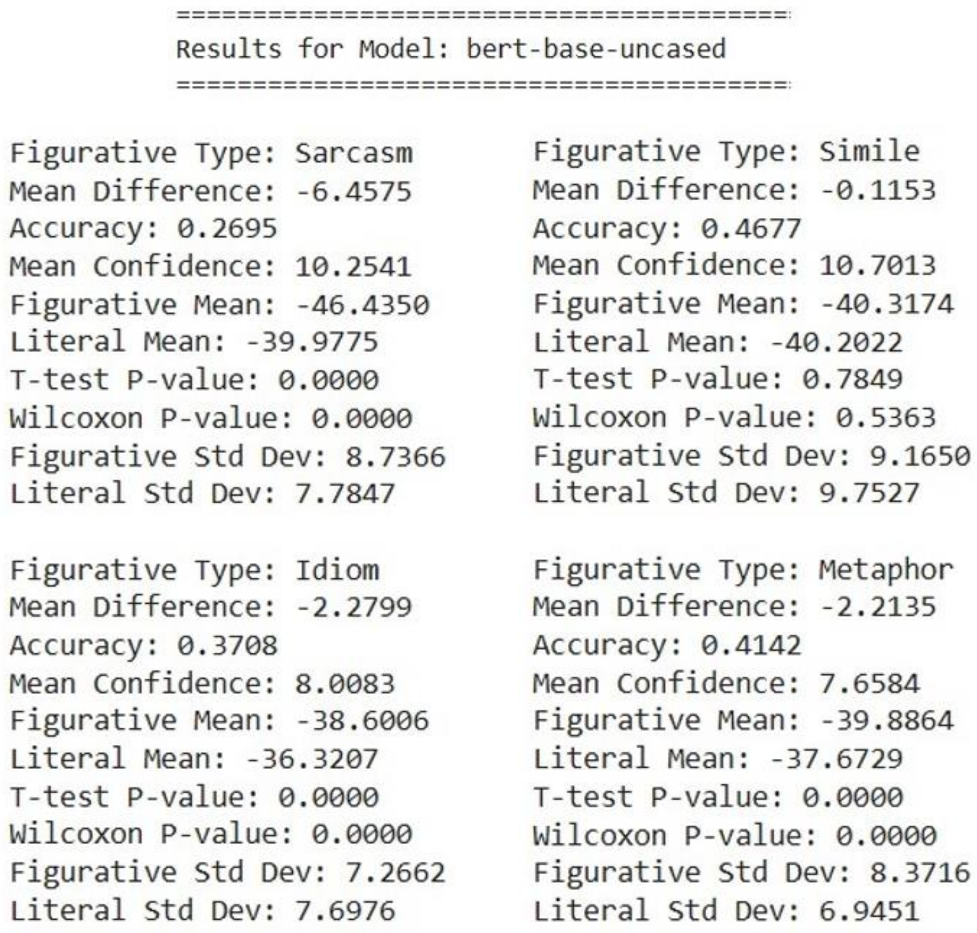
- **Mean Difference (1.5140):** A strong positive value indicates a clear preference for metaphorical language over literal text.
- **Accuracy (0.6815):** Best performance among all figurative types, showing roberta-base’s strength in metaphor comprehension.

- **Mean Confidence (2.4656):** Higher confidence compared to sarcasm and similes suggests the model felt more assured in handling metaphors.
- **T-test P-value (0.0000):** Statistically significant difference confirms that the model distinctly preferred metaphorical language.
- **Wilcoxon P-value (0.0000):** Strongly significant, confirming that roberta-base consistently favored metaphorical language over the literal, aligning with its high accuracy in this figurative type.
- **Standard Deviations (Figurative: 2.4981, Literal: 4.4951):** Lower variability in figurative scores points to consistent performance.

5-1-3) The Probing Task Results for the BERT Model

➢ **Sarcasm:**

- **Mean Difference (-6.4575):** A strong negative value indicates a clear preference for literal language over sarcastic expressions.
- **Accuracy (0.2695):** Extremely low, highlighting BERT's major struggle with sarcasm due to its lack of nuanced contextual understanding.
- **Mean Confidence (10.2541):** Surprisingly high confidence despite poor accuracy suggests BERT was confidently incorrect.
- **T-test P-value (0.0000):** Statistically significant difference confirms that the model treated sarcastic and literal text differently but failed in the accurate classification.
- **Wilcoxon P-value (0.0000):** Significant difference, but the preference is strongly skewed toward literal interpretations, reflecting BERT's difficulty with sarcastic text.
- **Standard Deviations (Figurative: 8.7366, Literal: 7.7847):** High variability shows inconsistency in sarcastic text handling.



```
                    ==========================================
                    Results for Model: bert-base-uncased
                    ==========================================

Figurative Type: Sarcasm          Figurative Type: Simile
Mean Difference: -6.4575          Mean Difference: -0.1153
Accuracy: 0.2695                  Accuracy: 0.4677
Mean Confidence: 10.2541          Mean Confidence: 10.7013
Figurative Mean: -46.4350         Figurative Mean: -40.3174
Literal Mean: -39.9775            Literal Mean: -40.2022
T-test P-value: 0.0000            T-test P-value: 0.7849
Wilcoxon P-value: 0.0000          Wilcoxon P-value: 0.5363
Figurative Std Dev: 8.7366        Figurative Std Dev: 9.1650
Literal Std Dev: 7.7847           Literal Std Dev: 9.7527

Figurative Type: Idiom            Figurative Type: Metaphor
Mean Difference: -2.2799          Mean Difference: -2.2135
Accuracy: 0.3708                  Accuracy: 0.4142
Mean Confidence: 8.0083           Mean Confidence: 7.6584
Figurative Mean: -38.6006         Figurative Mean: -39.8864
Literal Mean: -36.3207            Literal Mean: -37.6729
T-test P-value: 0.0000            T-test P-value: 0.0000
Wilcoxon P-value: 0.0000          Wilcoxon P-value: 0.0000
Figurative Std Dev: 7.2662        Figurative Std Dev: 8.3716
Literal Std Dev: 7.6976           Literal Std Dev: 6.9451
```

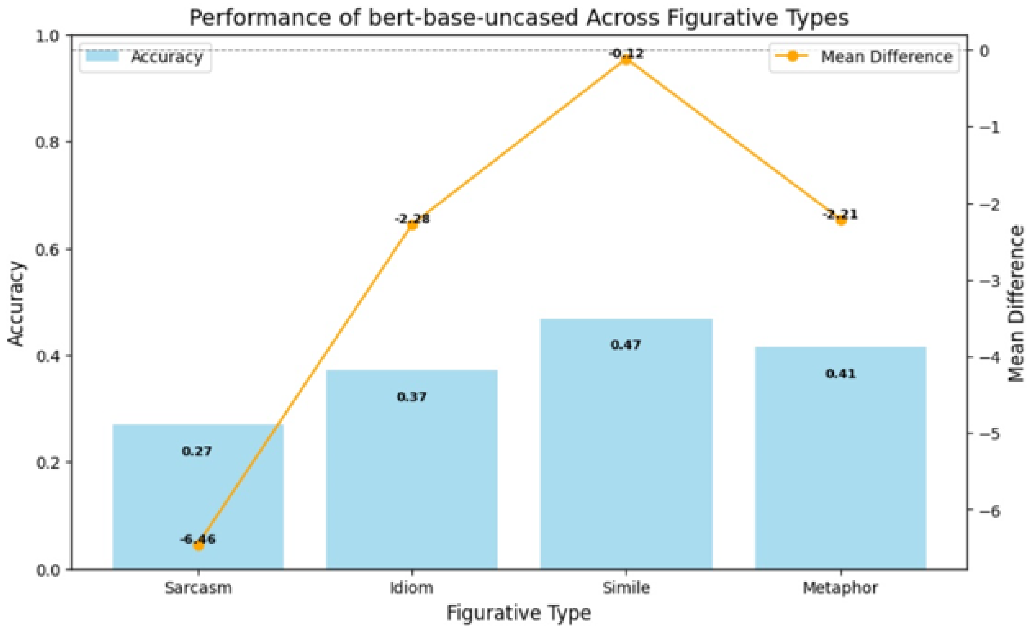


Figure 5-2: The Probing Task Results for the BERT Model

➢ **Idiom:**

- **Mean Difference (-2.2799):** A negative value shows a preference for literal text over idiomatic phrases.
- **Accuracy (0.3708):** Low accuracy reflects BERT's difficulty with idiomatic expressions, which often require contextual understanding.
- **Mean Confidence (8.0083):** High confidence, but this does not translate to accuracy.
- **T-test P-value (0.0000):** Significant difference, but again, BERT's literal bias prevailed.
- **Wilcoxon P-value (0.0000):** Statistically significant, indicating BERT treated idiomatic and literal text differently, though it consistently favored the literal.
- **Standard Deviations (Figurative: 7.2662, Literal: 7.6976):** High variability, indicating inconsistent performance.

➢ **Simile:**

- **Mean Difference (-0.1153):** A slight negative value indicates a marginal preference for literal language over similes.
- **Accuracy (0.4677):** Moderate but still lower than roberta-base, showing BERT's limited capability in recognizing similes.
- **Mean Confidence (10.7013):** Very high confidence, contrasting with low accuracy. That means the BERT was confidently wrong.
- **T-test P-value (0.7849):** No significant difference, indicating the model often treated the similes and the literal text similarly.
- **Wilcoxon P-value (0.5363):** Not statistically significant, showing that BERT's performance with similes was highly variable and inconsistent, despite its literal bias.
- **Standard Deviations (Figurative: 9.1650, Literal: 9.7527):** Extremely high variability highlights inconsistent predictions.

➢ **Metaphor:**

- **Mean Difference (-2.2135):** Negative value confirms a preference for literal interpretations over metaphorical language.
- **Accuracy (0.4142):** Low accuracy compared to roberta-base, reflecting BERT's struggle with metaphors.
- **Mean Confidence (7.6584):** High confidence despite poor performance.
- **T-test P-value (0.0000):** A significant difference, but the literal bias is evident.
- **Wilcoxon P-value (0.0000):** Significant, confirming that BERT consistently differentiated between metaphorical and literal language, albeit with a preference for literal text.
- **Standard Deviations (Figurative: 8.3716, Literal: 6.9451):** High variability in figurative scores suggests inconsistency.

5-1-4) The Probing Task Results Comparison

The comparison plot highlights that roberta-base outperformed bert-base-uncased across all figurative types during the probing task, with the most notable gaps in sarcasm (0.46 vs. 0.27) and idiom (0.64 vs. 0.37). This reinforced roberta-base's superior contextual embeddings, particularly in capturing complex language nuances, while bert-base-uncased struggled, especially with sarcasm and metaphor detection.

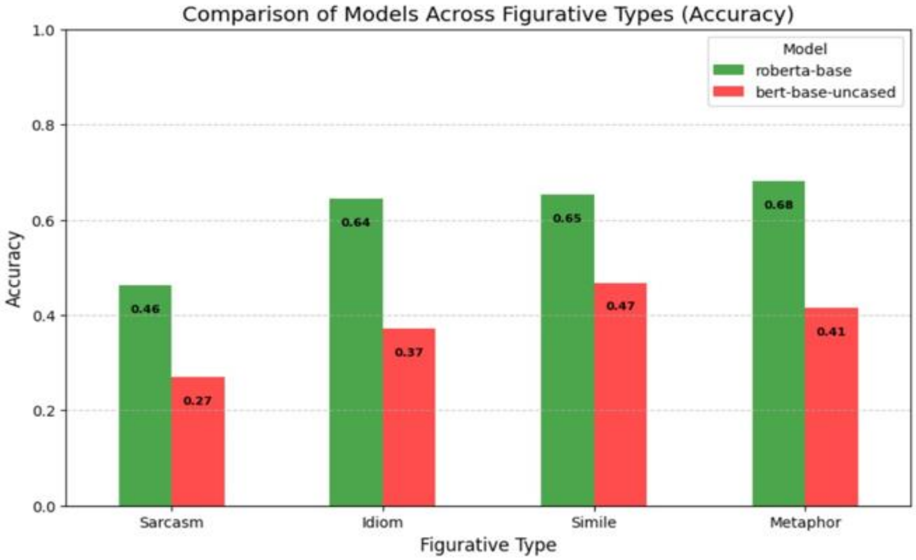


Figure 5-3: Accuracy Comparison of ROBERTA and BERT in the Probing Task

5-2) Training and Evaluation Results for the ROBERTA Model

5-2-1) Training and Validation Losses

The “Roberta-base” model was trained and evaluated on the figurative language classification task using the FLUTE dataset. The training process was performed with both training and validation losses recorded to monitor the model's performance.

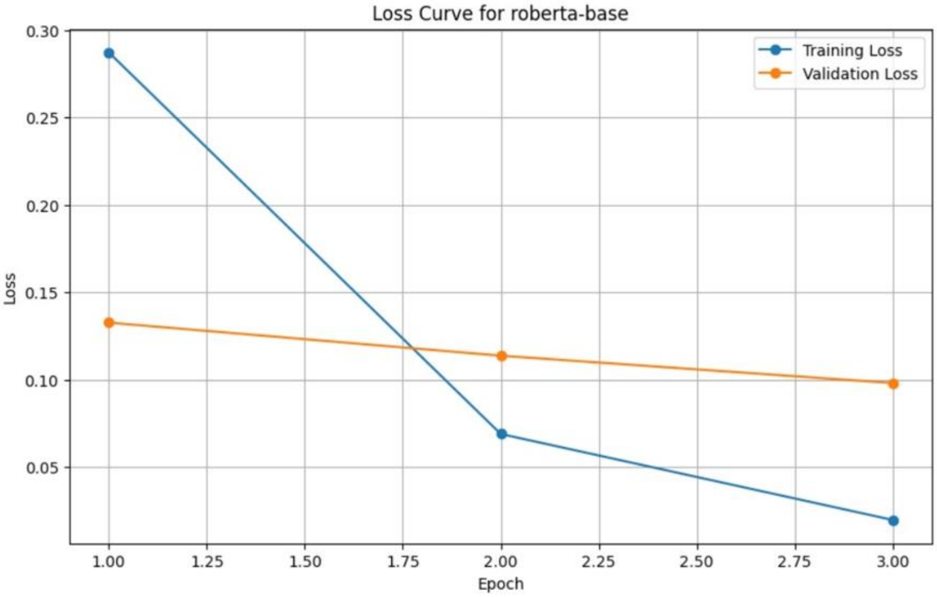


Figure 5-4: Training and Validation Loss Curve for ROBERTA Model

The training loss and validation loss decreased consistently across epochs, indicating that the model was learning effectively without overfitting. While the initial training loss was 0.2872, it decreased to 0.0197 by the final epoch. For the validation loss, the trend is the same. The initial validation loss started at 0.1326 and gradually reduced to 0.0870, showing a steady improvement in the model's ability to generalize to unseen validation data.

The loss curve plot reflects this trend, with the training loss declining more sharply than the validation loss, a common pattern in deep learning training. The proximity of training and validation losses at the end of training indicates that the model did not overfit and maintained good generalization.

5-2-2) Class-wise Accuracy

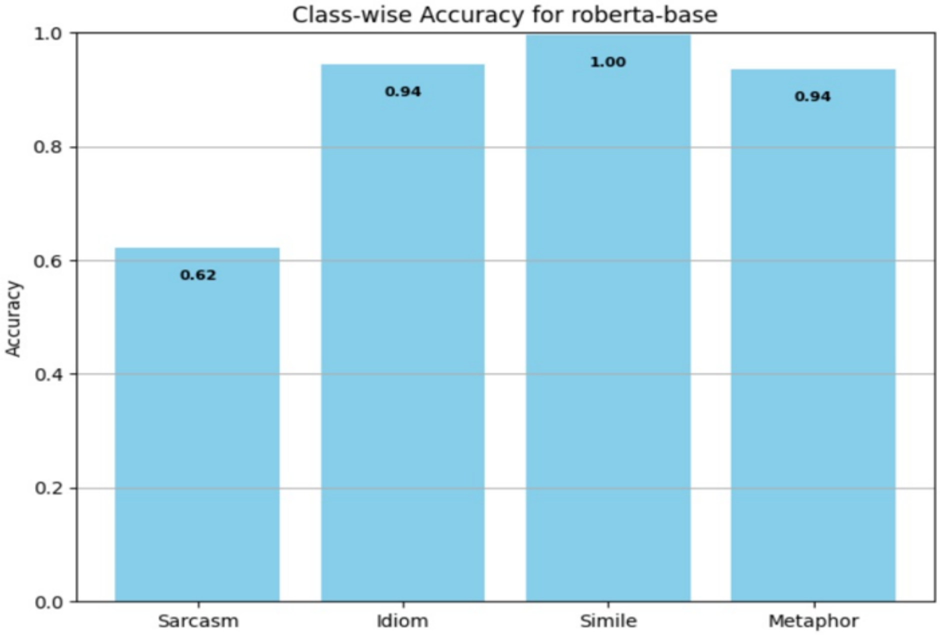


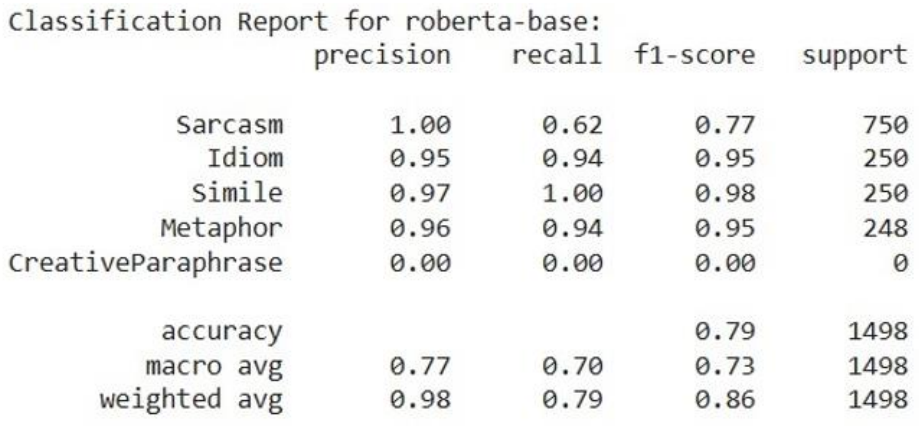
Figure 5-5: Class-wise Accuracy for ROBERTA Model

The ROBERTA model’s performance varied across different figurative language types:

- **Sarcasm:** Achieved the lowest accuracy at 0.62, indicating difficulty in recognizing sarcastic expressions, which often rely on implicit tone and context.
- **Idiom:** Performed well with an accuracy of 0.94, suggesting the model's ability to capture non-literal meanings within idiomatic phrases.
- **Simile:** Attained the highest accuracy of 1.00 (0.996 exactly), reflecting the model’s ease in identifying similes, likely due to their structured comparative format.
- **Metaphor:** Scored 0.94 in accuracy, demonstrating the model's capability to interpret abstract metaphorical language effectively.

The class-wise accuracy plot visually highlights this variation, with sarcasm lagging behind while similes stand out as the best-performing category.

5-2-3) Classification Report Analysis



```
Classification Report for roberta-base:
                    precision    recall  f1-score   support

           Sarcasm       1.00      0.62      0.77       750
             Idiom       0.95      0.94      0.95       250
            Simile       0.97      1.00      0.98       250
          Metaphor       0.96      0.94      0.95       248
CreativeParaphrase       0.00      0.00      0.00         0

          accuracy                           0.79      1498
         macro avg       0.77      0.70      0.73      1498
      weighted avg       0.98      0.79      0.86      1498
```

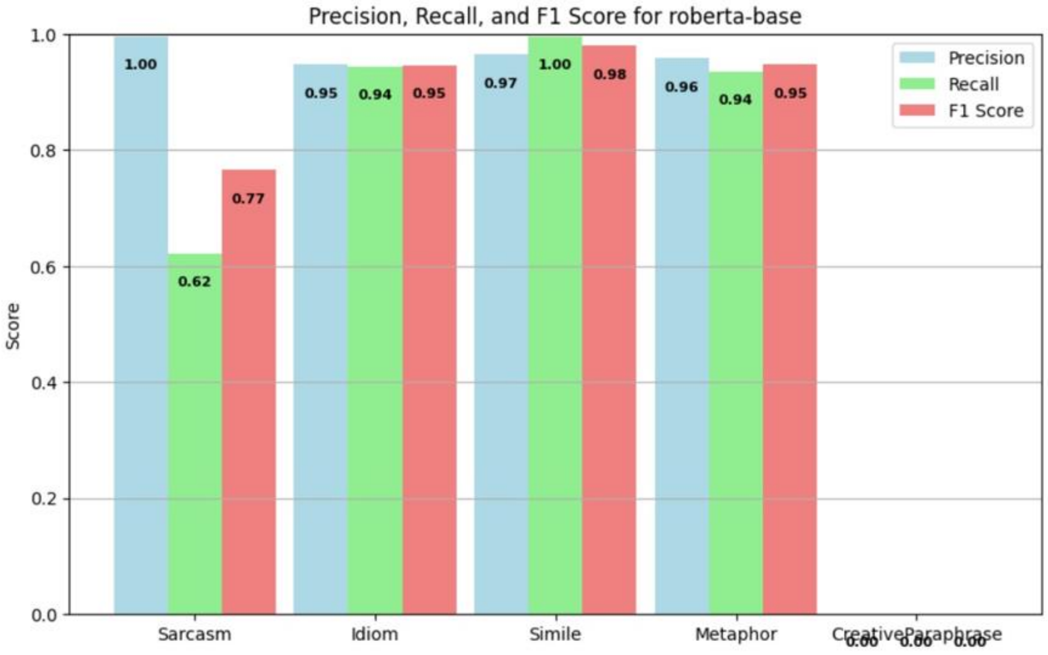


Figure 5-6: Classification Report Analysis for ROBERTA Model

The classification report for the Roberta-base model reveals the following performance metrics across different figurative language types:

- **Sarcasm:**

- **Precision: 1.00:** The model perfectly identified all sarcastic predictions it made, but this does not account for misclassified sarcastic examples.

- **Recall: 0.62:** The model struggled to recall sarcastic examples, indicating that a significant portion of sarcastic instances were misclassified.
- **F1-score: 0.77:** The harmonic mean of precision and recall shows that sarcasm detection was the weakest point, primarily due to poor recall despite perfect precision.

➢ **Idiom:**

- **Precision: 0.95:** High precision indicates that most idioms classified as idioms were indeed correct.
- **Recall: 0.94:** The model successfully identified most idioms, missing only a few instances.
- **F1-score: 0.95:** A strong overall performance in idiom detection due to both high precision and recall.

➢ **Simile:**

- **Precision: 0.97:** Nearly perfect precision in simile classification.
- **Recall: 1.00:** The model correctly identified all similes in the dataset.
- **F1-score: 0.98:** The highest overall F1-score, reflecting the ease with which the model recognized similes, likely due to their explicit comparative structure.

➢ **Metaphor:**

- **Precision: 0.96:** High precision indicates that the model rarely misclassified other categories as metaphor.
- **Recall: 0.94:** Slightly lower recall suggests a few metaphor instances were misclassified.
- **F1-score: 0.95:** Strong metaphor detection, with high precision and recall.

➢ **Creative Paraphrase:**

- **Precision, Recall, and F1-score: 0.00:** No instances of creative paraphrase in the test set.

➢ **Macro Average:**

- **Precision: 0.77:** The average precision across all classes, giving equal weight to each class regardless of its size.
- **Recall: 0.70:** The average recall across all classes was also equally weighted.
- **F1-score: 0.73:** The macro-average F1-score highlights the overall performance across classes, showing that while most classes performed well, the poor performance on sarcasm lowered the macro-average.

➢ **Weighted Average:**

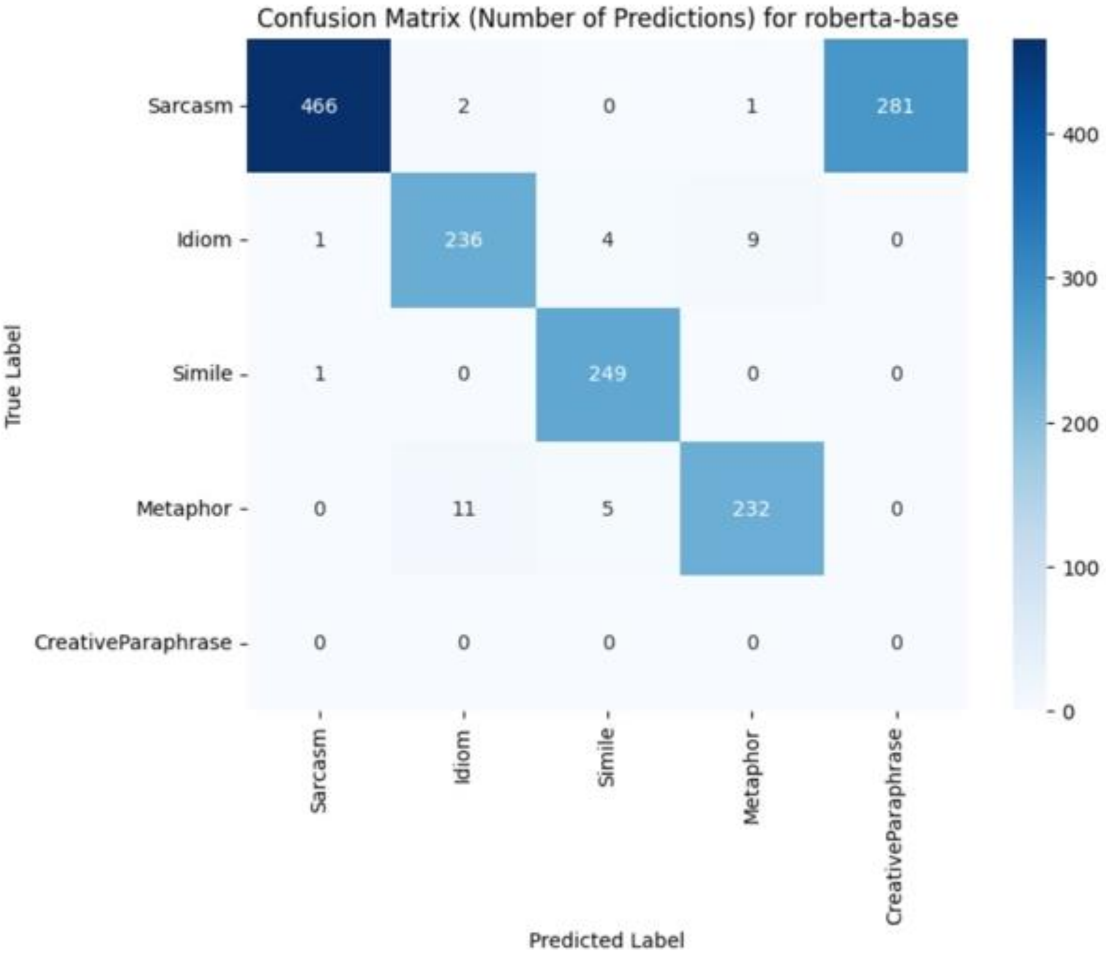
- **Precision: 0.98:** The precision averaged across all classes, weighted by the number of examples in each class. This high value reflects that most predictions in the largest classes (like sarcasm and idioms) were accurate.
- **Recall: 0.79:** The weighted recall was influenced by the model's ability to recall instances from all classes, and the lower recall for sarcasm significantly reduced this score.
- **F1-score: 0.86:** The weighted F1-score balances precision and recall across all classes based on the size of each class, showing a solid overall performance despite the model's struggles with sarcasm and the absence of creative paraphrase in the test set.

5-2-4) Confusion Matrix Analysis

The confusion matrix reveals the model's classification tendencies and errors:

- **Sarcasm Misclassification:** Out of 750 sarcastic examples, 281 were misclassified as CreativeParaphrase, showing a significant challenge in distinguishing sarcasm from paraphrased content.
- **Idiom Performance:** Most idiomatic expressions were correctly classified, with only a few misclassified as similes or metaphors.

- **Simile Perfect Score:** All simile instances were correctly classified (except 1 out of 250 cases), emphasizing the model's strong recognition capability for this category.
- **Metaphor Misclassification:** A few metaphors were confused with idioms and similes, but the majority were accurately classified.



Confusion Matrix (Number of Predictions) for roberta-base

True Label \ Predicted Label	Sarcasm	Idiom	Simile	Metaphor	CreativeParaphrase
Sarcasm	466	2	0	1	281
Idiom	1	236	4	9	0
Simile	1	0	249	0	0
Metaphor	0	11	5	232	0
CreativeParaphrase	0	0	0	0	0

Figure 5-7: Confusion Matrix Analysis for ROBERTA Model

5-2-5) Overall ROBERTA Model Performance

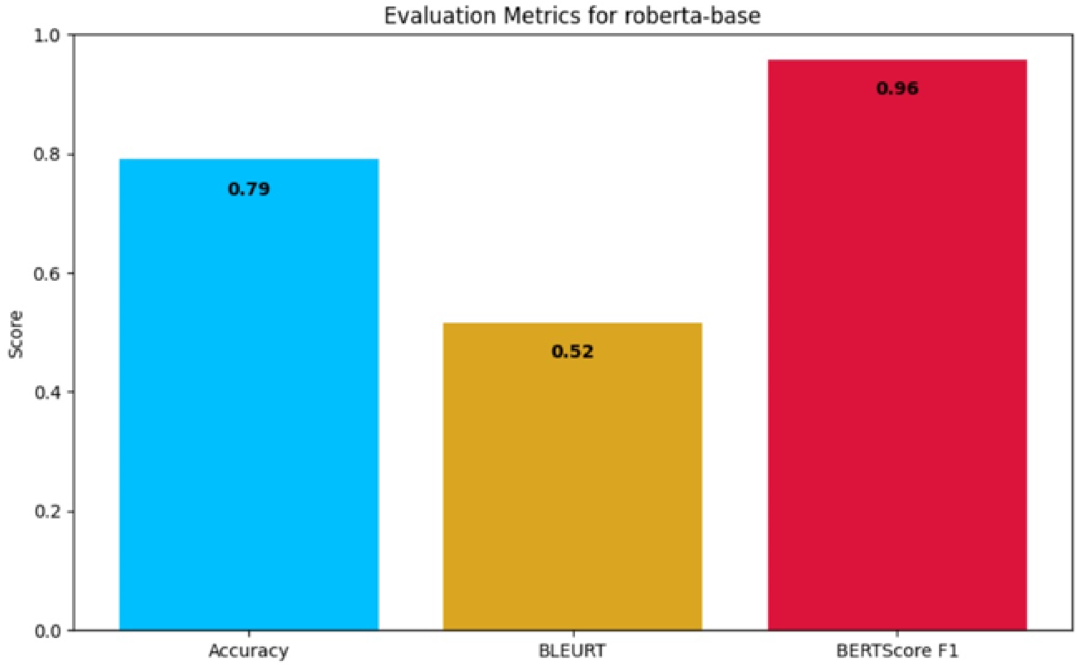


Figure 5-8: Evaluation Metrics Analysis for ROBERTA Model

The overall accuracy of the Roberta-base model on the test set was 0.79, demonstrating a solid performance across the classification task involving various types of figurative language. It indicates that nearly 80% of test samples were correctly classified. Additionally, the model achieved the “Average BLEURT Score” equal to 0.52, indicating moderate textual similarity between predicted and reference sentences. While the model captured some essential elements of the reference text, discrepancies in word choice, sentence structure, or nuanced semantic content reduced the overall similarity. The “Average BERTScore F1” was equal to 0.96 reflecting a strong high level of token-level semantic similarity between the predicted and reference sentences. This high score indicates that the model effectively captured the meaning and context of the figurative language, even if the exact phrasing differed.

5-3) Training and Evaluation Results for the BERT Model

5-3-1) Training and Validation Losses

The “BERT-base-uncased” model was trained and evaluated on the figurative language classification task using the FLUTE dataset. The training process was performed with both training and validation losses recorded to monitor the model's performance.

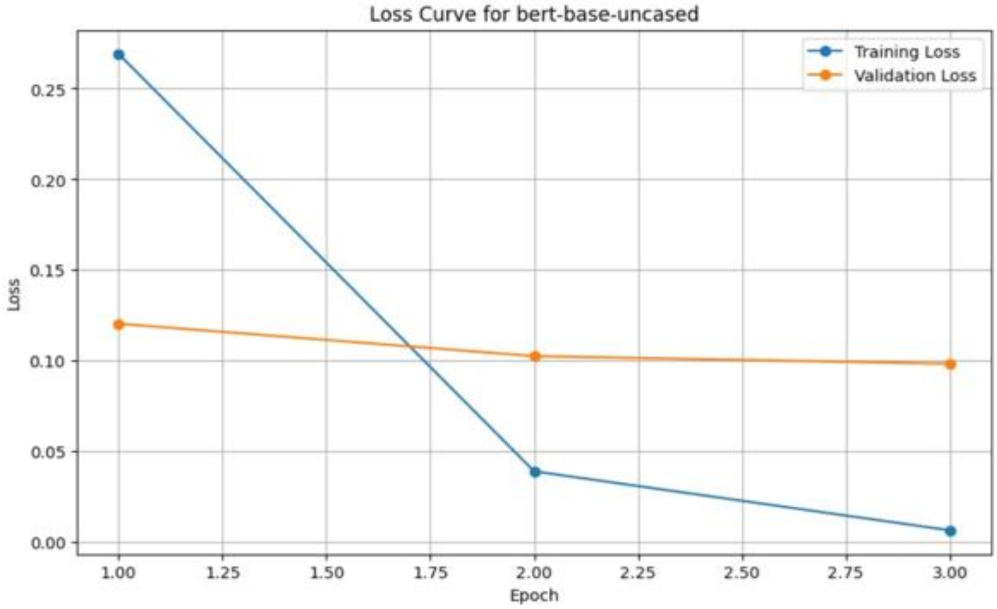


Figure 5-9: Training and Validation Loss Curve for BERT Model

The training loss and validation loss decreased consistently across epochs, indicating that the model was learning effectively without overfitting. While the initial training loss was 0.2692, it decreased to 0.0066 by the final epoch. For the validation loss, the trend is the same. The initial validation loss started at 0.1204 and gradually reduced to 0.0869, showing a steady improvement in the model's ability to generalize to unseen validation data.

The loss curve plot highlights a sharp decline in training loss, with validation loss steadily decreasing and remaining above training loss, suggesting that the model learned effectively without significant overfitting.

5-3-2) Class-wise Accuracy

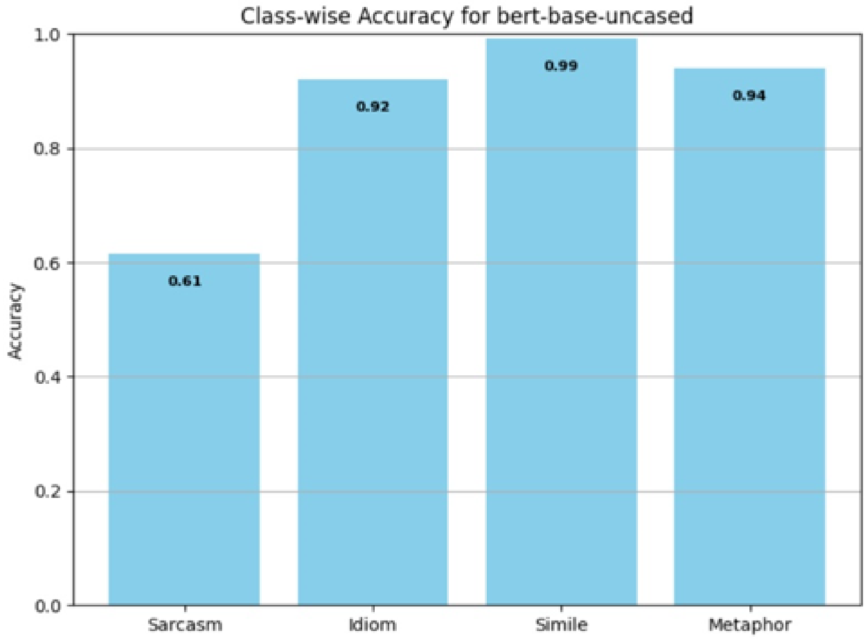


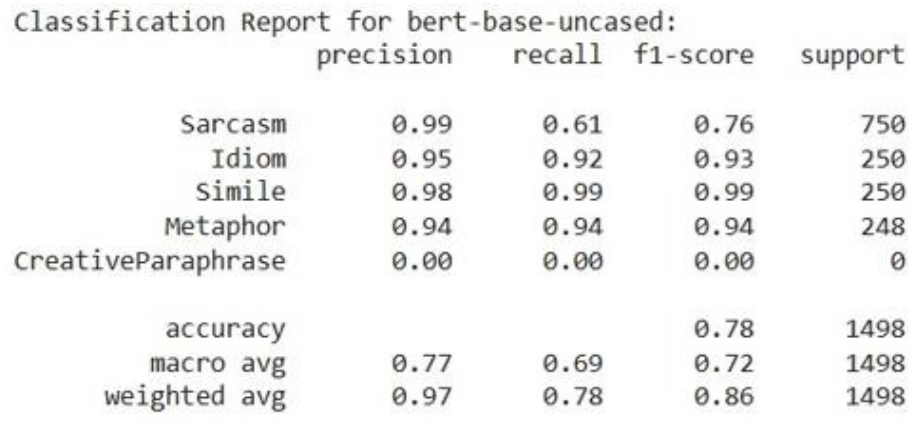
Figure 5-10: Class-wise Accuracy for BERT Model

The BERT model demonstrated varying levels of accuracy across different figurative language types:

- **Sarcasm:** Achieved the lowest accuracy at 0.61, indicating challenges in detecting sarcasm, which often depends on subtle contextual cues and implicit meaning.
- **Idiom:** Achieved an accuracy of 0.92, showing the model's relatively strong ability to recognize idiomatic expressions.
- **Simile:** Performed exceptionally well with an accuracy of 0.99, indicating that similes' explicit comparative structure was easily recognized.
- **Metaphor:** Scored 0.94, reflecting a strong capability to interpret metaphorical language, though slightly below the performance of similes.

The class-wise accuracy plot illustrates these differences, with sarcasm detection lagging while similes and idioms performed significantly better.

5-3-3) Classification Report Analysis



```
Classification Report for bert-base-uncased:
                    precision    recall  f1-score   support

           Sarcasm       0.99      0.61      0.76       750
             Idiom       0.95      0.92      0.93       250
            Simile       0.98      0.99      0.99       250
          Metaphor       0.94      0.94      0.94       248
CreativeParaphrase       0.00      0.00      0.00         0

          accuracy                           0.78      1498
         macro avg       0.77      0.69      0.72      1498
      weighted avg       0.97      0.78      0.86      1498
```

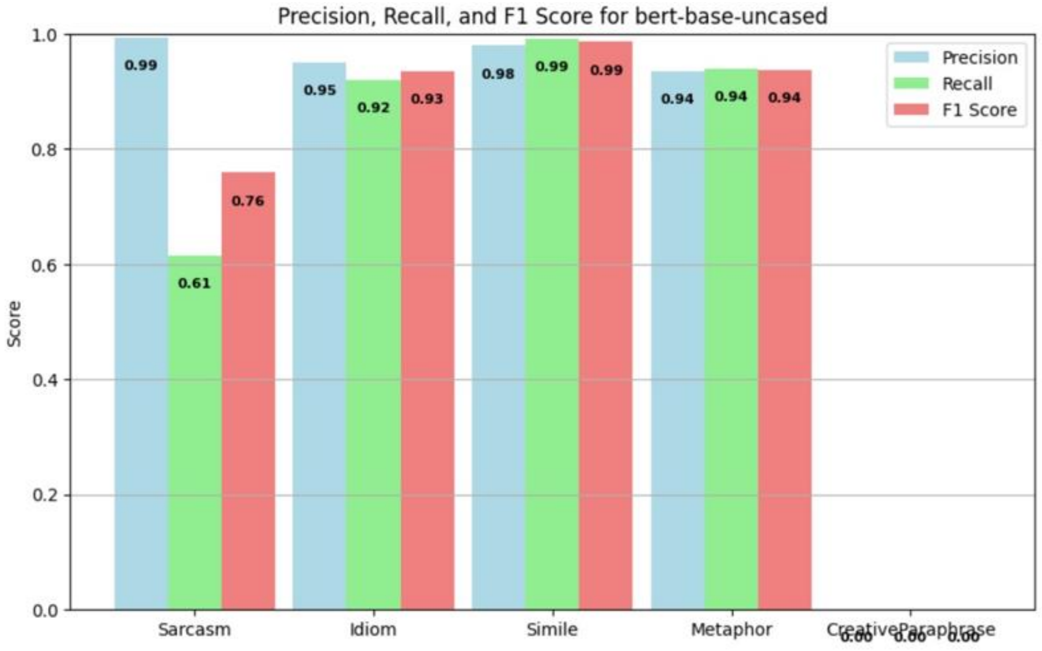


Figure 5-11: Classification Report Analysis for BERT Model

The classification report for the Roberta-base model reveals the following performance metrics across different figurative language types:

➢ **Sarcasm:**

- **Precision: 0.99:** High precision suggests that the model rarely misclassified other categories as sarcasm.
- **Recall: 0.61:** Low recall indicates that many sarcastic examples were missed.

- **F1-score: 0.76:** The imbalance between precision and recall highlights sarcasm detection as a challenge for the BERT-base.

➢ **Idiom:**

- **Precision: 0.95:** High precision, showing that most identified idioms were correct.
- **Recall: 0.92:** High recall indicates the model correctly identified most idiomatic examples.
- **F1-score: 0.93:** Strong overall performance for idioms due to the balance of precision and recall.

➢ **Simile:**

- **Precision: 0.98:** Almost perfect precision in simile classification.
- **Recall: 0.99:** Nearly all similes were correctly identified.
- **F1-score: 0.99:** The highest F1-score among all figurative types, reflecting the ease of detecting similes due to their explicit linguistic structure.

➢ **Metaphor:**

- **Precision: 0.94:** High precision indicates accurate identification of metaphorical language.
- **Recall: 0.94:** High recall shows that most metaphors were correctly classified.
- **F1-score: 0.94:** Balanced performance in recognizing metaphorical content.

➢ **Creative Paraphrase:**

- **Precision, Recall, and F1-score: 0.00:** No instances of creative paraphrase in the test set.

➢ **Macro Average:**

- **Precision: 0.77:** The average precision across all classes, giving equal weight to each class regardless of its size.
- **Recall: 0.69:** The average recall across all classes was also equally weighted.

- **F1-score: 0.72:** The macro-average F1-score highlights the overall performance across classes, showing that while most classes performed well, the poor performance on sarcasm lowered the macro-average.

➢ **Weighted Average:**

- **Precision: 0.97:** The precision averaged across all classes, weighted by the number of examples in each class. This high value reflects that most predictions in the largest classes (like sarcasm and idioms) were accurate.
- **Recall: 0.78:** The weighted recall was influenced by the model's ability to recall instances from all classes, and the lower recall for sarcasm significantly reduced this score.
- **F1-score: 0.86:** The weighted F1-score balances precision and recall across all classes based on the size of each class, showing a solid overall performance despite the model's struggles with sarcasm and the absence of creative paraphrase in the test set.

5-3-4) Confusion Matrix Analysis

The confusion matrix reveals the model's classification tendencies and errors:

- **Sarcasm Misclassification:** Out of 750 sarcastic examples, 288 were misclassified as CreativeParaphrase, showing a significant challenge in distinguishing sarcasm from paraphrased content.
- **Idiom Performance:** Most idiomatic expressions were correctly classified, with a few misclassified as sarcasm, similes or metaphors.
- **Simile Perfect Score:** All simile instances were correctly classified (except 2 out of 250 cases), emphasizing the model's strong recognition capability for this category.
- **Metaphor Misclassification:** A few metaphors were confused with idioms and similes, but the majority were accurately classified.

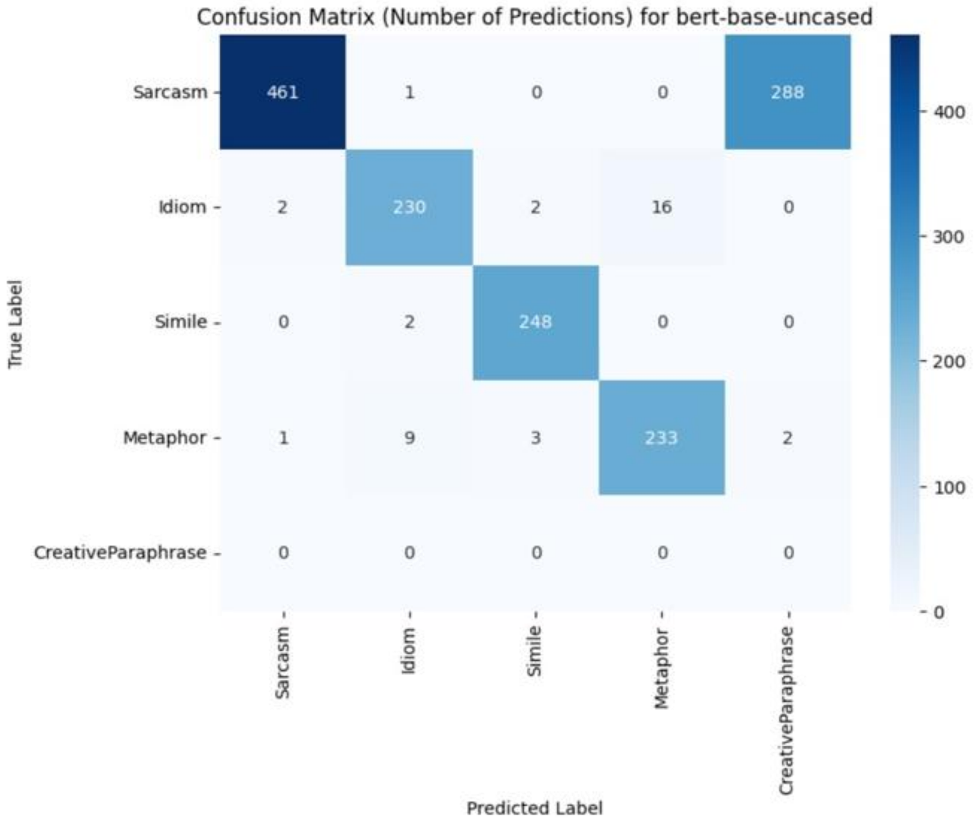


Figure 5-12: Confusion Matrix Analysis for BERT Model

5-3-5) Overall BERT Model Performance

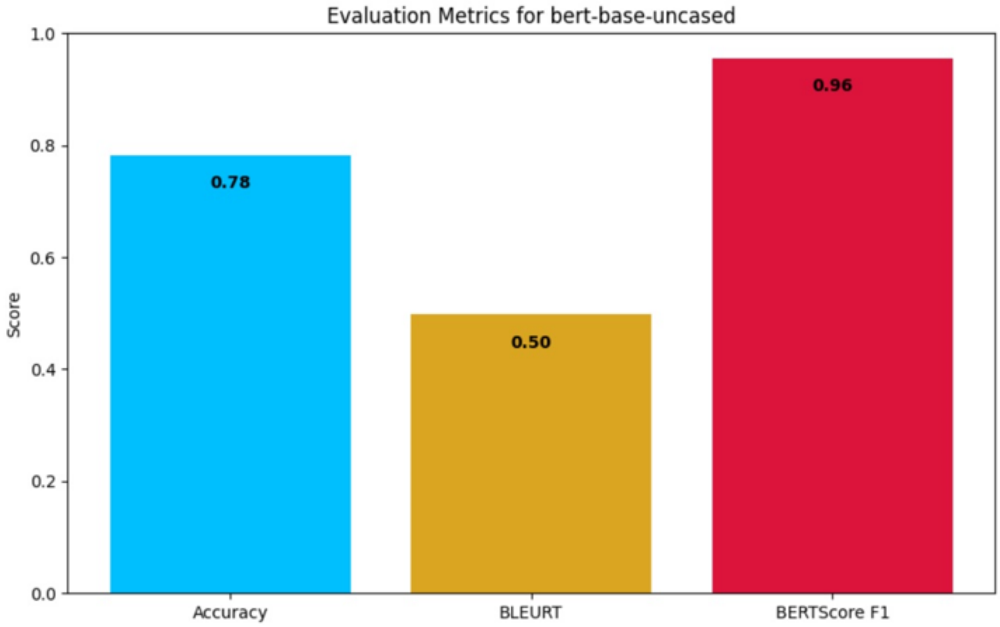


Figure 5-13: Evaluation Metrics Analysis for BERT Model

The overall accuracy of the bert-base-uncased model on the test set was 0.78, demonstrating a solid performance across the classification task involving various types of figurative language. It indicates that nearly 80% of test samples were correctly classified. Additionally, the model achieved the "Average BLEURT Score" equal to 0.50, indicating moderate textual similarity between predicted and reference sentences. While the model captured some essential elements of the reference text, discrepancies in word choice, sentence structure, or nuanced semantic content reduced the overall similarity. The "Average BERTScore F1" was equal to 0.96 reflecting a strong high level of token-level semantic similarity between the predicted and reference sentences. This high score indicates that the model effectively captured the meaning and context of the figurative language, even if the exact phrasing differed.

5-4) Comparison of Metrics Across Models

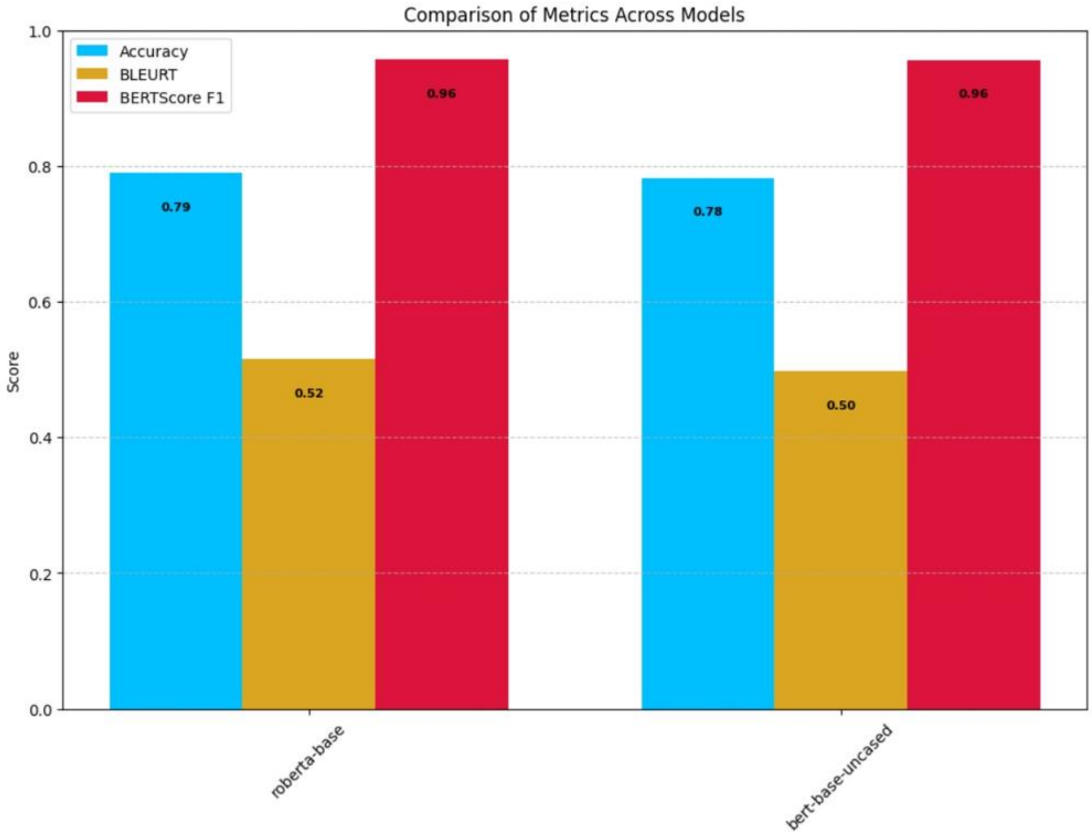
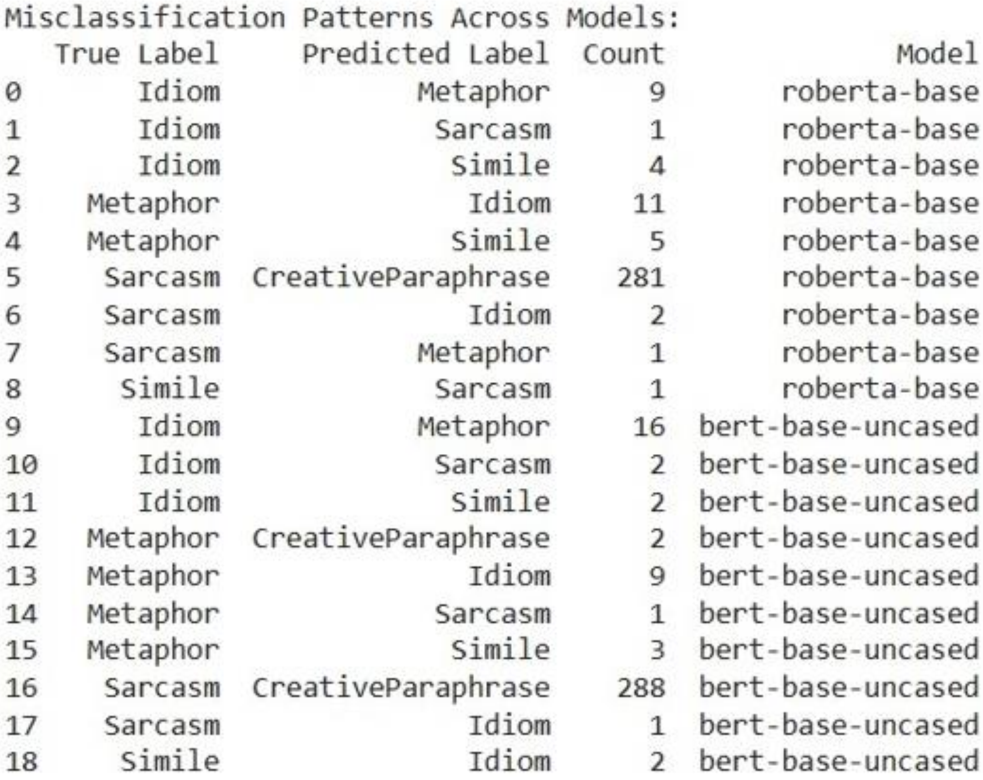


Figure 5-14: Comparison of Metrics Across Models

The performance comparison reveals that roberta-base slightly outperformed bert-base-uncased in both accuracy (0.79 vs. 0.78) and BLEURT score (0.52 vs. 0.50), reflecting its better textual alignment capabilities. However, both models achieved similar BERTScore F1 (0.96), indicating comparable performance in semantic similarity despite roberta-base’s marginal edge.

5-5) Error Patterns Analysis Results



```
Misclassification Patterns Across Models:
   True Label     Predicted Label  Count              Model
0       Idiom            Metaphor      9       roberta-base
1       Idiom             Sarcasm      1       roberta-base
2       Idiom              Simile      4       roberta-base
3    Metaphor               Idiom     11       roberta-base
4    Metaphor              Simile      5       roberta-base
5     Sarcasm  CreativeParaphrase    281       roberta-base
6     Sarcasm               Idiom      2       roberta-base
7     Sarcasm            Metaphor      1       roberta-base
8      Simile             Sarcasm      1       roberta-base
9       Idiom            Metaphor     16  bert-base-uncased
10      Idiom             Sarcasm      2  bert-base-uncased
11      Idiom              Simile      2  bert-base-uncased
12   Metaphor  CreativeParaphrase      2  bert-base-uncased
13   Metaphor               Idiom      9  bert-base-uncased
14   Metaphor             Sarcasm      1  bert-base-uncased
15   Metaphor              Simile      3  bert-base-uncased
16    Sarcasm  CreativeParaphrase    288  bert-base-uncased
17    Sarcasm               Idiom      1  bert-base-uncased
18     Simile               Idiom      2  bert-base-uncased
```

Figure 5-15: Misclassification Patterns Across the Models

The error patterns analysis for the Roberta-base and the BERT-base-uncased models provided valuable insights into how these models handled various figurative language types and where they tended to falter. By examining the misclassified examples, the patterns in the errors made by each model could be identified, and the strengths and weaknesses of their figurative language comprehension were highlighted.

5-5-1) Label Distribution in Training Set

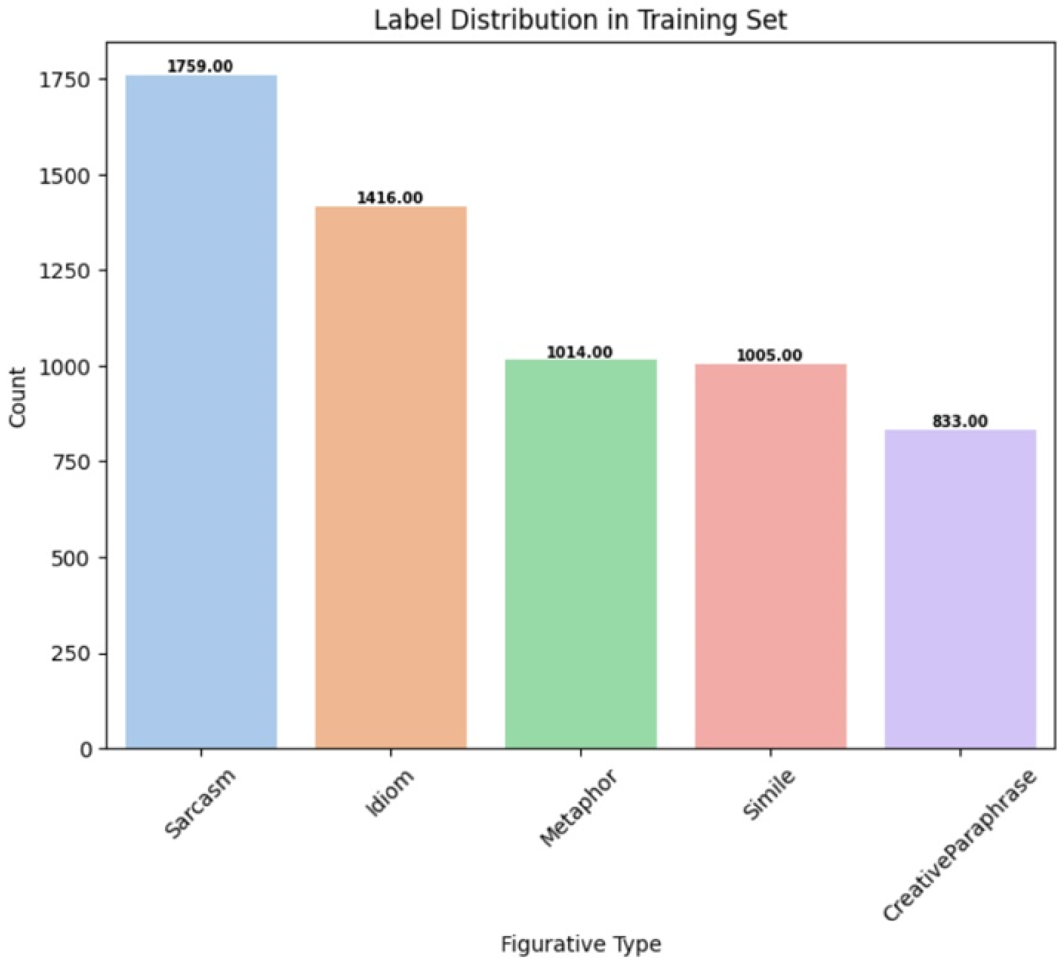


Figure 5-16: Label Distribution in Training Set

The label distribution plot in the training set revealed an imbalance across figurative language types:

- **Sarcasm** had the highest representation with 1759 examples, providing both models with more exposure to this type during training.
- **Idiom** followed Sarcasm with 1416 examples, ensuring the models have substantial training data for idiomatic expressions.
- **Metaphor** had moderate representations with 1014 examples.
- **Simile** had 1005 examples in the training set.

- **Creative Paraphrase** had the lowest count at 833 examples, indicating a potential challenge for both models due to limited training exposure.

This distribution imbalance influenced the models' performance, particularly their difficulties with Creative Paraphrase, as seen in the classification results. The higher presence of sarcasm in the training set explained why both models performed better at precision for sarcasm but still struggled with recall due to its nuanced nature.

5-5-2) Roberta Model’s Misclassification Patterns

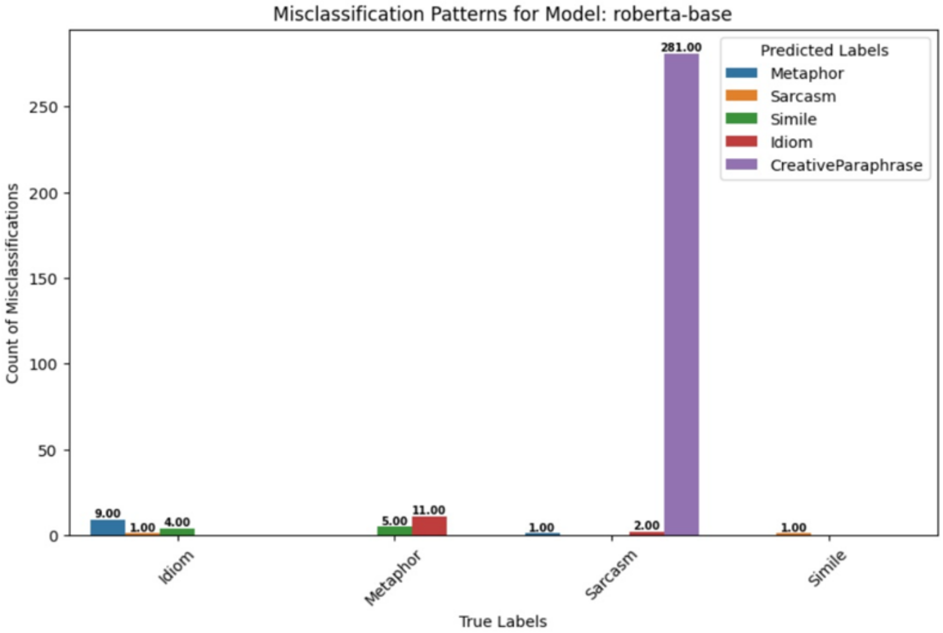


Figure 5-17: Misclassification Patterns for ROBERTA Model

- **Sarcasm → Creative Paraphrase:** 281 errors (most common), indicating that almost all sarcasm-related errors involved confusion with paraphrased content.
- **Idiom → Metaphor:** 9 errors, highlighting that almost 64% of idiom errors were misclassified as metaphor.
- **Metaphor → Idiom:** 11 errors, comprising almost 69% of metaphor errors, reflecting the conceptual similarity between these types.

- **Minimal Simile errors:** Only 1 instance of simile misclassification, confirming a high accuracy rate for similes.

5-5-3) Bert Model's Misclassification Patterns

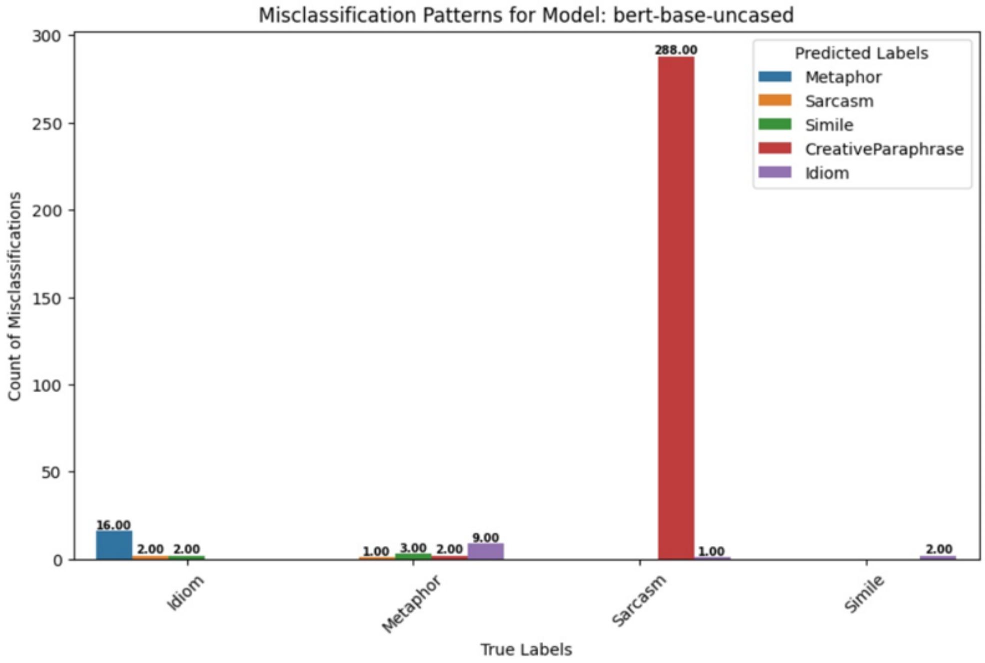


Figure 5-18: Misclassification Patterns for BERT Model

- **Sarcasm → Creative Paraphrase:** 288 errors (most common), indicating that almost all sarcasm-related errors involved confusion with paraphrased content.
- **Idiom → Metaphor:** 16 errors, highlighting that 80% of idiom errors were misclassified as metaphor.
- **Metaphor → Idiom:** 9 errors, comprising 60% of metaphor errors, reflecting the conceptual similarity between these types.
- **Minimal Simile errors:** Only 2 instances of simile misclassification, confirming a high accuracy rate for similes.

5-6) Fine-Tuning Results for ROBERTA Model

5-6-1) Training / Validation Losses and Validation Accuracy Progression

The fine-tuning process of the ROBERTA model aimed to enhance its performance in figurative language classification by optimizing hyperparameters such as learning rate, batch size, number of training epochs, weight decay, and mixed precision (fp16). The model was trained over 7 epochs, with the training and validation performance monitored throughout.

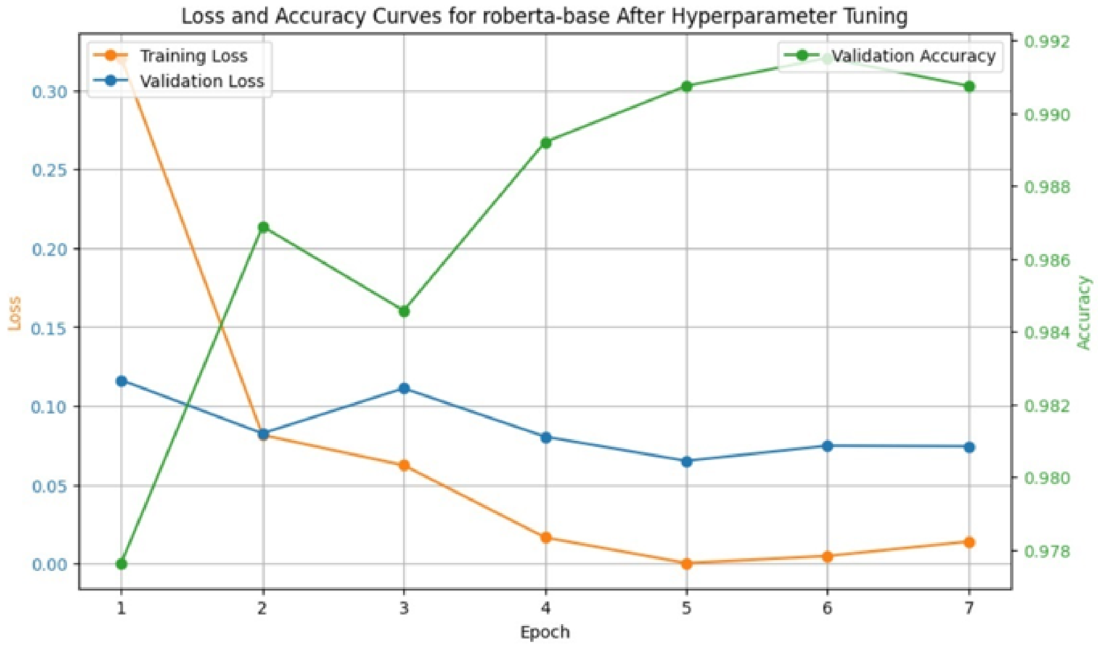


Figure 5-19: Loss and Accuracy Curves for ROBERTA Model After Hyperparameter Tuning

The initial epoch showed high training loss, as expected, while validation loss was significantly lower, indicating the model's initial grasp of the data. Then the training loss dropped sharply, but the validation loss increased slightly, suggesting that the model was still adjusting its parameters. Between the third and the seventh epoch, the training loss continued to decrease steadily, reaching 0.0001 by epoch 7, while the validation loss stabilized around 0.0744, indicating that the model successfully learned without overfitting.

The loss curve plot showed a rapid decrease in training loss, with validation loss showing minor fluctuations but stabilizing, demonstrating the model's consistent learning and generalization capacity after fine-tuning.

The validation accuracy improved consistently across epochs. The accuracy curve showed a steady increase, peaking at over 99% accuracy, reflecting the positive impact of fine-tuning on the model's performance.

5-6-2) Class-wise Accuracy

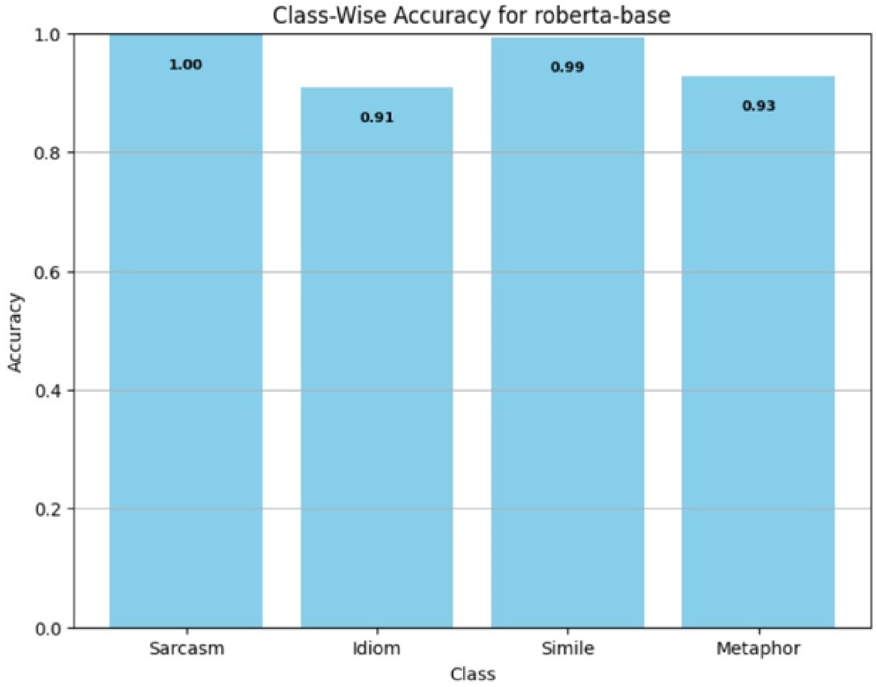


Figure 5-20: Class-wise Accuracy for ROBERTA Model After Hyperparameter Tuning

The class-wise accuracy for each figurative type for the ROBERTA model after fine-tuning is:

- **Sarcasm: 0.9973 (Almost 1.00):** Significant improvement, showing the model's enhanced ability to detect sarcasm, likely due to its nuanced tone being better captured through fine-tuning.

- **Idiom: 0.91:** Decreased from pre-tuning levels, suggesting that the model’s focus on sarcasm might have come at the cost of idiomatic comprehension.
- **Simile: 0.99:** Still high but lower than the pre-tuning perfect score, indicating that while fine-tuning helped the overall performance, some patterns for simile recognition may have been overshadowed.
- **Metaphor: 0.93:** Slightly lower than pre-tuning accuracy, possibly due to the model prioritizing certain linguistic cues during training.

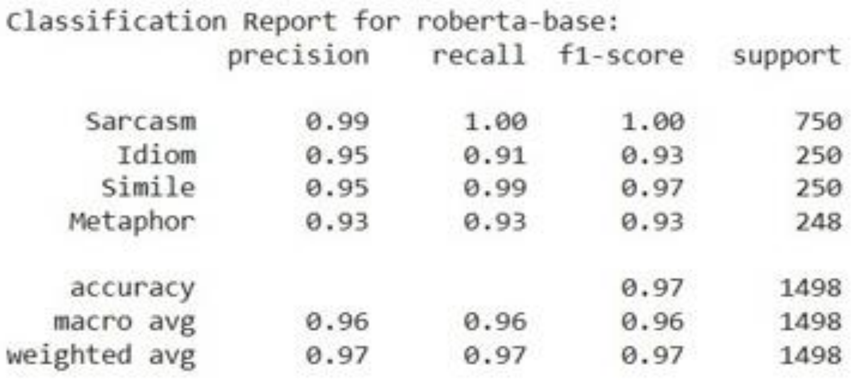
The fine-tuning process aimed to enhance the Roberta-base model’s ability to classify figurative language. However, while the overall performance improved, the class-wise accuracy declined for all categories except Sarcasm, indicating a shift in the model's performance post-tuning.

5-6-3) Classification Report Analysis

The classification report for the Roberta-base model reveals the following performance metrics across different figurative language types:

- **Sarcasm (Precision: 0.99, Recall: 1.00, F1-score: 1.00):** Sarcasm detection improved significantly due to fine-tuned contextual embeddings, likely because sarcasm was the most represented figurative type in the training set.
- **Idiom (Precision: 0.95, Recall: 0.91, F1-score: 0.93):** Although these values were high, they were lower than pre-tuning results for recall and F1-score, suggesting that the model’s focus on sarcasm during fine-tuning may have caused a decline in its ability to capture the nuanced cultural and contextual elements essential for idiom recognition.
- **Simile (Precision: 0.95, Recall: 0.99, F1-score: 0.97):** While still strong, these scores reflected a slight decline compared to the pre-tuning performance.

- **Metaphor (Precision: 0.93, Recall: 0.93, F1-score: 0.93):** The scores were lower than before fine-tuning, indicating that while the model improved in sarcasm detection, it struggled to maintain its understanding of abstract metaphorical relationships, potentially due to the shift in its learning priorities during fine-tuning.



```
Classification Report for roberta-base:
              precision    recall  f1-score   support

     Sarcasm       0.99      1.00      1.00       750
       Idiom       0.95      0.91      0.93       250
      Simile       0.95      0.99      0.97       250
    Metaphor       0.93      0.93      0.93       248

    accuracy                           0.97      1498
   macro avg       0.96      0.96      0.96      1498
weighted avg       0.97      0.97      0.97      1498
```

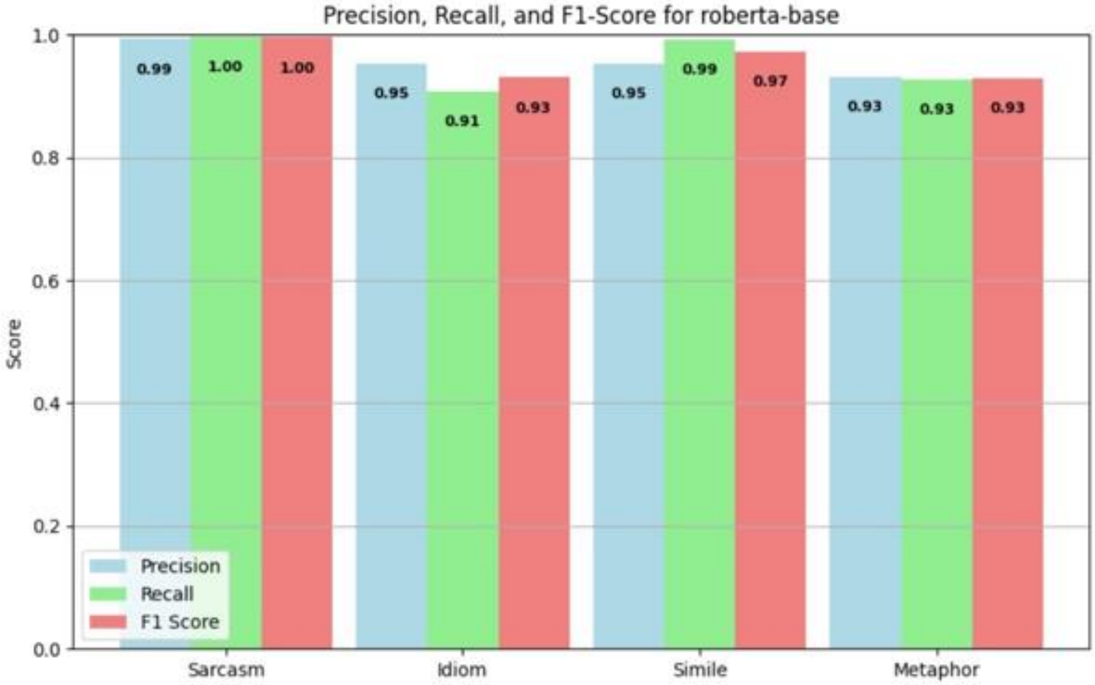


Figure 5-21: Classification Report Analysis for ROBERTA Model After Hyperparameter Tuning

- **Macro Average (Precision: 0.96, Recall: 0.96, F1-score: 0.96):** Indicates that, on average, the model was highly precise across all classes and correctly identified 96% of the figurative language examples across all classes.
- **Weighted Average (Precision: 0.97, Recall: 0.97, F1-score: 0.97):** Higher than the macro average values due to the significant number of sarcasm samples, which the model handled exceptionally well after fine-tuning. The higher weight of the sarcasm class boosted the overall precision and recall despite drops in the other figurative languages. The F1-score confirmed that the model performed consistently well across all classes, but it was notably influenced by sarcasm's high performance and larger sample size.

5-6-4) Confusion Matrix Analysis

The confusion matrix reveals the model's classification tendencies and errors:

- **Sarcasm:** 748 correctly predicted, with only 2 misclassifications.
- **Idiom:** 227 correct predictions, with 16 misclassifications as metaphors, 6 as simile and 1 as sarcasm.
- **Simile:** Perfect classification with 248 correct predictions and only 2 misclassifications with idiom.
- **Metaphor:** 230 correct predictions, with minimal misclassifications into sarcasm, idiom and simile.

Fine-tuning reduced misclassification rates significantly, particularly for sarcasm and metaphors, which were previously challenging categories for the model.

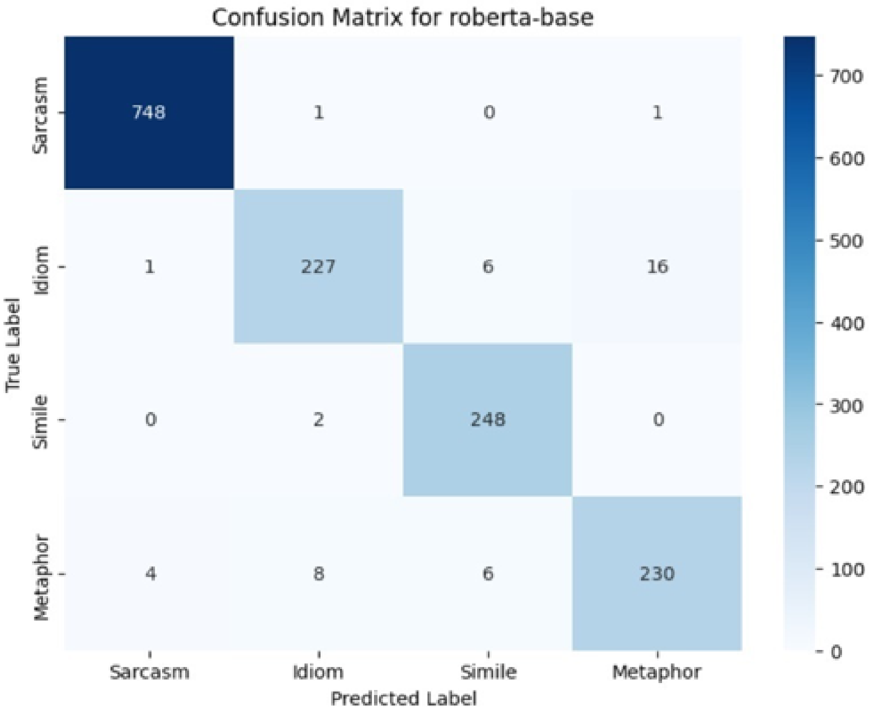


Figure 5-22: Confusion Matrix Analysis for ROBERTA Model After Hyperparameter Tuning

5-6-5) Overall ROBERTA Model Performance

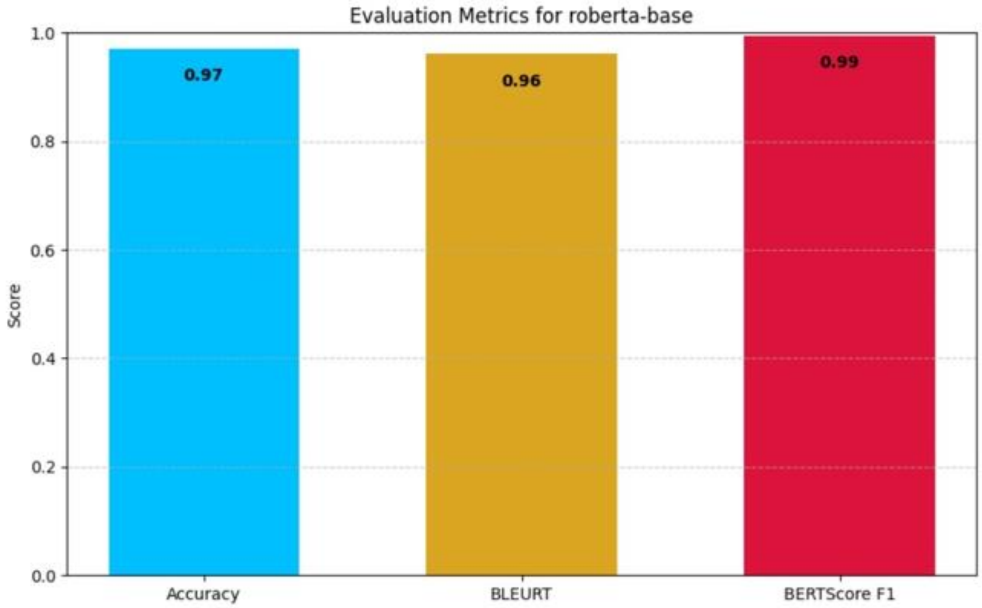


Figure 5-23: Evaluation Metrics Analysis for ROBERTA Model After Hyperparameter Tuning

The overall accuracy of the Roberta-base model on the test set was 0.97, which significantly improved from its pre-tuning value of 0.79. Additionally, the model achieved the “Average BLEURT Score” equal to 0.96, Highlighting a high degree of textual similarity between predictions and reference sentences. The “Average BERTScore F1” was equal to 0.99 Indicating near-perfect token-level semantic similarity. The bar plot of evaluation metrics for ROBERTA model after hyperparameter tuning showed all three metrics approached 1.0, indicating the success of fine-tuning in enhancing the model’s performance across different evaluation criteria.

5-7) Fine-Tuning Results for BERT Model

5-7-1) Training / Validation Losses and Validation Accuracy Progression

The fine-tuning process of the BERT model aimed to enhance its performance in figurative language classification by optimizing hyperparameters such as learning rate, batch size, number of training epochs, weight decay, and mixed precision (fp16). The model was trained over 7 epochs, with the training and validation performance monitored throughout.

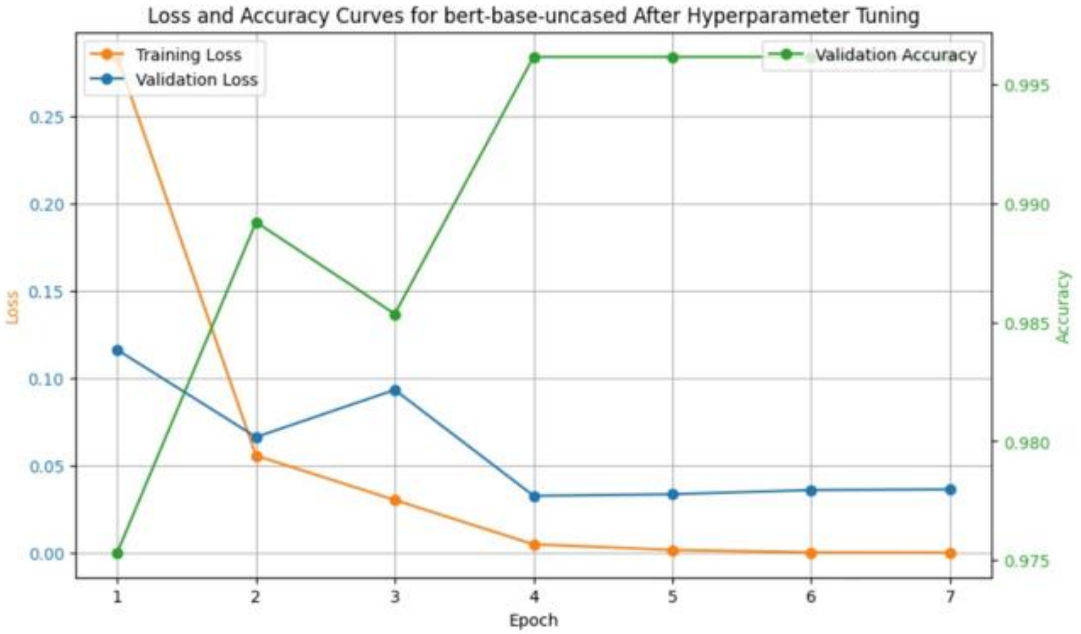


Figure 5-24: Loss and Accuracy Curves for BERT Model After Hyperparameter Tuning

The initial epoch showed a high training loss, as expected, while validation loss was significantly lower, indicating the model's initial grasp of the data. Then the training loss dropped sharply, but the validation loss fluctuated slightly, suggesting that the model was still adjusting its parameters. The training loss continued to decrease steadily, reaching 0.0001 by epoch 7, while the validation loss stabilized around 0.0363, indicating that the model successfully learned without overfitting.

The loss curve plot showed a rapid decrease in training loss, with validation loss showing minor fluctuations but stabilizing, demonstrating the model's consistent learning and generalization capacity after fine-tuning.

The validation accuracy improved consistently across epochs from 0.9753 in epoch 1 to 0.9961 in epoch 7, highlighting the model's enhanced performance with each epoch.

5-7-2) Class-wise Accuracy

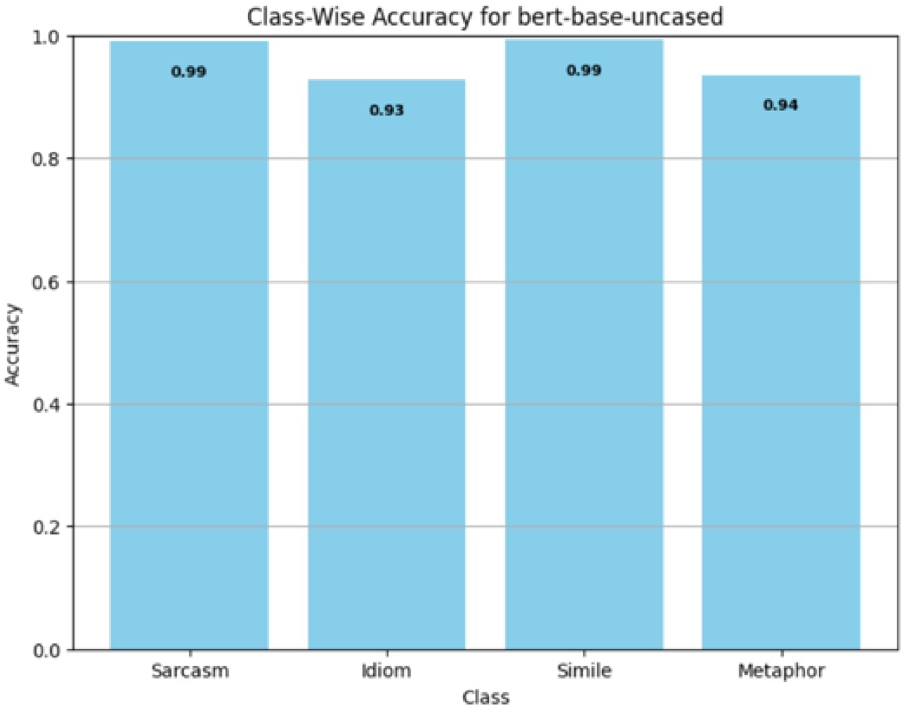


Figure 5-25: Class-wise Accuracy for BERT Model After Hyperparameter Tuning

The class-wise accuracy for each figurative type for the BERT model after fine-tuning is:

- **Sarcasm: 0.99:** Significant improvement, showing the model's enhanced ability to detect sarcasm, due to fine-tuning, which enhanced BERT's ability to detect sarcastic tone and context.
- **Idiom: 0.93:** Good performance but lower than sarcasm and simile, reflecting the challenge idioms posed due to their dependency on cultural and contextual knowledge.
- **Simile: 0.99:** Maintained high accuracy, suggesting that the model easily recognized similes due to their clear comparative structure.
- **Metaphor: 0.94:** Lower than sarcasm and simile but still strong, indicating that while BERT captured metaphorical meanings well, abstract and non-literal relationships posed some challenges.

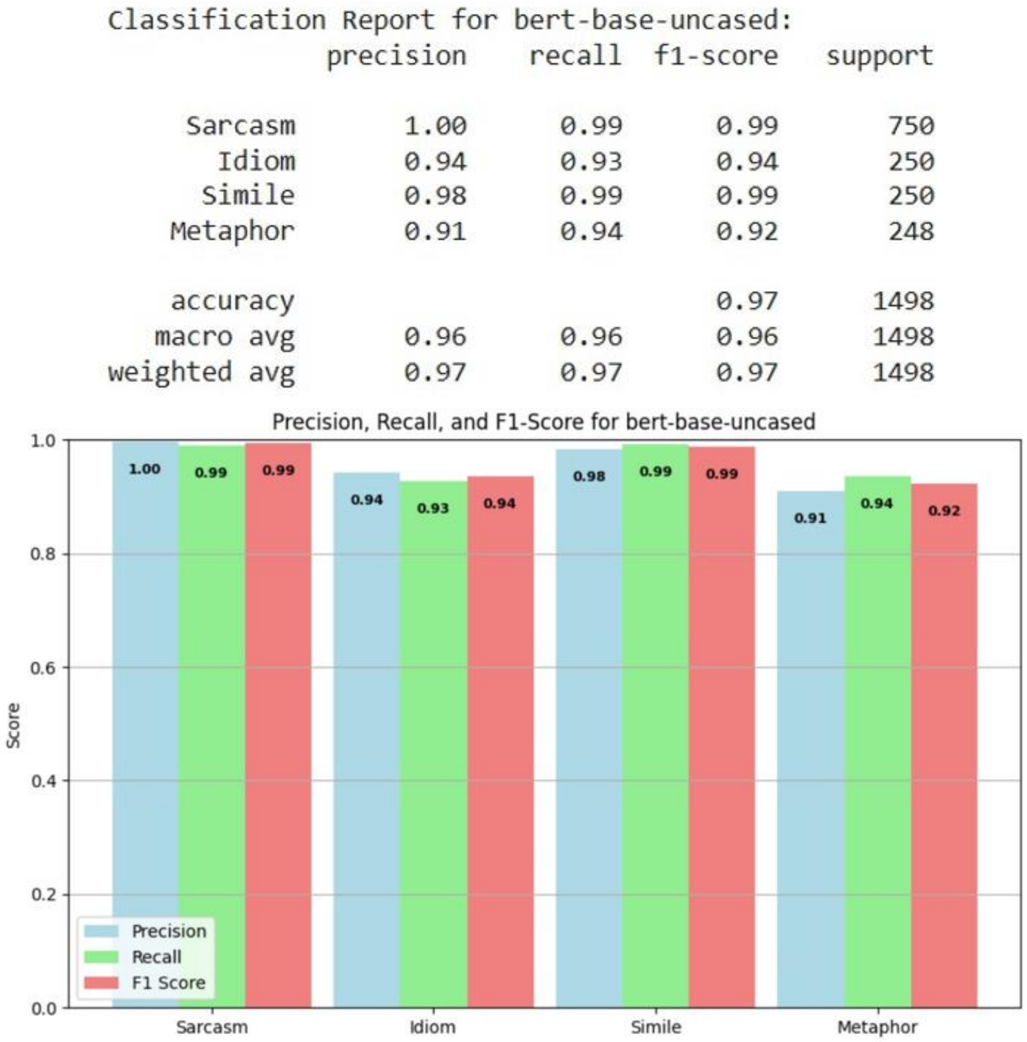
After fine tuning process, the BERT model exhibited a significant improvement in sarcasm detection, reflecting enhanced contextual understanding. Idiom classification saw a slight improvement, while the performance for simile and metaphor recognition remained consistent with pre-tuning results, indicating that fine-tuning helped maintain the model's strengths while slightly boosting its ability to handle more nuanced figurative language like idioms.

5-7-3) Classification Report Analysis

The classification report for the BERT model reveals the following performance metrics across different figurative language types:

- **Sarcasm (Precision: 1.00, Recall: 0.99, F1-score: 0.99):** Sarcasm detection improved significantly due to fine-tuned contextual embeddings, likely because sarcasm was the most represented figurative type in the training set.

- **Idiom (Precision: 0.94, Recall: 0.93, F1-score: 0.94):** High but slightly lower than sarcasm and simile. The model struggled with idioms, reflecting the challenge of recognizing context-dependent, culturally specific expressions.
- **Simile (Precision: 0.98, Recall: 0.99, F1-score: 0.99):** Maintained high performance due to the explicit comparative structure of similes, which BERT's token-level embeddings captured well.
- **Metaphor (Precision: 0.91, Recall: 0.94, F1-score: 0.92):** Slightly lower F1-score suggests that while BERT captured metaphorical meanings, the abstract nature of metaphors posed challenges.



```
Classification Report for bert-base-uncased:
              precision    recall  f1-score   support

     Sarcasm       1.00      0.99      0.99       750
       Idiom       0.94      0.93      0.94       250
      Simile       0.98      0.99      0.99       250
    Metaphor       0.91      0.94      0.92       248

    accuracy                           0.97      1498
   macro avg       0.96      0.96      0.96      1498
weighted avg       0.97      0.97      0.97      1498
```

Figure 5-26: Classification Report Analysis for BERT Model After Hyperparameter Tuning

- **Macro Average (Precision: 0.96, Recall: 0.96, F1-score: 0.96):** Indicates that, on average, the model was highly precise across all classes and correctly identified 96% of the figurative language examples across all classes.
- **Weighted Average (Precision: 0.97, Recall: 0.97, F1-score: 0.97):** Higher than the macro average values due to the significant number of sarcasm samples, which the model handled exceptionally well after fine-tuning. The higher weight of the sarcasm class boosted the overall precision and recall masking slight weaknesses in idioms and metaphors. The F1-score confirmed that the model performed consistently well across all classes, but it was notably influenced by sarcasm's high performance and larger sample size.

5-7-4) Confusion Matrix Analysis

The confusion matrix reveals the model's classification tendencies and errors:

- **Sarcasm:** 743 correctly predicted, with 2 misclassifications as idiom and 5 as metaphor.
- **Idiom:** 232 correct predictions, with 18 misclassifications as metaphor.
- **Simile:** Perfect classification with 248 correct predictions and only 2 misclassifications with idiom.
- **Metaphor:** 232 correct predictions, with minimal misclassifications into sarcasm, idiom and simile.

Fine-tuning reduced misclassification rates significantly, particularly for sarcasm, which was previously challenging category for the model.

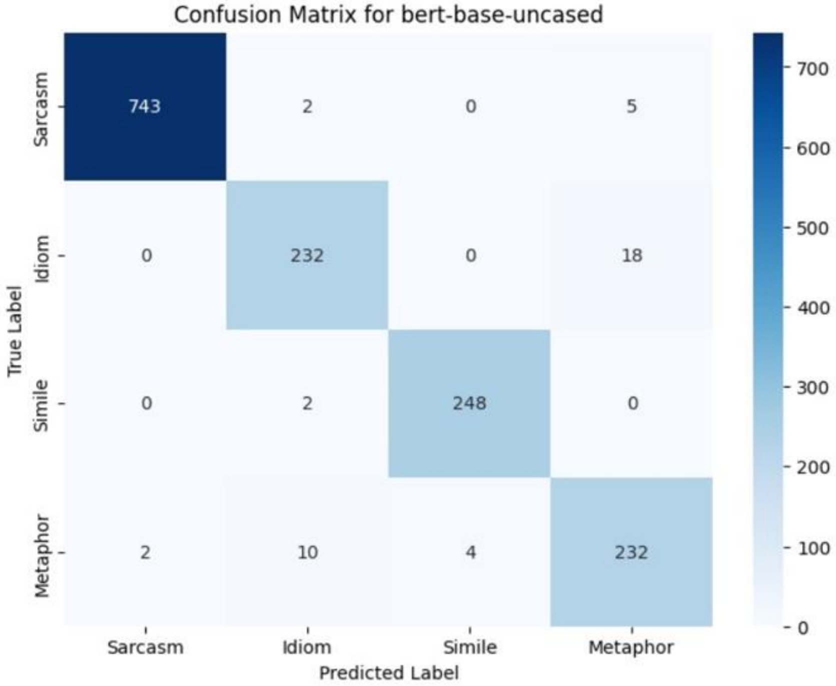


Figure 5-27: Confusion Matrix Analysis for BERT Model After Hyperparameter Tuning

5-7-5) Overall BERT Model Performance

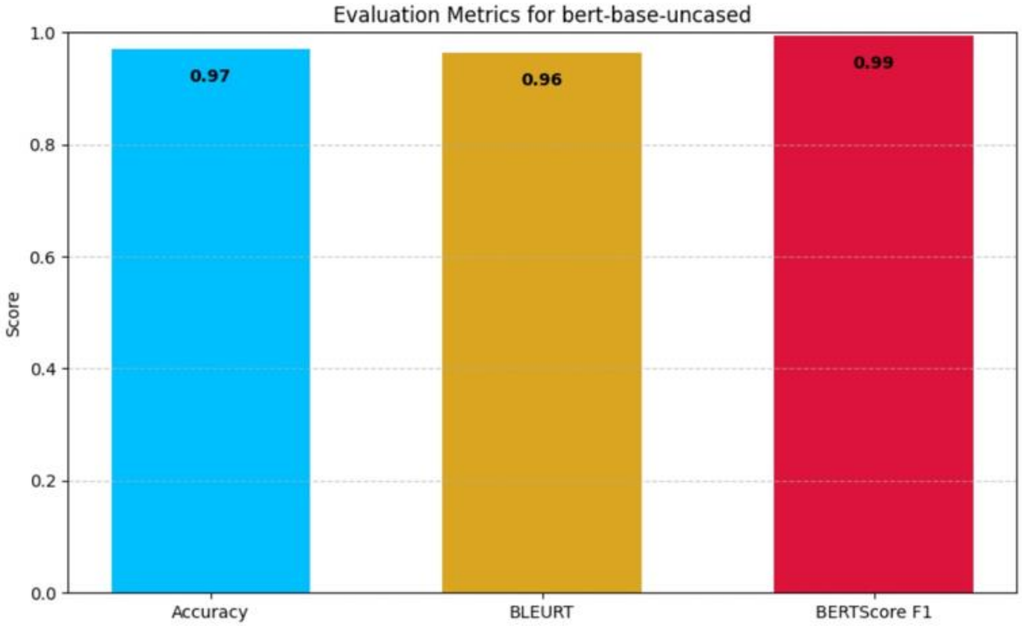


Figure 5-28: Evaluation Metrics Analysis for BERT Model After Hyperparameter Tuning

The overall accuracy of the BERT model on the test set was 0.97, which significantly improved from its pre-tuning value of 0.78. Additionally, the model achieved the “Average BLEURT Score” equal to 0.96, Highlighting a high degree of textual similarity between predictions and reference sentences. The “Average BERTScore F1” was equal to 0.99 Indicating near-perfect token-level semantic similarity. The bar plot of evaluation metrics for BERT model after hyperparameter tuning showed all three metrics approached 1.0, indicating the success of fine-tuning in enhancing the model’s performance across different evaluation criteria.

5-8) Comparison of Metrics Across Models After Hyperparameter Tuning

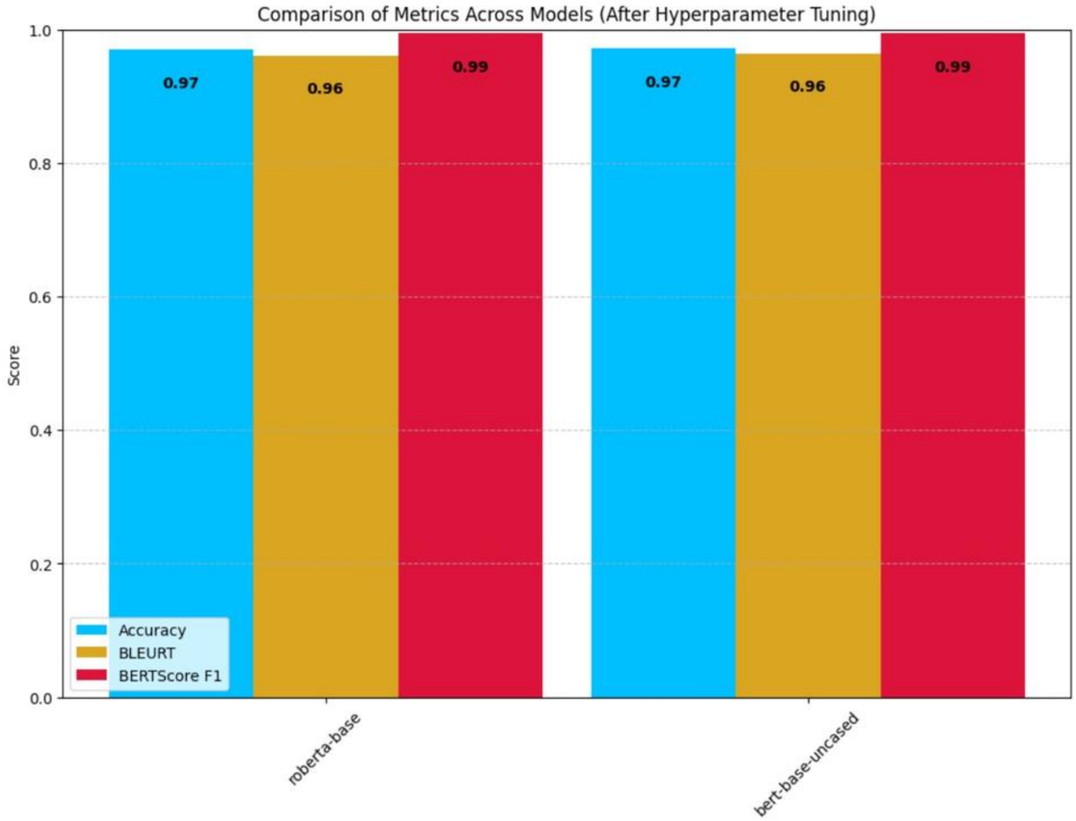


Figure 5-29: Comparison of Metrics Across Models After Hyperparameter Tuning

After hyperparameter tuning, both ROBERTA and BERT models demonstrated significant performance improvements, achieving identical scores across all metrics: Accuracy (0.97), BLEURT (0.96), and BERTScore F1 (0.99), indicating that fine-tuning successfully enhanced their figurative language classification capabilities to near-optimal levels.

5-9) Comparison of the Overall ROBERTA and BERT Performances Before and After Hyperparameter Tuning

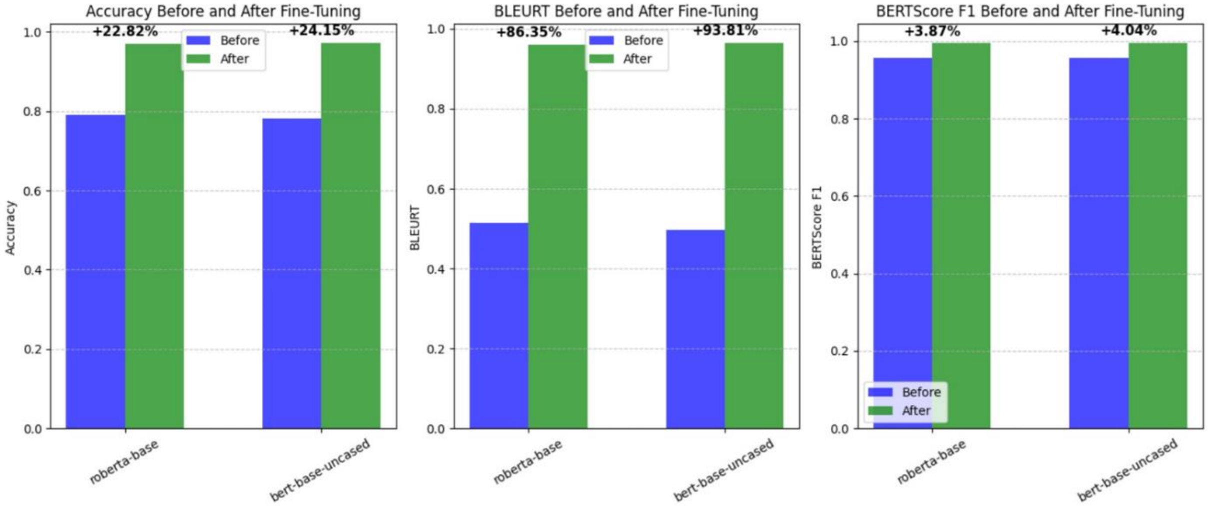


Figure 5-30: Comparison of Overall Performances of the Models Before and After Hyperparameter Tuning

The comparison of model performances before and after hyperparameter tuning highlights substantial improvements in both ROBERTA and BERT models. Notably, accuracy increased by 22.82% and 24.15% respectively, while BLEURT scores saw dramatic boosts of 86.35% and 93.81%. The BERTScore F1 improvements, though more modest at 3.87% and 4.04%, demonstrate that fine-tuning effectively enhanced the overall quality and reliability of both models.

5-10) T5 Model Fine-Tuning and Evaluation Results

5-10-1) Training and Validation Losses

The T5-base model was fine-tuned for the task of figurative language classification, distinguishing between sarcasm, idiom, simile, and metaphor. The fine-tuning involved converting the classification task into a text-to-text format, where the input consisted of a prompt specifying the classification task along with the premise and hypothesis, and the output was the corresponding figurative label.

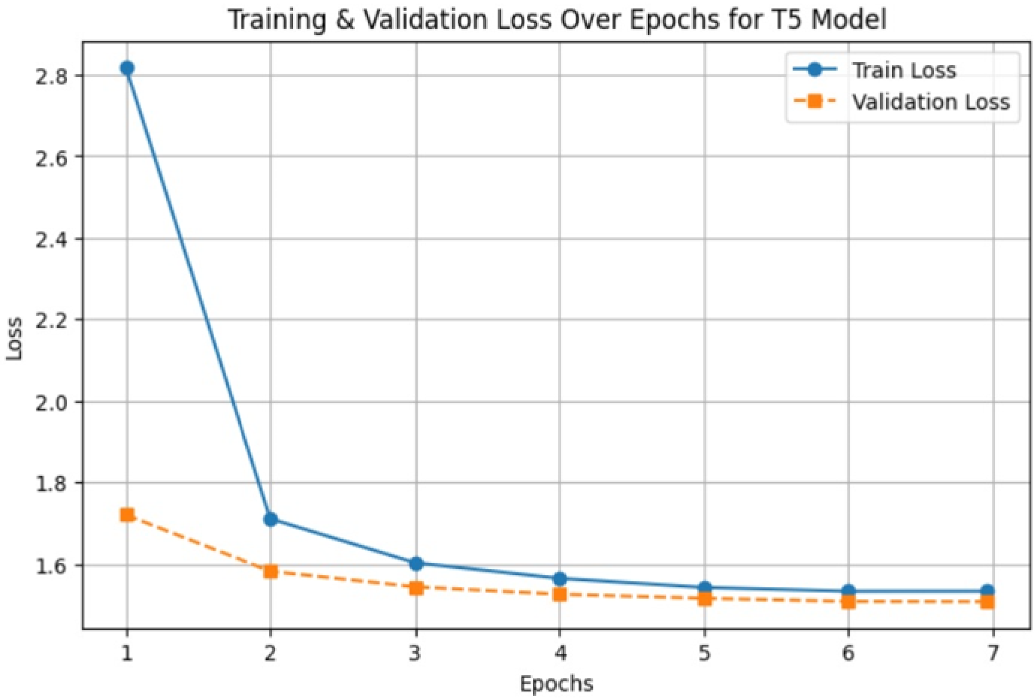


Figure 5-31: Training and Validation Loss Over Epochs for T5 Model

The training and validation loss curves showed a steep decline in the initial epochs, indicating rapid learning during the early stages. After epoch 3, the loss values stabilized, with minimal fluctuations, suggesting that the model converged effectively. The final training and validation loss values were approximately 1.53 and 1.51 respectively, indicating that while the model generalized well, there was some room for improvement, especially in handling more complex figurative language examples.

5-10-2) Class-wise Accuracy

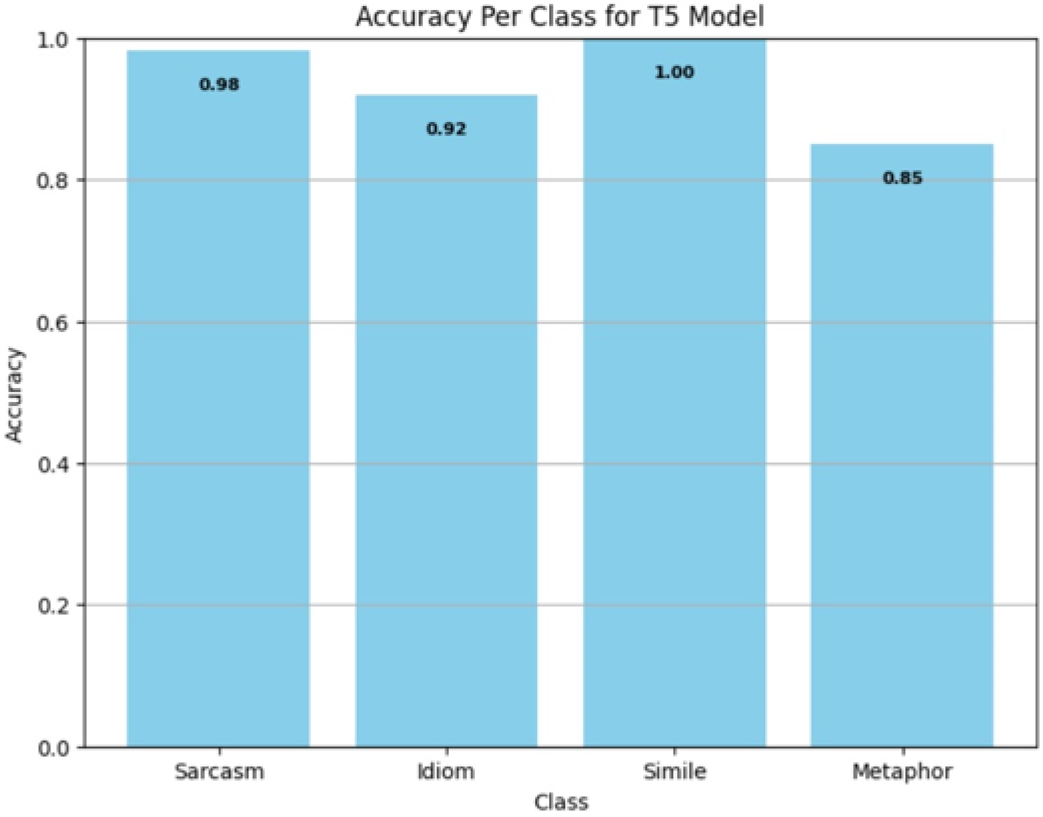


Figure 5-32: Class-wise Accuracy for T5 Model

The class-wise accuracy plot for the T5 model demonstrated its strong performance across all figurative language categories.

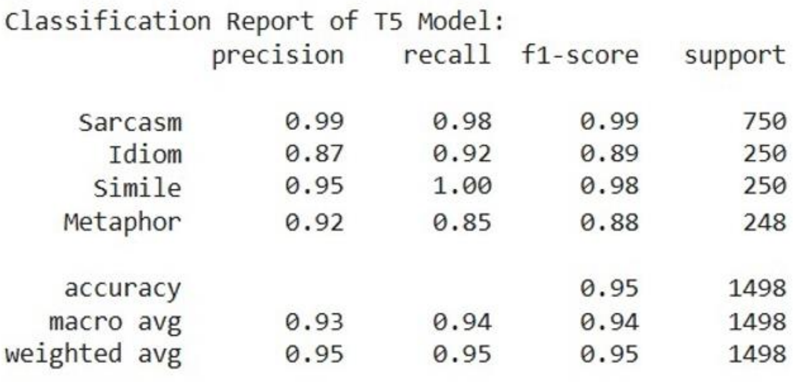
- **Sarcasm: 0.98:** Followed closely with an accuracy of 0.98, reflecting T5's high sensitivity to sarcastic expressions.
- **Idiom: 0.92:** The model performed well for Idiom but lower than sarcasm, reflecting the challenge idioms posed due to their dependency on cultural and contextual knowledge.
- **Simile: 1.00:** The model achieved perfect accuracy for simile indicating its exceptional capability to classify similes correctly.
- **Metaphor: 0.85:** The lowest accuracy was observed in the Metaphor category. Despite being the lowest, the metaphor

classification accuracy was still relatively high, demonstrating the T5 model's robust generalization ability across diverse figurative types.

5-10-3) Classification Report Analysis

The classification report for the T5 model reveals the following performance metrics across different figurative language types:

- **Sarcasm (Precision: 0.99, Recall: 0.98, F1-score: 0.99):** Sarcasm achieved a near-perfect performance, showing that the model was highly effective in identifying sarcastic statements with minimal false positives and false negatives.
- **Idiom (Precision: 0.87, Recall: 0.92, F1-score: 0.89):** While the precision was slightly lower, the high recall indicated that the model captured most idiomatic expressions, though it might sometimes misclassify non-idiomatic text as idioms.
- **Simile (Precision: 0.95, Recall: 1.00, F1-score: 0.98):** Simile exhibited strong performance, demonstrating that the T5 model accurately identified similes without missing any examples.
- **Metaphor (Precision: 0.92, Recall: 0.85, F1-score: 0.88):** While the model was generally accurate, it tended to miss a few metaphorical expressions.
- **Macro Average (Precision: 0.93, Recall: 0.94, F1-score: 0.94):** Reflecting strong and consistent performance across all figurative types, with only a slight drop in recall due to the lower performance on idiom and metaphors.
- **Weighted Average (Precision: 0.95, Recall: 0.95, F1-score: 0.95):** The model performed well even for more challenging classes like metaphor and idiom, despite their fewer examples compared to sarcasm.



```
Classification Report of T5 Model:
              precision    recall  f1-score   support

     Sarcasm       0.99      0.98      0.99       750
       Idiom       0.87      0.92      0.89       250
      Simile       0.95      1.00      0.98       250
    Metaphor       0.92      0.85      0.88       248

    accuracy                           0.95      1498
   macro avg       0.93      0.94      0.94      1498
weighted avg       0.95      0.95      0.95      1498
```

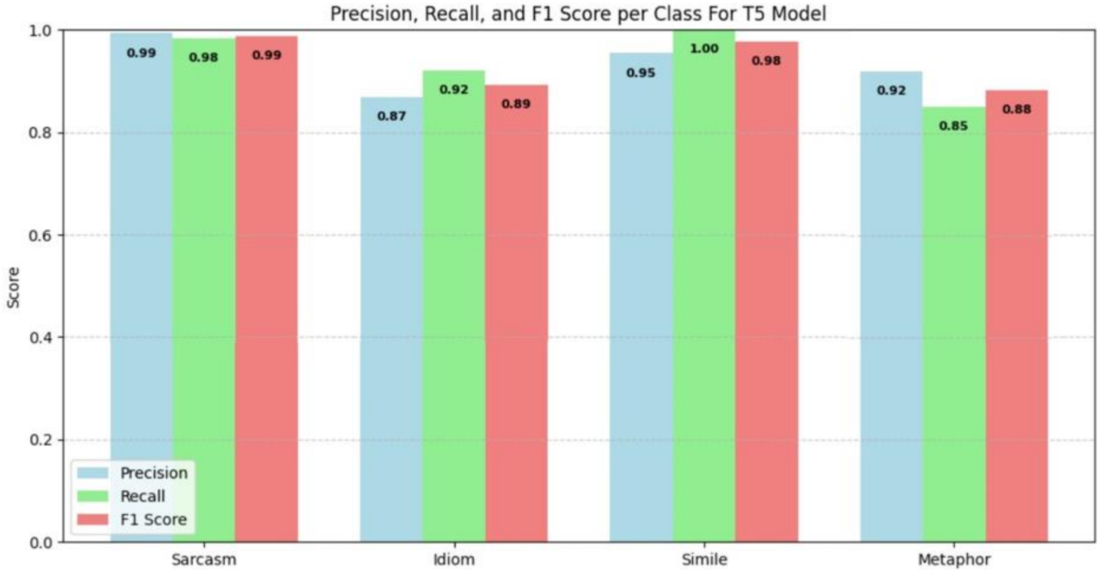


Figure 5-33: Classification Report Analysis for T5 Model

5-10-4) Confusion Matrix Analysis

The confusion matrix reveals the model’s classification tendencies and errors:

- **Sarcasm:** 737 correctly predicted, with 9 misclassifications as idiom, 2 as simile, and 2 as metaphor.
- **Idiom:** 230 correct predictions, with 16 misclassifications as metaphor and 4 as simile.
- **Simile:** Perfect classification with all 250 correct predictions.

- **Metaphor:** 211 correct predictions, with 26 misclassifications as idiom, 6 as simile, and 5 as sarcasm.

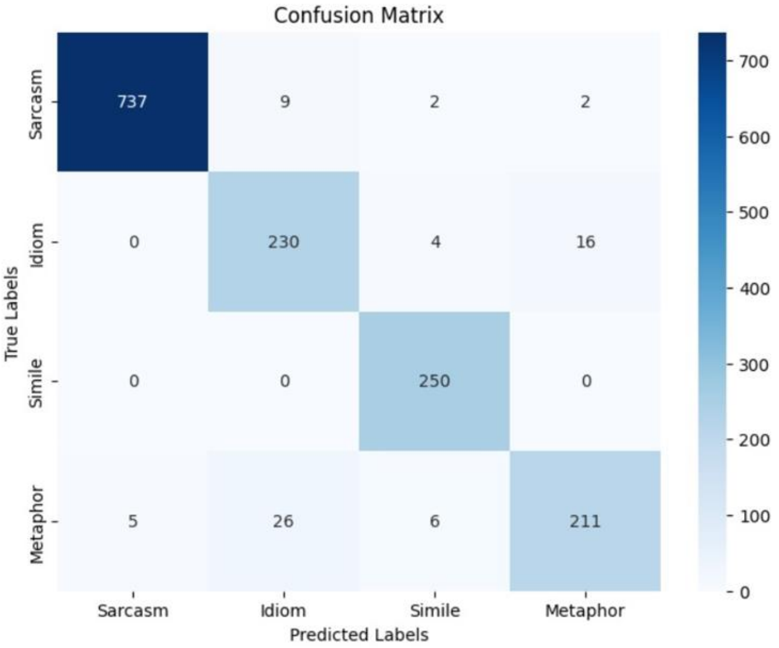


Figure 5-34: Confusion Matrix Analysis for T5 Model

5-10-5) Overall T5 Model Performance

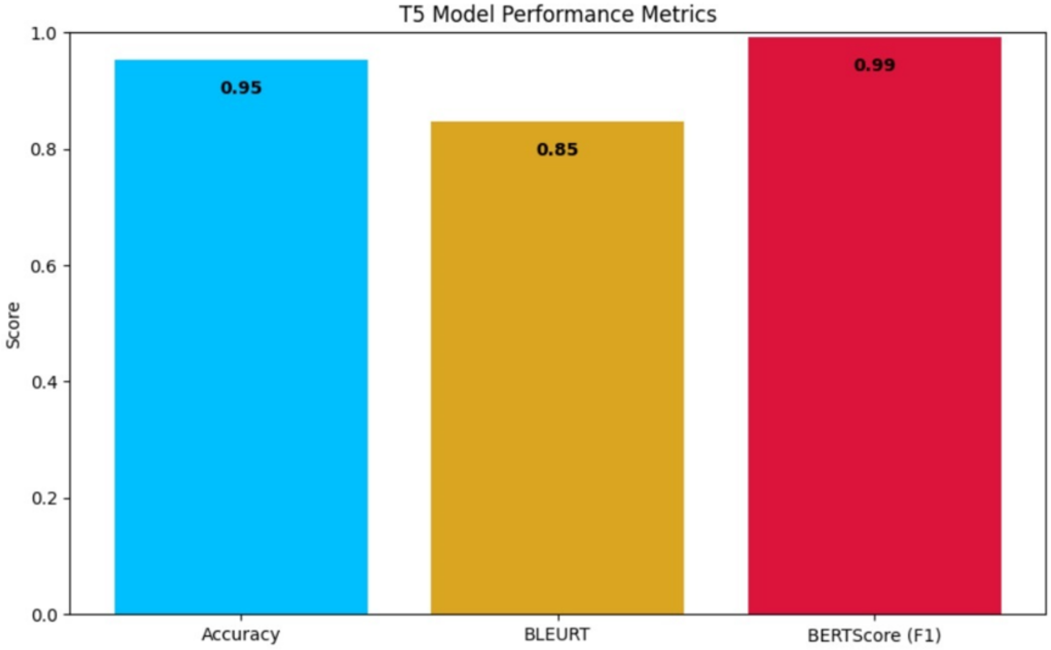


Figure 5-35: Evaluation Metrics Analysis for T5 Model

The overall accuracy of the T5 model on the test set was 0.95, reflecting high overall classification performance across all figurative types. The model achieved the “Average BLEURT Score” equal to 0.85, indicating strong semantic similarity between predictions and ground truth. The “Average BERTScore F1” was equal to 0.99 highlighting exceptional token-level alignment with the correct labels.

5-11) Results Comparison of T5, ROBERTA, and BERT Models

5-11-1) Class-wise Accuracy Comparison

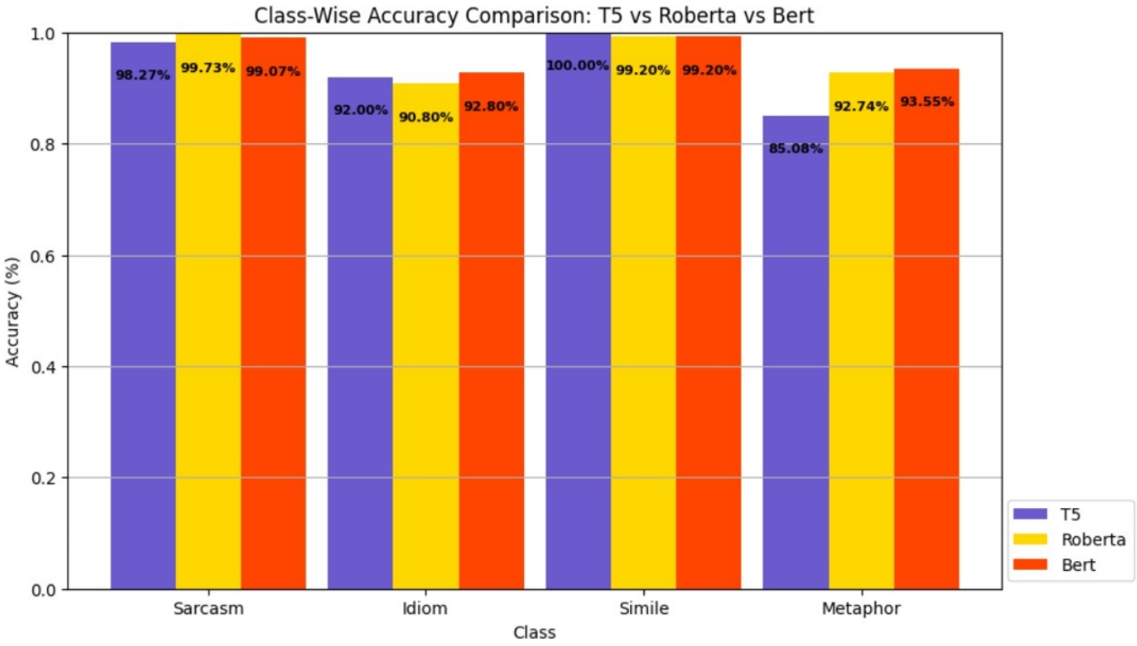


Figure 5-36: Class-Wise Accuracy Comparison: T5 vs ROBERTA vs BERT

The class-wise accuracy comparison between the T5, and the fine-tuned ROBERTA and BERT models reveals notable differences in performance across figurative language types:

- **Sarcasm:** T5 achieved an accuracy of 98.27%, slightly lower than ROBERTA's 99.73% and BERT's 99.07%, indicating that both fine-tuned

Roberta and Bert were marginally more consistent in detecting sarcastic instances.

- **Idiom:** The BERT model led with an accuracy of 92.80%, followed closely by T5 at 92% and ROBERTA at 90.80%, highlighting that while all models performed well, fine-tuned BERT demonstrated slightly better precision in capturing idiomatic expressions.
- **Simile:** T5 stood out with a perfect accuracy of 100%, outperforming both ROBERTA and BERT, each at 99.20%, demonstrating T5's superior ability to correctly identify all simile instances without errors.
- **Metaphor:** BERT achieved the highest accuracy at 93.55%, slightly surpassing ROBERTA at 92.74%, while T5 trailed with 85.08%, indicating that metaphor detection posed more challenges for the T5 model compared to its counterparts.

This comparison highlighted T5's dominance in simile classification, while ROBERTA and BERT performed better in sarcasm, idiom and metaphor detection, suggesting that different models exhibited strengths in varying figurative language categories.

5-11-2) Precision, Recall, and F1-score Comparison

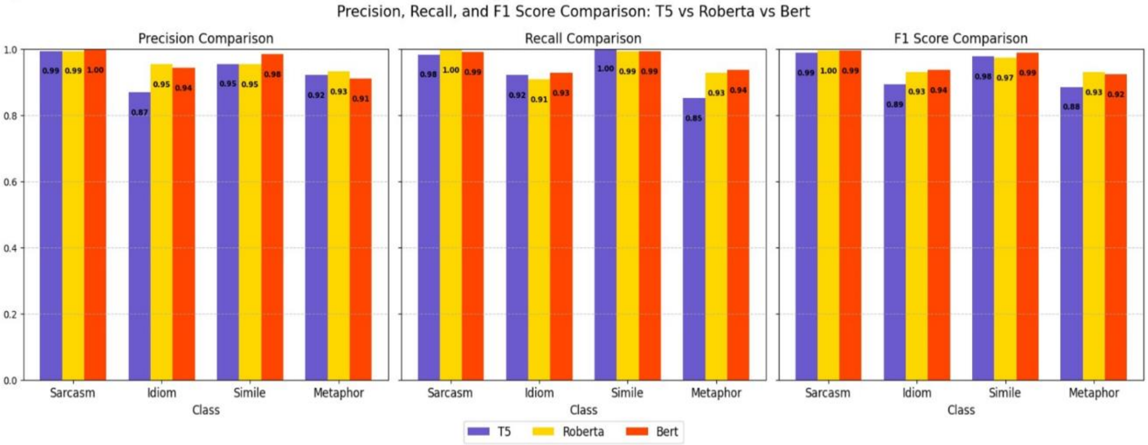


Figure 5-37: Precision, Recall, and F1-score Comparison: T5 vs ROBERTA vs BERT

The precision, recall, and F1-score comparison between the T5, and the fine-tuned ROBERTA and BERT models across figurative language classes showcased the following insights:

➢ **Precision:**

- **Sarcasm:** The fine-tuned BERT model achieved the highest precision at 1.00, while both T5 and ROBERTA scored 0.99, indicating BERT's perfect accuracy in identifying sarcastic instances approximately without any false positives.
- **Idiom:** The fine-tuned ROBERTA led with 0.95, followed closely by BERT at 0.94 and T5 at 0.87, showing ROBERTA's better handling of idiomatic expressions.
- **Simile:** The fine-tuned BERT model achieved the highest precision at 0.98, with T5 and ROBERTA both at 0.95.
- **Metaphor:** The fine-tuned ROBERTA scored 0.93, T5 at 0.92, and BERT slightly lower at 0.91, indicating ROBERTA's superior precision in recognizing metaphoric content.

➢ **Recall:**

- **Sarcasm:** The fine-tuned ROBERTA excelled in sarcasm with a perfect recall of 1.00, followed by BERT at 0.99 and T5 at 0.98.
- **Idiom:** The fine-tuned BERT model achieved the highest recall at 0.93, with T5 and ROBERTA at 0.92 and 0.91 respectively.
- **Simile:** The T5 model achieved the highest recall at 1.00, with both ROBERTA and BERT at 0.99.
- **Metaphor:** BERT model had the highest recall at 0.94, while ROBERTA and T5 scored 0.93 and 0.85, respectively, indicating BERT's better ability to capture all metaphoric instances.

➢ **F1-Score:**

- **Sarcasm:** The fine-tuned ROBERTA model achieved the highest F1-score at 1.00, with both T5 and BERT at 0.99.
- **Idiom:** ROBERTA achieved an F1-score of 0.93, BERT at 0.94, and T5 at 0.89, demonstrating BERT's balanced performance.
- **Simile:** The fine-tuned BERT stood out with an F1-score of 0.99, compared to 0.98 for T5 and 0.97 for ROBERTA.
- **Metaphor:** BERT achieved an F1-score of 0.92, ROBERTA at 0.93, and T5 at 0.88, showing that fine-tuned ROBERTA and BERT models handled metaphors more consistently.

5-11-3) Overall Models' Performance Comparison

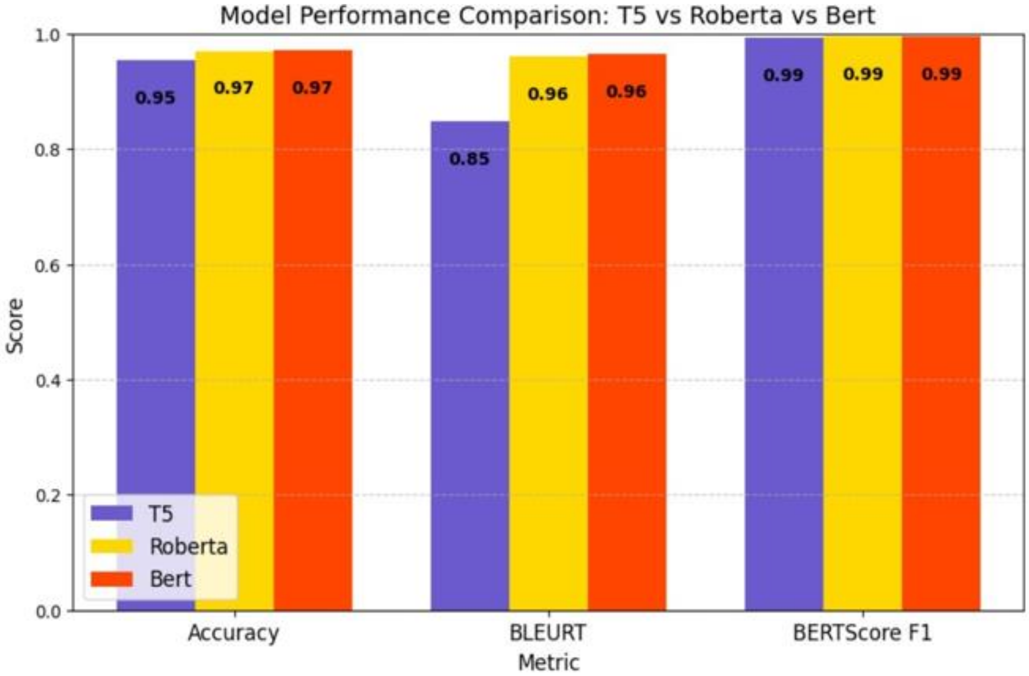


Figure 5-38: Overall Models' Performance Comparison: T5 vs ROBERTA vs BERT

The model performance comparison across the T5, and the fine-tuned ROBERTA and BERT models based on Accuracy, BLEURT, and BERTScore F1 metrics, highlights key distinctions:

- **Accuracy:** ROBERTA and BERT both achieved the highest accuracy at 0.97, showcasing their superior overall classification performance, while T5, with an accuracy of 0.95, trailed slightly behind, indicating a marginally lower correctness rate in its predictions compared to the other two models.
- **BLEURT Score:** ROBERTA and BERT again shared the top position with a BLEURT score of 0.96, reflecting their robustness in generating predictions that were closely aligned with human annotations. T5 in contrast, showed a notable gap, scoring 0.85, suggesting its generated outputs were slightly less aligned with human references, possibly due to its generation-based approach compared to the classification-based ROBERTA and BERT.
- **BERTScore F1:** All three models performed exceptionally well, each achieving a score of 0.99, indicating that the quality of predictions across the models was highly consistent when compared to ground truth labels.

In conclusion, the fine-tuned ROBERTA and BERT models demonstrated comparable and superior performance in terms of accuracy and BLEURT, while T5 exhibited slightly lower accuracy and BLEURT scores, though it matched the other models in BERTScore F1, underscoring its capability to produce high-quality predictions despite its lower alignment and accuracy.

6) Discussion

This study explored the effectiveness of transformer-based large language models in understanding and classifying various forms of figurative language, including sarcasm, idioms, similes, and metaphors. Leveraging the FLUTE dataset, the research employed a combination of probing tasks and classification models, specifically ROBERTA and BERT, while also comparing their fine-tuned performance to that of a text generation-based model, T5. The analysis revealed notable performance variations across different figurative language types and models, underscoring the inherent complexity of processing figurative expressions within natural language processing (NLP) systems.

The findings highlight the strengths of transformer-based models in handling intricate linguistic constructs, particularly when fine-tuned on diverse, multi-type figurative datasets such as FLUTE. This study underscores the critical role of tailored training processes in enhancing the interpretative capabilities of NLP models across various figurative forms, demonstrating the nuanced challenges and potential advancements in figurative language understanding within the field of artificial intelligence.

6-1) Key findings

The study revealed significant variations in the performance of transformer-based models when processing different types of figurative language. Sarcasm posed the greatest challenge prior to fine-tuning, with BERT exhibiting the highest misclassification rate among the transformer-based models. Although ROBERTA performed slightly better, both models struggled with the implicit nature of sarcasm, which heavily relies on context and tone for accurate interpretation.

Idioms, despite maintaining relatively high accuracy across models, did not show notable improvement after fine-tuning. This consistent performance can be attributed to the structured and formulaic nature of idioms, which often adhere to fixed linguistic patterns that transformer models can more easily recognize.

Similes achieved the highest classification accuracy, largely due to their explicit comparative structure (e.g., "X is like Y"), which aligns well with token-based pattern recognition inherent in transformer architectures. Metaphors, while generally well-recognized, presented interpretative challenges due to their abstract and context-dependent nature, with T5’s text-to-text generation approach demonstrating weaker performance in metaphor processing compared to classification-based models.

Fine-tuned ROBERTA and BERT models outperformed T5 in terms of accuracy and BLEURT scores, indicating their superior precision and consistency in figurative language classification. However, T5 maintained comparable performance in BERTScore F1, suggesting that while generation-based models can produce high-quality textual outputs, classification-based models like ROBERTA and BERT are more reliable for nuanced figurative language tasks.

6-2) Interpretation and Comparison with Existing Literature

The findings of this study align with existing literature, which acknowledges the advancements in NLP models' figurative language comprehension while highlighting persistent challenges in handling context-dependent and non-literal expressions. Sarcasm emerged as the most challenging figurative type to process, consistent with prior research indicating that sarcasm often relies on multi-modal cues such as tone and facial expressions (Poria et al., 2016), which are absent in textual datasets. This study supports Ghosh et al. (2018), who identified sarcasm as particularly difficult for computational models due to its implicit and context-sensitive nature.

Idioms, on the other hand, benefited from their lexicalized and structured nature, making them easier for transformer models to recognize, even when fine-tuning yielded limited improvements. These findings align with Katz and Fodor's (1963) view of idioms as fixed expressions and recent studies by Feldmann et al. (2021), which demonstrated that fine-tuned transformers effectively capture such linguistic patterns.

The high classification accuracy of similes corroborates Veale's (2012) research, which highlighted the advantage of their explicit comparative structure (e.g., "X is like Y") for machine learning models. The predictable pattern of similes provides clear linguistic markers that transformers can readily exploit for accurate classification.

While metaphors were generally well-recognized, their interpretive complexity remained a challenge, reflecting Lakoff and Johnson's (1980) argument that the non-compositional nature of metaphorical meaning poses significant difficulties for computational models. This study reinforces the notion that metaphors, due to their abstractness and reliance on contextual understanding, continue to challenge even state-of-the-art transformer-based models.

6-3) Implications for NLP and Future Model Development

The findings of this study offer valuable insights into the future development of NLP systems aimed at processing and generating human-like language, particularly in the context of figurative language comprehension. The results suggest several key directions for enhancing model performance:

- Enhanced sarcasm detection mechanisms are essential, as current transformer-based models struggle with the implicit nature of sarcasm. Integrating external sentiment analysis tools or employing contextual embeddings that capture discourse-level information could improve the ability of NLP models to detect sarcasm by providing a richer understanding of conversational context.
- Expanding training datasets with real-world usage examples could significantly enhance the comprehension of idioms and metaphors. Incorporating more diverse and contextually rich data would allow models to better recognize and interpret these figurative expressions, addressing the challenges identified in this study.
- Hybrid models that combine rule-based approaches with deep learning may offer improved performance, particularly when distinguishing between literal and figurative meanings. While deep learning models excel at pattern recognition, rule-based methods can provide explicit guidelines for handling complex figurative constructs, creating a more balanced and accurate system.
- Interactive and multi-modal learning approaches, such as incorporating tone, gestures, or speaker intent, could further enhance the detection of sarcasm. Given that sarcasm often relies on non-textual cues, integrating

multi-modal data could bridge the gap identified in text-based NLP models and improve their interpretative capabilities.

These implications highlight the need for more sophisticated and context-aware models, reinforcing the challenges and opportunities uncovered in this study of figurative language understanding within transformer-based NLP systems.

6-4) Limitations of the Study

While in this study, I tried to provide valuable insights into the capabilities of transformer-based large language models in processing figurative language, several limitations should be acknowledged:

- **Dataset Constraints:** Although the FLUTE dataset is comprehensive and covers multiple types of figurative language, it may not fully capture the richness and diversity of figurative expressions encountered in real-world contexts. Figurative language is highly dynamic and context-sensitive, and the dataset's scope may limit the generalizability of the findings.
- **Lack of Real-World Discourse Context:** This study primarily evaluated sentence-level understanding of figurative language, which does not account for broader discourse structures where figurative expressions often derive meaning from context, sequential information, and conversational flow. Future research incorporating discourse-level analysis could provide a more holistic assessment of NLP models' figurative language comprehension.
- **Limitations of Transformer-Based Models:** Despite their advanced capabilities, transformer-based models continue to face challenges in deep semantic understanding, particularly when processing non-literal meanings. Their reliance on surface-level lexical patterns often leads to difficulties in capturing the nuanced and abstract nature of figurative language, such as metaphors and sarcasm.

- **Absence of Human Evaluation:** This study relied solely on automated metrics and model-based evaluations, without incorporating human judgment to assess the quality and accuracy of figurative language processing. Human evaluation could have provided a more nuanced assessment, particularly for figurative expressions that are inherently subjective and context dependent.

Recognizing these limitations offers a foundation for future research aimed at enhancing NLP models' ability to process complex linguistic constructs, particularly within the realm of figurative language.

6-5) Future Research Directions

Building on the findings and limitations of this study, future research could explore several promising directions to enhance the performance and robustness of NLP models in processing figurative language:

- **Integration of Diverse Figurative Language Datasets:** Expanding the variety and scope of datasets by incorporating more diverse figurative language examples from longer texts, dialogues, and real-world discourse could improve model generalization. Such datasets would provide richer contextual references, enabling models to better handle complex figurative constructs like sarcasm, idioms, metaphors, and similes.
- **Human-in-the-Loop Evaluation Processes:** Incorporating human evaluations into the training and testing phases could offer deeper insights into model performance. Human feedback would be particularly valuable for assessing subtle figurative expressions, where automated metrics often fall short, and could guide model adjustments in real-time.
- **Advanced Training Techniques:** Future studies could investigate advanced methods such as multi-task learning, where models are trained on multiple related tasks simultaneously, and adversarial training, which enhances robustness by exposing models to challenging or deceptive

inputs. Both approaches could improve the interpretative accuracy of figurative language processing.

- **Exploration of Transformer Attention Mechanisms:** Analyzing attention mechanisms in deeper layers of transformer models could reveal whether certain layers specialize in processing specific types of figurative language. This understanding could inform targeted model improvements and fine-tuning strategies.
- **Reinforcement Learning for Dynamic Adaptation:** Implementing reinforcement learning-based training could enable models to adapt dynamically to various figurative contexts, continuously improving their performance through iterative learning and feedback.
- **Cross-Linguistic Comparisons:** Investigating figurative language processing across multilingual datasets could provide insights into how models handle figurative expressions in different languages and cultural contexts, highlighting potential biases or limitations.
- **Human-Model Collaboration:** Future research could also explore human-model collaboration by incorporating user feedback into model training and deployment, enhancing the ability of NLP systems to process figurative language in real-world applications.

These directions aim to address existing challenges while paving the way for more sophisticated and context-aware NLP models capable of handling the intricacies of figurative language with greater accuracy and reliability.

6-6) Conclusion

This study contributes to the growing field of figurative language processing by evaluating the performance of transformer-based large language models, including BERT, ROBERTA, and T5, on a multi-type figurative language dataset (FLUTE). The research demonstrated that fine-tuning these models on diverse figurative constructs significantly improves their ability to classify and interpret non-literal language, including sarcasm, idioms, similes, and metaphors.

The incorporation of the Holmes probing task provided valuable insights into the internal representations of these models, revealing how different layers and components process various figurative language types. This methodological approach influenced the study by offering a granular analysis of model behavior, highlighting both strengths and persistent gaps in figurative language comprehension.

Despite this progress, notable challenges persist, particularly in detecting sarcasm and comprehending abstract metaphors, which rely heavily on context, tone, and nuanced semantic understanding. The findings underscore the importance of refining model architectures and training methodologies to enhance figurative language processing. Leveraging hybrid approaches that combine rule-based systems with deep learning techniques, integrating broader contextual cues from real-world discourse, and expanding multi-modal inputs such as tone and speaker intent can push the boundaries of computational understanding of figurative language. These advancements are essential for developing AI-driven communication tools that are not only accurate but also capable of interpreting human language with the subtlety and nuance characteristic of human communication.

7) References

Gibbs, R. W., Jr. (1994). “The Poetics of Mind: Figurative Thought, Language, and Understanding.” Cambridge University Press.

Florman, Ben. (2017). “Figurative Language.” LitCharts. LitCharts LLC, 5 May 2017.

Michaeli et al. (2008). “Figurative Language: “Meaning” is often more than just a sum of the parts.” Association for the Advancement of Artificial Intelligence.

Deignan, A. (2015). “Figurative language and lexicography.” International Handbook of Modern Lexis and Lexicography. Springer, Berlin, Heidelberg.

Hanks, P., de Schryver, GM. (2015). “International Handbook of Modern Lexis and Lexicography.” Springer, Berlin, Heidelberg.

Lal, Y., & Bastan, M. (2022). “SBU Figures It Out: Models Explain Figurative Language.” Proceedings of the 3rd Workshop on Figurative Language Processing (FLP).

Imran, M.M., Chatterjee, P., & Damevski, K. (2023). “Shedding Light on Software Engineering-Specific Metaphors and Idioms.” 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), 2555-2567.

Lakoff, G., & Johnson, M. (1980). “Metaphors We Live” By. University of Chicago Press.

Katz, J., & Fodor, J. A. (1963). “The Structure of a Semantic Theory.” Language, Vol. 39, No. 2. (Apr. - Jun. 1963), pp. 170-210.

Veale, T. (2012). “Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity.” Bloomsbury Academic.

Ghosh, D., Fabbri, A. R., & Muresan, S. (2018). “Sarcasm analysis using conversation context.” Computational Linguistics, 44(4), 755-792.

Colston, H. L., & O'Brien, J. (2000). “Contrast of kind versus contrast of magnitude: The pragmatic accomplishments of irony and hyperbole.” Discourse Processes, 30(2), 179–199.

Long, D. (2018). “Meaning Construction of Personification in Discourse Based on Conceptual Integration Theory.” Studies in Literature and Language, 17, 21-28.

Sun, C. (2024). “A Study on the Application of Metonymy in English Poetry.” Scientific and Social Research.

Brown, C.H. (1979). “A Theory of Lexical Change (with Examples from Folk Biology, Human Anatomical Partonomy and Other Domains)” Anthropological Linguistics, 21.

Semaeva, O. (2024). “The Effectiveness of Using Specific English Alliteration in Russian-Language Names and Advertising Slogans.” Linguistics and Intercultural Communication.

Bredin, H. (1996). “Onomatopoeia as a figure and a linguistic principle.” New Literary History, 27(3), 555–569.

Giorgadze, M. (2014). “Linguistic Features of Pun, Its Typology and Classification.” European Scientific Journal, ESJ, 10(10).

Hyewon Jang, Qi Yu, and Diego Frassinelli. (2023). “Figurative Language Processing: A Linguistically Informed Feature Analysis of the Behavior of Language Models and Humans.” Association for Computational Linguistics: ACL 2023, pages 9816–9832

Bisikalo, O.V., Ivanov, Y.E., & Sholota, V. (2019). “Modeling the Phenomenological Concepts for Figurative Processing of Natural-Language Constructions.” International Conference on Computational Linguistics and Intelligent Systems.

Nguyen, H. L., Trung, D. N., Hwang, D., & Jung, J. J. (2015). “KELabTeam: A Statistical Approach on Figurative Language Sentiment Analysis in Twitter.” International Workshop on Semantic Evaluation.

Karamouzas, D., Mademlis, I., & Pitas, I. (2022). “Neural Knowledge Transfer for Sentiment Analysis in Texts with Figurative Language.” 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP).

Tomáš Hercig and Ladislav Lenc. (2017). “The Impact of Figurative Language on Sentiment Analysis.” The International Conference Recent Advances in Natural Language Processing, RANLP 2017, pages 301–308, Varna, Bulgaria. INCOMA Ltd.

Karamouzas, D., Mademlis, I., & Pitas, I. (2022). “Neural Knowledge Transfer for Sentiment Analysis in Texts with Figurative Language.” 2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP).

Tummala, P., & Roa, C.K. (2024). “Exploring T5 and RGAN for Enhanced Sarcasm Generation in NLP.” IEEE Access, 12, 88642-88657.

Jhamtani, H., Gangal, V., Hovy, E., & Berg-Kirkpatrick, T. (2021). “Investigating Robustness of Dialog Models to Popular Figurative Language Constructs.” Conference on Empirical Methods in Natural Language Processing.

Repeko, A.P., Radchikova, N.P. (2001). “Natural Language Understanding: Problems of Figurative Language Processing.” Belarusan Foundation of Fundamental Research.

Vulchanova, M.D., & Vulchanov, V. (2018). “Figurative language processing: A developmental and NLP Perspective.” CLIB.

Mason, Z.J. (2004). “CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System.” Association for Computational Linguistics, 30, 23-44.

Nagels, A., Kauschke, C., Schrauf, J., Whitney, C., Straube, B., & Kircher, T. (2013). “Neural substrates of figurative language during natural speech perception: an fMRI study.” Frontiers in Behavioral Neuroscience.

Reyes, A. (2013). “Linguistic-based Patterns for Figurative Language Processing: The Case of Humor Recognition and Irony Detection.” Proces. del Leng. Natural, 50, 107-109.

Potamias, R.A., Siolas, G. & Stafylopatis, A. (2020). “A transformer-based approach to irony and sarcasm detection.” Neural Comput & Applic 32, 17309–17320

Huiyuan Lai and Malvina Nissim. (2024). “A Survey on Automatic Generation of Figurative Language: From Rule-based Systems to Large Language Models.” ACM Comput. Surv. 56, 10, Article 244.

Nurdinova, G. Sh., & Egamnazarova, D. Sh. (2022). “Irony as a Multipurpose Stylistic Device.” Current Research Journal of Philological Sciences.

Brown, T., et al. (2020). “Language Models are Few-Shot Learners.” NeurIPS.

Devlin, J., et al. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” NAACL-HLT.

Feldman, N., et al. (2021). “Analyzing Idiomatic and Literal Usage with Pre-trained Language Models.” ACL.

Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. (2022). “FLUTE: Figurative Language Understanding through Textual Explanations.” In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7139–7159, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rei, M., et al. (2017). “Jointly Learning Semantic Parser and Natural Language Generator.” EMNLP.

Papineni, K., et al. (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation.”

Liu, J., et al. (2020). “Metaphor Detection Using Contextual Word Embeddings from Transformers.” ACL.

Turney, P. D., et al. (2001). “Mining the Web for Synonyms.” Information Retrieval Journal.

Andreas Waldis, et al. (2024). “Holmes ⌕ A Benchmark to Assess the Linguistic Competence of Language Models.” Association for Computational Linguistics 2024; 12 1616–1647.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). “BERTScore: Evaluating Text Generation with BERT.” International Conference on Learning Representations (ICLR).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” arXiv preprint arXiv:1907.11692

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” Journal of Machine Learning Research (JMLR), 21(140), 1-67.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. (2020). “A Primer in BERTology: What We Know About How BERT Works.” Transactions of the Association for Computational Linguistics, 8:842–866.

Raschka, S., & Mirjalili, V. (2019). “Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2.” Packt Publishing.

Wilcoxon, F. (1945). “Individual Comparisons by Ranking Methods.” Biometrics Bulletin, 1(6), 80–83.

Sasaki, Y. (2007). “The truth of the F-measure. Tech Report.” University of Manchester.

Fawcett, T. (2006). “An introduction to ROC analysis.” Pattern Recognition Letters, 27(8), 861-874.

Sellam, T., Das, D., & Parikh, A. P. (2020). “BLEURT: Learning Robust Metrics for Text Generation”. arXiv preprint arXiv:2004.04696.

Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). “A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks.” International Conference on Computational Linguistics.