



INAOE

A Dual Attention-Based Representation for the Detection of Abusive Language in Texts and Memes

by

Horacio Jesús Jarquín Vásquez

A Dissertation Submitted in Partial Fulfillment of
the Requirement for the Degree of

**Doctor of Science in the Area of Computer
Science**

At

**Instituto Nacional de Astrofísica, Óptica y
Electrónica**

April 2025

Tonantzintla, Puebla, México

Supervised by:

Dr. Hugo Jair Escalante Balderas, INAOE

Dr. Manuel Montes y Gómez, INAOE

©INAOE 2025

The author grants INAOE permission to make
partial or total copies of this work and
distribute them, provided that the
source is mentioned.



A Dual Attention-Based Representation for the Detection of Abusive Language in Texts and Memes

doctoral thesis

BY:

Horacio Jesús Jarquín Vásquez

ADVISORS:

Dr. Hugo Jair Escalante Balderas

Dr. Manuel Montes y Gómez

Instituto Nacional de Astrofísica Óptica y Electrónica
Coordinación de Ciencias Computacionales

Table of Contents

List of Figures	v
List of Tables	ix
Acknowledgments	xiii
Abstract	xv
1. Introduction	1
1.1. Motivation and Justification	4
1.2. Problem Statement	5
1.3. Hypothesis	6
1.4. Research Questions	7
1.5. Objectives	7
1.6. Contributions	8
1.7. Overview of the Research and Main Findings	9
1.8. Scope and Limitations	10
1.9. Publications Derived from this Doctoral Research	10
1.10. Organization of the Document	12
2. Background	13
2.1. Text Classification	13
2.2. Multimodal Machine Learning	14
2.3. Artificial Neural Networks	16
2.4. Deep Learning	18

2.4.1.	Attention Mechanisms	19
2.4.2.	Recurrent Neural Network	27
2.4.3.	Transformer Deep Neural Network	29
2.5.	BERT: Bidirectional Encoder Representations from Transformers . . .	34
2.6.	Evaluation Metrics and Statistical Test	35
2.6.1.	Accuracy	36
2.6.2.	Precision	36
2.6.3.	Recall	36
2.6.4.	F_1 Score	36
2.6.5.	Weighted F_1 Score	37
2.6.6.	Macro-average F_1 Score	37
2.6.7.	Maximum Possible Accuracy	37
2.6.8.	Coincident Failure Diversity	38
2.6.9.	Inter-Annotator Agreement	38
2.6.10.	Bayesian Comparison of Classifiers Using the Wilcoxon Signed-Rank Test	41
3.	Related Work	45
3.1.	Attention Mechanisms and Transformer-based Approaches	45
3.2.	Abusive language detection in social media	50
3.2.1.	Detection of abusive language in text	50
3.2.2.	Detection of abusive language in memes	52
3.3.	Evaluation campaigns for abusive language detection in social media .	54
3.4.	Discussion	57
4.	Proposed Dual Attention Mechanism	59
4.1.	Dual Attention Mechanism	59
4.1.1.	Construction of the Dual-Attention Mechanism	60
4.1.2.	Adapted Architectures for the Evaluation of the DA Mechanism	63
4.2.	Multi-Level Dual Attention	65
4.2.1.	Gated Hierarchical Attention Architecture	66
4.2.2.	Adapted baselines for the Evaluation of the GHA Architecture	69
4.3.	Evaluation and Implementation Details	70
4.3.1.	Evaluation Datasets for the Detection of Abusive Language . .	71
4.3.2.	Evaluation Metrics for the Detection of Abusive Language . .	74

4.3.3.	Adapted baselines for the Evaluation of the DA Mechanism . .	75
4.3.4.	Implementation Details	76
4.4.	Quantitative Results	81
4.4.1.	Effectiveness of the Proposed Dual Attention Mechanism . . .	81
4.4.2.	Effectiveness of the Proposed Multi-Level Dual Attention Ar- chitectures	84
4.4.3.	On the Relevance of the Dual Attention Mechanism	87
4.4.4.	Statistical Significance Analysis	88
4.5.	Analysis of Results	91
4.5.1.	Relevance Analysis of the BERT Encoding Layers Using the GHA Architecture	91
4.5.2.	Analysis of Attention Values in the Proposed Dual Attention Mechanism	93
5.	Proposed Cross-Modal Dual Attention	97
5.1.	Cross-Modal Dual Attention Mechanism	97
5.1.1.	Construction of the CMDA Mechanism	98
5.1.2.	Proposed Bi-contextual CMDA Architecture	100
5.2.	Proposed Baselines and Implementation Details	101
5.2.1.	Proposed Baselines for the Evaluation of the CMDA Mechanism	102
5.2.2.	Implementation Details	104
5.3.	Quantitative Results	106
5.3.1.	Effectiveness of the Proposed Cross-Modal Dual Attention Me- chanism	106
5.3.2.	On the Relevance of the Cross-Modal Dual Attention Mechanism	108
5.3.3.	Statistical Significance Analysis	109
5.4.	Analysis of Results	110
5.4.1.	Visualization of Attention Values in the CMDA Mechanism . .	111
5.4.2.	Error Analysis	112
6.	Conclusions and Future Work	115
6.1.	Addressing the Research Questions	115
6.2.	Conclusions	117
6.3.	Future Work	119

Bibliography**121**

List of Figures

1.1. General taxonomy in abusive language phenomena. Figure inspired by Poletto et al. (2021).	2
1.2. Examples of offensive memes. Figure taken from: (Kiela et al., 2020).	6
2.1. General process for supervised text classification.	14
2.2. Taxonomy of the Multimodal machine learning.	14
2.3. Representation of a multi-layer perceptron, Figure inspired by Aggarwal (2018).	17
2.4. Contextual attention vs. self-attention representations, where each h_i represents the encoding of a feature, typically corresponding to a word or token. In contextual attention (left figure), the relevance score α_i of each feature is computed through a dot product with a context vector u_h . In contrast, in self-attention (right figure), the relevance is calculated based on the relationships between elements within the same sequence. Figure inspired by (Yang et al., 2016).	20
2.5. Illustration of the self-attention mechanism, Figure taken from (Vaswani et al., 2017).	20
2.6. Illustration of the multi-head self-attention mechanism, Figure inspired by (Vaswani et al., 2017).	23
2.7. Illustration of the cross-modal attention mechanism. Figure inspired by (Ye et al., 2019).	23
2.8. Illustration of the contextual attention mechanism, Figure inspired by: (Yang et al., 2016).	27
2.9. Example of an RNN unit.	28
2.10. Transformer DNN architecture. Figure taken from Vaswani et al. (2017).	30
2.11. Architecture of the BERT model. Figure taken from Devlin et al. (2019).	34

2.12. Example of the visualization of the Bayesian Wilcoxon signed-rank test when comparing two classifiers. The Figure was generated using the library provided by Benavoli et al. (2014).	44
3.1. General scheme for the detection of AL in memes.	53
4.1. General Visualization of our Proposed Dual Attention Unit.	61
4.2. Extended Visualization of our Proposed Dual Attention Unit, where the \otimes symbol denotes matrix multiplication and the \odot symbol denotes element-wise multiplication.	63
4.3. Adapted architectures for the integration of the proposed DA mechanism, both architectures integrate the proposed DA mechanism at the last encoding level. The left-hand sided architecture is based on the Bi-GRU network, on the other hand, the right-hand sided one is based on Transformer NNs.	64
4.4. Illustration of the Proposed GHA Architecture, this architecture is designed to be applied to any stacked-based encoding architecture (e.g. RNN and Transformer-based architectures).	67
4.5. Illustration of the adapted Gated Multimodal Unit (GMU) for the intermediate representation of the encoding levels in RNN and Transformer-based neural networks. Each GMU layer is applied to an individual word/token/image region, generating an intermediate representation with respect to its L different encoding layers. This Figure is inspired by Arevalo et al. (2020).	68
4.6. Adapted AHA architecture for the evaluation of the proposed GHA architecture, the DA mechanism is used at each encoding level of the architecture, and the output of each DA mechanism is combined with the addition and normalization layer.	69
4.7. Classes distribution of the evaluation datasets for the detection of AL in text, this Figure presents the distribution of all the training sets. .	72
4.8. Classes distribution of the evaluation datasets for the detection of AL in memes, this Figure presents the distribution of all the training sets.	73

4.9. Visualization of the Bayesian Test for comparing: 1) the integration of the DA mechanism in the Bi-GRU, Bi-GRU _S , and BERT architectures (subsections a, b, and c); 2) the AHA architectures vs. the sole use of the DA mechanism over the last encoding layer of the Bi-GRU _S and BERT architectures (subsections d and e); 3) the GHA architecture vs. the AHA architecture using the Bi-GRU _S and BERT architectures (subsections f and g); and 4) the integration of the DA mechanism into the VisualBERT model vs. the fine-tuned one, and the GHA architecture vs. the AHA architecture integrated into the VisualBERT model (subsections h and i).	90
4.10. Visualization of the relevance of the encoding layers using the GHA architecture; the left-hand side heatmap presents the relevance by dataset, on the other hand, the right-hand heatmap illustrates the relevance by type of abusive language.	92
4.11. Visualization of the relevance of the encoding layers by instance samples; the samples were taken from the Davidson and Waseem dataset, which corresponds to an offensive and sexist sample, respectively. The left-hand heatmap presents the offensive instance: “@user your a fucking queer fagot bitch”, while the right-hand heatmap presents the sexist instance: “Im not sexist but bitches cannot drive”.	92
5.1. Proposed Cross-Modal Dual Attention Mechanism.	99
5.2. Proposed Bi-contextual CMDA architecture.	100
5.3. Image captioning example.	101
5.4. Visualization of the Bayesian Test for comparing: 1) the integration of the CMDA mechanism vs. the CMA mechanism (subsection a), 2) the integration of the Bi-contextual architecture with the CMDA mechanism vs. the standalone use of the CMDA mechanism, where the best configurations obtained were compared in both cases (subsection b), and 3) our best approach (the Bi-contextual architecture) vs. the Hate-CLIPper model (subsection c).	110
5.5. Visualization of the attention values for an instance containing hate speech. This example was taken from the HMC dataset (Kiela, Wang, and Cho, 2018).	111

- 5.6. Visualization of the attention values for a non-offensive instance. This example was taken from the HMC dataset (Kiela, Wang, and Cho, 2018).112

List of Tables

2.1.	Interpretation ranges of Kappa values.	40
3.1.	Comparative table of our dual attention mechanism versus various single- and multi-sequence attention mechanism variants, based on self-attention and contextual attention mechanisms.	48
4.1.	Hyperparameters of the RNN-based encoding architectures.	77
4.2.	Hyperparameters of the Transformer-based encoding architecture. . .	78
4.3.	Hyperparameters of the VisualBERT encoding configurations.	80
4.4.	Comparison results from our three baseline architectures, and our proposed Dual Attention mechanism variants in six datasets for the AL detection task in text. The <i>Waseem</i> , <i>Davidson</i> , and <i>Golbeck</i> datasets were evaluated with the weighted-average F_1 score, the <i>SemEval 2019 task 6</i> and <i>HASOC 2019</i> datasets were evaluated using the macro-average F_1 score, finally, the <i>AMI 2018</i> dataset was evaluated using the accuracy. Note that “AM” and “EA” refer to Attention Mechanism and Encoding Architecture, respectively.	82
4.5.	Comparison results of our four baseline architectures and the proposed integration of the Dual Attention mechanism into VisualBERT across three datasets for AL detection in memes. The <i>MAMI</i> and <i>DIMEMEX</i> datasets were evaluated using the macro-average F_1 score, while the <i>HMC</i> dataset was evaluated using accuracy. Note that “AM” and “EA” refer to Attention Mechanism and Encoding Architecture, respectively.	83
4.6.	Comparison results from our two baseline architectures, our proposed GHA architecture, and state-of-the-art approaches in six datasets for the detection of AL in text.	84

4.7. Comparison results from our baseline architecture, our proposed GHA architecture, and state-of-the-art approaches in three datasets for the detection of AL in memes.	86
4.8. Comparison results of the complementarity and diversity of the SA and CA mechanisms contrasted with the performance of the DA mechanism. For evaluation, all AMs were integrated with a Bi-GRU encoding architecture for the analysis of text datasets, while the Visual-BERT model was used for the analysis of meme datasets. The SA, CA, and DA result columns report accuracy.	88
4.9. Bayesian signed-rank test results for each proposed approach. The A and B columns indicate the integration of the proposed DA mechanism, as well as the proposed approaches of their multilevel integration, over the encoding architectures; the ‘-’symbol over the B column, denotes the absence of the DA mechanism.	89
4.10. Top-20 words obtained with the proposed <i>Local Attention Score</i> (Equation 4.5.1) over the abusive class, with the use of the proposed DA mechanism. The words indicated in bold, represent the words contained in the <i>Hatebase</i> lexicon.	94
4.11. Examples of non-offensive words captured with the local attention score, which are used in offensive contexts.	95
4.12. The intersection percentage of the top-50 words with the highest Local Attention scores and the <i>Hatebase</i> lexicon database, (-) indicates the absence of the proposed DA mechanism.	96
5.1. Example of the zero-shot classification scheme used to categorize a non-offensive instance. The image was taken from the MAMI dataset (Fersini et al., 2022).	104
5.2. Hyperparameters of the proposed CMDA mechanisms, assuming a latent adaptation of $\beta \rightarrow \alpha$, where β represents the image modality and α represents the text modality. The description of the hyperparameters includes details of the classification layers.	105

5.3.	Evaluation results of the unimodal baseline architectures and the proposed approaches for assessing the performance of the CMDA mechanism in AL detection in memes. We report the mean and standard deviation over 5 runs for each proposed approach, except for the Gemini 1.5 baseline due to request limitations to the server, and for the SOTA models, as we used the results reported on the respective leaderboards. <i>NOTE: The column “#P” indicates the number of parameters of each approach. The letters I, C, and T refer to the use of the image modality, the captions extracted from the image, and the text extracted from the meme, respectively.</i>	107
5.4.	Comparison of the complementarity and error diversity between the proposed CMDA-based configurations, including the Bi-contextual architecture, and their counterparts based on the CMA mechanism. . .	109
5.5.	Bayesian signed-rank test results for each proposed approach. The A and B columns indicate the integration of the proposed CMDA mechanism variants, as well as the proposed baseline approaches, over the encoding architectures; Bi-C(CMDA) denotes the results of the Bi-contextual architecture Bi-C ($I \rightarrow (C \rightarrow T)$).	110
5.6.	Samples of memes that were incorrectly classified by the proposed CMDA and Bi-contextual architecture. These memes were taken from the DIMEMEX dataset of subtask 1 (Jarquín-Vásquez et al., 2024). . . .	113
5.7.	Samples of memes that were incorrectly classified by the proposed CMDA and Bi-contextual architecture. These memes were taken from the DIMEMEX dataset of subtask 2 (Jarquín-Vásquez et al., 2024). . . .	114

Acknowledgments

This research was carried out thanks to the support provided by the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT), through Scholarship No. 925996.

I wish to express my deepest gratitude to my advisor, Dr. Hugo Jair Escalante Balderas, and my co-advisor, Dr. Manuel Montes y Gómez. Their guidance, dedication, and unwavering commitment to research have been invaluable throughout this process. I am profoundly grateful for their constant encouragement, insightful feedback, and constructive criticism, which significantly contributed to the successful completion of this dissertation.

I also extend my sincere appreciation to the distinguished members of my defense committee: Dr. Luis Villaseñor Pineda, Dr. Ariel Carrasco Ochoa, Dr. María del Pilar Gómez Gil, Dr. Delia Irazú Hernández Farías, and Dr. Viviana Patti. Their valuable observations, thoughtful suggestions, and expert advice were instrumental in refining and improving the quality of this research.

Furthermore, I am grateful to the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), its staff, and all the professors who generously shared their knowledge and expertise. Their support and dedication to academic excellence have played a vital role in my professional and personal development, allowing me to reach this important milestone.

Finally, I would like to thank all those who, directly or indirectly, contributed to this achievement.

Abstract

In recent years, deep neural networks have gained widespread popularity for a variety of unimodal and multimodal classification tasks. Among these, Transformer-based models have emerged as a dominant approach due to their adaptability across diverse tasks through fine-tuning and their outstanding performance in text classification, image analysis, and multimodal tasks involving both text and images. One of the key components of these architectures is the self-attention mechanism, which enables the measurement of relevance among elements within an input sequence. This mechanism is particularly effective in modeling long-range dependencies, making it a cornerstone of modern neural architectures.

In addition to self-attention, the literature has introduced various other attention mechanisms, which can be broadly categorized based on how they compute the similarity between elements in two main branches. Self-attention measures the similarity among elements within the same sequence, while the contextual attention mechanism calculates the similarity of elements with respect to a contextual vector learned during the training process. Despite their utility, these mechanisms have complementary limitations: self-attention disregards the contextual relationships of elements with the global context learned during training, whereas contextual attention neglects internal relationships within the elements of a sequence. These limitations highlight the need for a mechanism that combines the strengths of both approaches.

To address these challenges, this doctoral research proposes the Dual Attention (DA) mechanism, which integrates both contextual and internal relationships within a sequence to create a more comprehensive representation. The DA mechanism was evaluated on the task of abusive language detection in both textual data and memes. This task was selected due to its inherent complexity, requiring both local and global contextual understanding to accurately interpret instances. Abusive language often relies on subtle contextual cues and multimodal signals, making it an ideal testbed

for the proposed mechanism.

The proposed DA mechanism was rigorously tested across multiple datasets for abusive language detection in text and memes, achieving outstanding results in the majority of cases. To further extend its applicability, the mechanism was adapted for scenarios involving pairs of sequences, particularly for the multimodal task of AL detection in memes. This extension, known as Cross-Modal Dual Attention (CMDA), incorporates the relationships between elements of two sequences, such as textual and visual modalities, enhancing the model’s ability to interpret complex multimodal interactions.

The experiments conducted demonstrated the advantages of the proposed DA and CMDA mechanisms across various encoding architectures. Notably, the proposed mechanisms not only achieved state-of-the-art results but also offered significant memory efficiency, being over 1,000 times more memory-efficient than one of the leading vision-and-language models. This efficiency underscores the practical applicability of the mechanisms in real-world scenarios where computational resources are limited. In addition, the proposed mechanisms enable the extraction of relevant elements—such as words or image regions—that are critical for detecting abusive language. This capability provides valuable insights into the decision-making process of the models.

Chapter 1

Introduction

The pervasive integration of social media platforms into the daily lives of billions of users has transformed the landscape of global communication. These platforms facilitate a vast number of social interactions, enabling the creation and dissemination of a wide array of content, as well as the exchange of different opinions and points of view. Unlike traditional media, social media empowers individuals to voice opinions that might otherwise remain unheard. However, this democratization of communication comes with significant challenges. Among these is the proliferation of Abusive Language (AL), a phenomenon that has escalated with the growth of social media usage. This increase in AL is often exacerbated by the anonymity afforded to users and the lack of effective regulation provided by these platforms (Guberman and Hemphill, 2017).

Although there is no global consensus on the definitions of key terms such as Hate Speech (HS) and AL, for the purposes of this research, we adopt the definition of AL as *verbal messages that employ harsh, rude, offensive, and/or insulting words in an inappropriate manner, which may also include profanity and slurs intended to demean the dignity of an individual or group of people* (Cecillon et al., 2019). The term AL is often used as an umbrella expression encompassing a range of related phenomena, from the use of simple obscenities and profanities to threats and severe insults (Kiritchenko and Nejadgholi, 2020).

Recent studies have explored the interrelationships between various phenomena such as HS, offensive language, aggressiveness, abusiveness/toxicity, and other manifestations of hatred targeting specific groups, including misogyny, racism, and homophobia (Poletto et al., 2021). As illustrated in Figure 1.1, AL encompasses these diverse manifestations. Specifically, in this research, we concentrate on the detection of AL in social media.

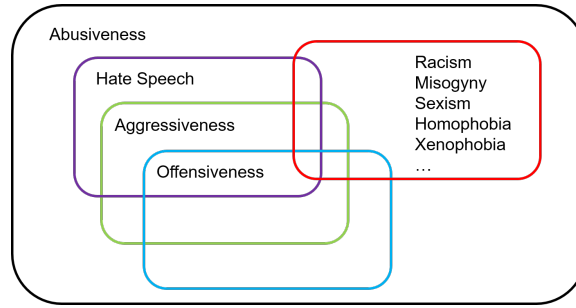


Figure 1.1: General taxonomy in abusive language phenomena. Figure inspired by Poletto et al. (2021).

The widespread dissemination of AL on social media has garnered significant attention from both governments and supplier companies due to its severe social implications (Kumar et al., 2018). On an individual level, AL can cause direct harm to users who are targeted, potentially leading to psychological trauma or, in extreme cases, suicide. On a broader societal level, the prevalence of AL contributes to the deterioration of public discourse, fostering a more polarized and fragmented society (MacAvaney et al., 2019; Naseem et al., 2019).

The task of detecting AL on social media presents substantial challenges. Traditional approaches, such as employing content filters or human moderators, are neither efficient nor scalable given the sheer volume of content generated daily on these platforms. As a result, more sustainable and automated solutions are necessary. In recent years, multiple initiatives have been undertaken to mitigate the proliferation of AL. These efforts include the implementation of content regulations and policies on social media platforms (Corazza et al., 2020), as well as the adoption of Machine Learning (ML) techniques for the automated analysis of social media content (Schmidt and Wiegand, 2017; Wenjie and Arkaitz, 2021).

Regarding the norms and regulations, different countries have implemented restrictions on the dissemination of potentially offensive content. For instance, the European Union, in collaboration with social media platforms such as Facebook¹, Twitter², Microsoft³, and YouTube⁴, has recently signed a code of conduct⁵. This agreement

¹<https://time.com/5739688/facebook-hate-speech-languages/>

²<https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy>

³<https://opensource.microsoft.com/codeofconduct/>

⁴<https://www.youtube.com/howyoutubeworks/policies/community-guidelines/>

⁵http://ec.europa.eu/justice/fundamental-rights/files/hate_speech_code_of_conduct_en.pdf

commits these companies to review the majority of valid notifications for the removal of AL content within 24 hours. While these efforts represent a significant step towards combating online HS and AL, they are not scalable or sustainable in the long term, as they rely heavily on manual moderation and oversight. Moreover, the constant evolution of offensive content presents further challenges to ensuring compliance and effective moderation at scale.

On the other hand, ML approaches to detecting AL have predominantly been addressed from a supervised learning perspective, where most solutions focus on analyzing AL within text-based content (Poletto et al., 2021). Natural Language Processing (NLP) has played a crucial role in this task, with a wide array of methods being proposed, ranging from traditional approaches using bag-of-words (BoW) and classical machine learning classifiers (Fortuna and Nunes, 2018; MacAvaney et al., 2019), to more advanced techniques such as Deep Learning (DL) architectures, Attention Mechanisms (AM), and Transformer-based neural language models. These models, particularly the Transformer-based ones, represent the current state-of-the-art in the AL detection task (Chakrabarty, Gupta, and Muresan, 2019; Mutanga, Naicker, and Olugbara, 2020; Jahan and Oussalah, 2023).

Despite the encouraging results achieved by these models in detecting AL, a significant limitation remains: the vast majority of current approaches have been primarily focused on the analysis of textual information alone. This text-centric approach overlooks the multimodal nature of social media, where content often includes not just text, but also images, audio, and video. As a result, these approaches may fail to capture the full context or intent behind potentially offensive content, leading to low performance in complex, multimodal scenarios.

One of the most prevalent examples of multimodal information found on social media is memes (Kiela et al., 2020). Memes are defined as *the combination of text and an image that together convey a specific meaning (often humorous or ironic), and where the absence of one of these elements (text or image) may result in a different interpretation* (Sharma et al., 2020). Despite its daily use in humorous and ironic publications, the use of memes to transmit AL transcends the social media platforms (Suryawanshi et al., 2020). The detection of AL in memes presents a particularly challenging task, as the interpretation of a meme relies heavily on both its textual and visual components. This inherent complexity makes it difficult for traditional text-based or image-based approaches to fully capture the intended meaning of memes. In

an effort to advance the field, this research focuses on the detection of AL in both text and memes. By addressing the challenges posed by the multimodal nature of memes, we aim to develop a more comprehensive solution to AL detection in social media.

1.1. Motivation and Justification

The use of deep architectures has gained widespread popularity in recent years for various classification tasks (Zhang et al., 2023). In particular, Transformer-based pre-trained models have emerged as the state-of-the-art approach for detecting AL in both unimodal (text) and multimodal (memes) scenarios (Mogadala, Kalimuthu, and Klakow, 2021; MacAvaney et al., 2019; Afridi et al., 2020; Khan et al., 2021; Wenjie and Arkaitz, 2021; Jahan and Oussalah, 2023). The remarkable performance of Transformer Neural Network (TNN)-based architectures can be primarily attributed to the use of the Self-Attention (SA) mechanism (Chaudhari et al., 2021), which plays a pivotal role in capturing the internal relationships between elements within a sequence.

In the context of NLP, SA is particularly effective in capturing the intricate dependencies between each pair of words in a sentence, allowing for a more nuanced and comprehensive representation of textual data. Similarly, in multimodal applications involving both vision and language, SA excels in capturing the relationships between textual components and the corresponding visual regions within a paired image and text input. This capability is especially relevant to the domain of this doctoral research, which focuses on the detection of AL in memes, as it allows the model to effectively represent the internal relationships between the text and visual elements present in the memes.

Given the outstanding results achieved through the SA mechanism in both unimodal and multimodal tasks, this research extends the SA mechanism's capabilities. The goal was to explore how enhancing the SA mechanism, in conjunction with the Contextual Attention (CA) mechanism, can lead to improved performance in the detection of AL in complex multimodal data like memes. By leveraging and extending these attention-based techniques, we developed a more robust, scalable, and efficient approach to addressing the challenges posed by AL detection in social media contexts.

1.2. Problem Statement

The use of Attention Mechanisms (AMs) has gained considerable relevance within DL approaches, primarily due to their ability to enable classification models to focus selectively on a subset of inputs or features, while also effectively modeling long-term dependencies between elements of a sequence (Chaudhari et al., 2021). AMs have been successfully applied across a wide range of DL architectures, consistently delivering state-of-the-art results in various domains. According to (Niu, Zhong, and Yu, 2021), AMs can be classified into two main categories based on how similarity between elements is calculated: 1) Self-Attention mechanisms, and 2) Contextual Attention (CA) mechanisms.

Despite their impressive performance, both SA and CA mechanisms have inherent limitations. On the one hand, CA mechanisms do not account for the internal relationships between elements within a sequence, focusing primarily on the external context, which is represented by patterns learned during the training process. On the other hand, SA mechanisms excel at modeling local dependencies within a sequence but fall short in considering global relationships between elements from different sequences. This oversight can result in the loss of relevant information, particularly in application domains where both local and global contexts are critical for accurate interpretation. Interestingly, these limitations are complementary, suggesting the potential for improvement through their integration.

The integration of SA mechanisms into DL architectures, such as Transformer Neural Networks (TNNs), has proven highly effective in tasks where correct interpretation depends heavily on the internal context of elements within a sequence (i.e., local dependencies) (Kora and Mohammed, 2023; Zhang et al., 2023). However, this type of integration is often insufficient in the context of AL detection, particularly in memes, where both textual and visual modalities are present. The challenge here lies in the need to incorporate not only local dependencies but also global context, which is crucial for correctly interpreting multimodal content. This limitation can lead to the loss of important information necessary to enhance the representations of both image and text.

For instance, in the memes shown in Figure 1.2, global contextual information is essential for accurate interpretation. In the meme on the left, understanding the strong smell of a skunk is key to grasping the irony and abusive nature of the content. To

address this issue, this work introduces the extension of SA mechanisms through the development of novel approaches that integrate CA mechanisms, aiming to combine the strengths of both. This novel integration of SA and CA mechanisms is referred to as Dual Attention (DA) mechanisms. Formally, the DA mechanism is defined as follows:

Let $\mathbf{S} \in \mathbb{R}^{n \times d}$ be the representation obtained from the SA mechanism, where n represents the number of features (e.g., the number of words in a sequence) and d is the embedding dimension. Similarly, let $\mathbf{C} \in \mathbb{R}^{n \times d}$ be the representation obtained from the CA mechanism. The goal of the DA mechanism is to combine both representations, \mathbf{S} and \mathbf{C} , into a unified representation $\mathbf{D} \in \mathbb{R}^{n \times d}$.

Formally, this combination is defined as a differentiable function $f : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$, which merges the local dependencies captured by SA and the global dependencies captured by CA, resulting in a contextually enriched representation:

$$\mathbf{D} = f(\mathbf{S}, \mathbf{C})$$

The function f can be any operation that ensures the combination of both representations, resulting in a high-level representation \mathbf{D} that leverages both local (SA) and global (CA) contexts.



Figure 1.2: Examples of offensive memes. Figure taken from: (Kiela et al., 2020)

1.3. Hypothesis

The integration of SA and CA mechanisms in the form of a novel dual attention mechanism within deep encoding architectures could significantly enhance the performance of AL detection tasks in both textual content and memes. This approach

has the potential to generate a model that is not only modular and scalable but also efficient to train, particularly in terms of the number of parameters to optimize. Compared to the current state-of-the-art pre-trained Vision & Language models, the proposed mechanism aims to reduce the computational complexity while maintaining or even improving detection performance.

1.4. Research Questions

This doctoral research addressed the following key questions:

- 1.- Which fusion approach yields the best performance in the integration of SA and CA mechanisms for the task of AL detection?
- 2.- Which deep learning architectures are best suited to incorporate the proposed attention mechanisms in terms of maximizing performance for detecting AL in both textual data and memes?
- 3.- What are the most significant textual and visual features that contribute to the deep representation of text and images in the context of AL detection?
- 4.- Is the unique integration of textual and visual modalities sufficient for the effective detection of AL in memes, or are additional sources of information required?

1.5. Objectives

The general objective of this doctoral research was:

To develop a novel dual attention mechanism based on the integration of the CA mechanism into the SA mechanism, aimed at extracting both internal and contextual relationships between the elements of a sequence. This mechanism was subsequently evaluated in the detection of AL in text and memes using a variety of DL architectures, with the goal of surpassing the results achieved by traditional and state-of-the-art approaches.

To achieve this general objective, the following specific objectives were proposed:

- 1.- Propose a novel dual attention mechanism based on the integration of SA and CA mechanisms, enabling the extraction of internal and contextual relationships within the elements of a sequence.

- 2.- Integrate the proposed dual attention mechanism into a variety of standard and well-established DL architectures, including pre-trained Transformer models, in both unimodal (text) and multimodal (text and image) scenarios for the detection of AL in text and memes.
- 3.- Extend the dual attention mechanism to a cross-modal approach, aiming to obtain a better alignment of features from the text and image modalities by incorporating contextual information (learned during the training stage) from one modality into the other.
- 4.- Integrate the cross-modal dual attention mechanism into unimodal pre-trained Transformer models, with the aim of enhancing the representation of both text and image modalities for improved AL detection in memes.
- 5.- Evaluate the effectiveness of the proposed dual attention mechanism both qualitatively and quantitatively, using a wide range of AL datasets for text and memes, and assess its performance compared to existing approaches.

1.6. Contributions

In this doctoral thesis, we made the following contributions:

- 1.- A novel attention-based mechanism that improves the alignment of features between the elements of a sequence. This mechanism incorporates the relevance of each element in the sequence with respect to the training task, allowing for a more accurate representation of contextual relationships.
- 2.- A new cross-modal attention mechanism is introduced, enhancing the alignment between textual and visual features. This mechanism improves the model's ability to capture and leverage the intricate interplay between text and images, particularly in scenarios where understanding the context of both modalities is crucial for interpretation.
- 3.- A deeper understanding of the advantages that both unimodal and multimodal attention mechanisms offer in the detection of AL in text and memes. This insight helps elucidate how each modality contributes to the overall performance

and highlights the potential benefits of leveraging multimodal approaches in complex AL detection tasks.

1.7. Overview of the Research and Main Findings

Throughout this doctoral research, three different dual attention mechanisms were proposed. The first approach involved the early fusion of features obtained by applying both SA and CA mechanisms to a sequence of encoded features derived from a Gated Recurrent Unit (GRU) network. In the initial evaluation phase, this mechanism was tested on four different English datasets to detect AL in text. This first mechanism, called the Self-Contextualized Attention (SCA) Mechanism, was presented at the Ninth International Workshop on Natural Language Processing for Social Media in 2021. The results were promising, as the SCA mechanism demonstrated improvements over the independent use of SA and CA mechanisms. Moreover, the SCA mechanism was able to correct some of the errors made by both SA and CA in AL detection. For a detailed explanation of the SCA mechanism, we refer the reader to the corresponding paper (Jarquín-Vásquez, Escalante, and Montes, 2021).

The second dual attention mechanism introduced a transversal combination approach, by integrating input sequences at every stage of the representation to unify both SA and CA representations. Additionally, we extended this dual attention mechanism with a hierarchical perspective to leverage the multi-level feature encoding capabilities of different deep learning architectures, particularly focusing on Recurrent Neural Networks (RNNs) and Transformer-based architectures. The evaluation of this second attention mechanism was conducted on six English-language datasets for AL detection in text and three multimodal datasets for AL detection in memes—two in English and one in Spanish. The results were encouraging, achieving state-of-the-art performance on four out of the six text-based datasets and yielding consistently improved results across all three meme-based datasets. The incorporation of the dual and hierarchical attention mechanisms led to noticeable performance gains, demonstrating their effectiveness across both textual and multimodal contexts.

Finally, the third dual attention mechanism builds on the second by incorporating cross-modal attention, enabling the combination of different modalities. This mechanism is referred to throughout this research as the Cross-Modal Dual Attention mechanism. It was evaluated using the same three datasets for AL detection in me-

mes that were used to assess the second mechanism. The results were promising: the CMDA mechanism outperformed state-of-the-art approaches on one of the three datasets while achieving competitive results on the remaining two, all while maintaining a lower number of parameters compared to other leading models. A comprehensive description of the second and third attention mechanisms is provided in Chapters 4 and 5, respectively.

1.8. Scope and Limitations

This research focused on the design, implementation, and evaluation of the proposed dual attention mechanisms, along with their adaptation into different DL architectures. The effectiveness of the proposed contributions was evaluated in the task of detecting AL in both text and memes, using various publicly available English datasets as well as collections from different evaluation campaigns. Given the nature of these evaluation datasets, where labels are manually annotated, there is a potential for inherent social biases in the annotators' judgments. Consequently, the various configurations of the proposed models could inadvertently learn and propagate these biases, which might lead to errors when applied to data of a different nature or from diverse contexts.

Additionally, the proposed dual attention mechanisms were integrated and evaluated using a specific set of pre-trained Transformer models. It is important to note that integrating these mechanisms into alternative pre-trained models may yield varying performance levels in the task of detecting AL, as different models have different capacities for feature extraction and representation.

1.9. Publications Derived from this Doctoral Research

Below are the publications that have been derived from this doctoral research, which reflect the key findings and contributions made throughout this work.

- 1.- Horacio Jarquín-Vásquez, Hugo Jair Escalante, and Manuel Montes. (2021). Self-Contextualized Attention for Abusive Language Identification. In Proceedings of the Ninth International Workshop on Natural Language Processing for

Social Media, pages 103–112, Online. Association for Computational Linguistics.

- 2.- Horacio Jarquín-Vásquez, Hugo Jair Escalante, and Manuel Montes. (2023). Improving the Identification of Abusive Language Through Careful Design of Pre-training Tasks. In: Rodríguez-González, A.Y, Pérez-Espinosa, H, Martínez-Trinidad, J.F, Carrasco-Ochoa, J.A, Olvera-López, J.A. (eds) Pattern Recognition. MCPR 2023. Lecture Notes in Computer Science, vol 13902. Springer, Cham.
- 3.- Horacio Jarquín-Vásquez, Hugo Jair Escalante, Manuel Montes-y-Gómez. (2024). Enhancing abusive language detection: A domain-adapted approach leveraging BERT pre-training tasks, Pattern Recognition Letters, ISSN 0167-8655, <https://doi.org/10.1016/j.patrec.2024.05.007>.
- 4.- Horacio Jarquín-Vásquez, Itzel Tlelo-Coyotecatl, Marco Casavantes, Delia Irazú Hernández-Farías, Hugo Jair Escalante, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez. (2024). Overview of DIMEMEX at IberLEF 2024: Detection of Inappropriate Memes from Mexico. *Procesamiento de Lenguaje Natural*, vol. 72.
- 5.- Horacio Jarquin-Vasquez, Hugo Jair Escalante, Manuel Montes-y-Gomez and Fabio A. Gonzalez. (2024). GHA: a Gated Hierarchical Attention Mechanism for the Detection of Abusive Language in Social Media. In *IEEE Transactions on Affective Computing*, pp. 1-14, <https://10.1109/TAFFC.2024.3483010>.

Additionally, the publications in which active collaboration took place throughout this doctoral research are listed.

- 1.- Flor Miriam Plaza-del-Arco, Marco Casavantes, Hugo Jair Escalante, M. Teresa Martín-Valdivia, Arturo Montejo-Ráez, Manuel Montes-y-Gómez, Horacio Jarquín-Vásquez, Luis Villaseñor-Pineda. (2021). Overview of MeOffendEs at IberLEF 2021: Offensive Language Detection in Spanish Variants. *Procesamiento del Lenguaje Natural*, vol. 67.
- 2.- Horacio Jarquín-Vásquez, Delia Irazú Hernández-Farías, Luis Joaquín Arellano, Hugo Jair Escalante, Luis Villaseñor-Pineda, Manuel Montes-y-Gómez, Fernando Sanchez-Vega. (2023). Overview of DA-VINCIS at IberLEF 2023: Detection

of Aggressive and Violent Incidents from Social Media in Spanish. *Procesamiento del Lenguaje Natural*, Vol. 71.

1.10. Organization of the Document

The remainder of this document is structured as follows: Chapter 2 presents and describes in detail the background concepts and techniques necessary for understanding this research. Chapter 3 introduces the related work on attention mechanisms and the detection of AL in text and memes, and highlights the key differences between this doctoral research and prior studies. Chapter 4 presents the dual attention mechanism, along with its adaptations to various DL architectures, as well as the respective results. Chapter 5 introduces the cross-modal dual attention mechanism, its adapted architectures, and the corresponding results. Finally, Chapter 6 addresses the research questions, and presents the conclusions and future work.

Chapter 2

Background

This section provides an overview of the various techniques and core concepts essential for understanding the key ideas presented in this doctoral thesis. Given that the scope of this thesis encompasses both unimodal (text) and bimodal (text and image) representations, the section is structured as follows: first, an overview of Text Classification (TC) and Multimodal Machine Learning (MML) is presented. Then, the concept of Deep Learning (DL) is introduced, along with some of the main underlying ideas and relevant Deep Neural Network (DNN) architectures used for representing and classifying text and image modalities. Finally, the different evaluation metrics and statistical significance tests employed to assess the various approaches proposed throughout this doctoral thesis are discussed.

2.1. Text Classification

TC is the process of assigning categories or labels to a text or document based on its content. TC can be used to categorize and structure a set of documents by topics, languages, or conversations. It has a broad range of applications, including sentiment analysis, intent detection, and information filtering (Aggarwal and Zhai, 2012).

Formally, given a text document $x \in \mathcal{X}$ and a predefined set of categories $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$, the task of text classification is to find a function $f : \mathcal{X} \rightarrow \mathcal{C}$ that maps each document x to a category $c \in \mathcal{C}$. This function f can be constructed through different methodologies, most commonly using machine learning techniques that learn patterns from labeled training data.

TC can be performed in two ways: i) automatically, where machine learning algorithms are used to classify text more quickly and cost-effectively, and ii) manually,

where a human annotator reviews the text and categorizes it based on their interpretation of the content (Zhou, 2020). TC has become a crucial tool in business, enabling companies to derive insights from data and automate the analysis of various processes. Figure 2.1 illustrates the general process for supervised TC. In this process, the model receives a set of documents and their corresponding categories as input. The model is then trained using a machine learning algorithm. Once trained, the model is used to classify new documents, producing the assigned categories as output. For a more detailed reference on TC, we direct the reader to the following survey (Li et al., 2022).

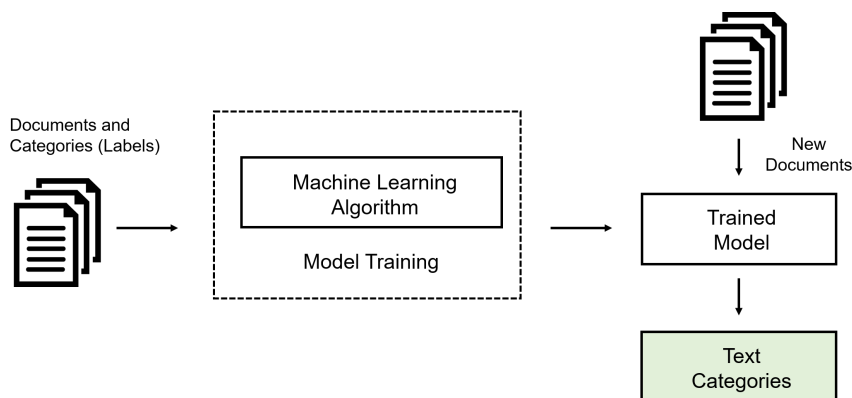


Figure 2.1: General process for supervised text classification.

2.2. Multimodal Machine Learning

MML aims to build models that can process and relate information from multiple modalities (e.g., text, image, video, audio). According to Baltrusaitis, Ahuja, and Morency (2019) there are five core technical challenges surrounding the MML, Figure 2.2 presents this taxonomy.

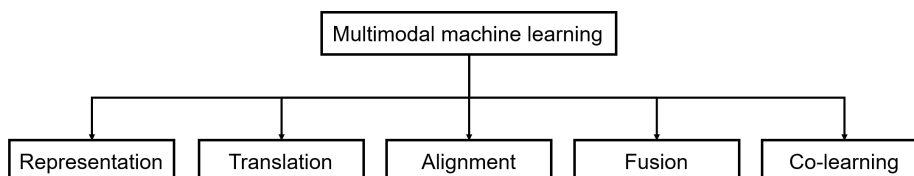


Figure 2.2: Taxonomy of the Multimodal machine learning.

The following outlines the five core technical challenges in MML:

- **Representation:** The first fundamental challenge is learning how to effectively represent and summarize multimodal data in a manner that leverages the complementarity and redundancy inherent in multiple modalities. Due to the heterogeneity of multimodal data where different modalities have distinct characteristics, it is particularly difficult to design representations that can accommodate this diversity while capturing meaningful patterns and relationships between the modalities. Successful representation should capture shared information across modalities while preserving modality-specific details.
- **Translation:** The second challenge involves translating or mapping data from one modality to another. For example, translating visual information into textual descriptions (image captioning) or converting speech into text (speech recognition). This task requires not only understanding the inherent features of each modality but also learning how information from one can be faithfully rendered into another, preserving its meaning and context while adapting to the target modality's format and constraints.
- **Alignment:** The third challenge is to establish direct relations between the elements from two or more different modalities. This involves determining how components of one modality (such as words in a sentence) correspond to components in another modality (such as regions in an image or frames in a video). Alignment is crucial for tasks like image-text retrieval or video-text synchronization, where the correct matching between elements of different modalities plays a central role in the performance of the system.
- **Fusion:** The fourth challenge focuses on the integration of information from multiple modalities to make a unified prediction. Fusion requires the effective combination of multimodal data streams to create a more robust and informed decision-making process. Various fusion strategies can be employed, such as early fusion (combining data at the feature level), late fusion (combining data at the decision level), or hybrid approaches that blend both strategies.
- **Co-learning:** The fifth challenge is co-learning, which addresses the transfer of knowledge across modalities. This encompasses the joint learning of representations and predictive models where knowledge obtained from one modality can inform and enhance the learning in another. Co-learning strategies, such

as cross-modal supervision or multitask learning, allow the system to exploit shared structure across modalities, enabling improved generalization, especially in cases where one modality is underrepresented or noisy.

In this research, we focus on the fusion and alignment of the vision and language modalities. Our approach is grounded in identifying relationships between elements from both modalities by employing an attention-based DL approach. The goal is to effectively capture the interactions between visual and textual features, leveraging the attention mechanism to highlight the most relevant components across modalities.

2.3. Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the way neurons in the human brain process and transmit information (Bengio, 2009). Each neuron in these models processes incoming signals and passes an output to subsequent layers, enabling the network to learn complex patterns and capture intricate relationships between inputs and outputs (Alzubaidi et al., 2021). Over the years, ANNs have evolved from simple perceptron-based systems to sophisticated architectures that drive modern breakthroughs in a wide variety of tasks (Li and Dong, 2014).

One of the most common types of ANNs is the Multi-Layer Perceptron (MLP). According to Aggarwal (2018), a typical MLP consists of three main components: an input layer, one or more hidden layers, and an output layer. Each layer in an MLP is made up of interconnected neurons, which pass data from one layer to the next. The connections between neurons are governed by weights that are adjusted during training to optimize the network's performance. Figure 2.3 illustrates the architecture of a typical MLP, showing how multiple layers interact to process and transform the input data, ultimately producing a final prediction or decision.

A neural network architecture consists of a multi-layer representation that applies activation functions to perform non-linear transformations of the inputs which can be described as follows:

$$f_l^{W,b} = f_l\left(\sum_{j=1}^{N_l} W_{lj}X_j + b_l\right), 1 \leq l \leq L \quad (2.3.1)$$

Where the number of hidden units is given by N_l . The predictor is in charge of modeling a high-dimensional mapping F through the composition of functions, as can

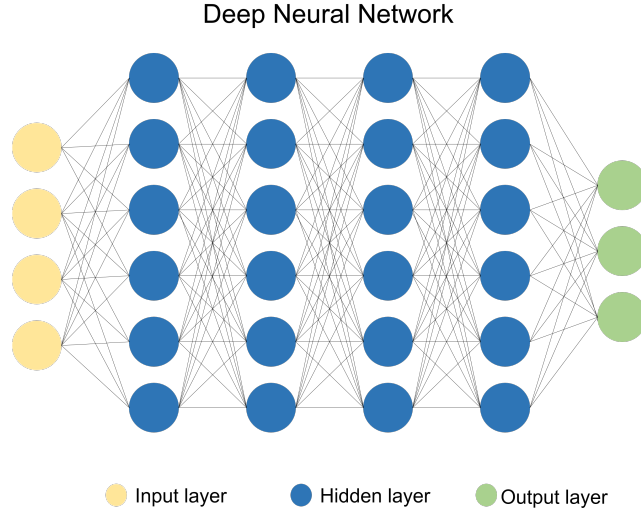


Figure 2.3: Representation of a multi-layer perceptron, Figure inspired by Aggarwal (2018).

be defined in Equation 2.3.2.

$$Y(X) = F(X) = (f_1^{W_1, b_1} \circ \dots \circ f_L^{W_L, b_L}) \quad (2.3.2)$$

The final output is the answer of Y , this can be categorical or numerical. The explicit structure of the prediction rule is:

$$\begin{aligned}
 Z^{(1)} &= f^{(1)}(W^{(0)}X + b^{(0)}), \\
 Z^{(2)} &= f^{(2)}(W^{(1)}Z^{(1)} + b^{(1)}), \\
 &\dots \\
 Z^{(L)} &= f^{(L)}(W^{(L-1)}Z^{(L-1)} + b^{(L-1)}), \\
 Y^{(X)} &= W^{(L)}Z^{(L)} + b^{(L)}
 \end{aligned} \quad (2.3.3)$$

Where $Z^{(L)}$ is defined as the L -th layer, $W^{(L)}$ is the weight matrix and $b^{(L)}$ is the bias. $Z^{(L)}$ contains the extracted hidden features, in other words, the deep approach uses hierarchical predictors that comprise a series of non-linear transformations in L applied to X . Each of the L transformations refers to a layer where the original input is X , the output of the first transformation is the input of the second layer and so on until the output Y as the layer $(L + 1)$. $l \in \{1, \dots, L\}$ is used to index the layers,

which are called hidden layers. The number of layers L represents the depth of the deep architecture.

2.4. Deep Learning

DL is a subfield of ML that focuses on learning hierarchical models with multiple layers of representation and abstraction from raw input data, such as images, audio, and text. The fundamental idea behind DL is to automatically discover the underlying structures and patterns in data by progressively extracting higher-level features through successive layers of neural networks.

Over time, the family of DL methods has grown substantially, encompassing a wide range of algorithms designed for both supervised and unsupervised learning. These methods can address a variety of tasks, from classification and regression to more complex challenges such as image generation, language translation, and reinforcement learning. The increasing diversity of DL techniques has made them powerful tools in fields as varied as Computer Vision (CV), NLP, speech recognition, and beyond (Abdel-Jaber et al., 2022). As a result, DL has become a cornerstone of modern artificial intelligence, enabling breakthroughs in many applications that were previously unattainable.

Deep Neural Networks (DNNs) are a prominent example of DL models. They can be understood as an extension of traditional multi-layer perceptrons, where the architecture consists of multiple hidden layers, enabling the network to solve complex problems that are not linearly separable. Unlike simpler neural networks, DNNs leverage these additional layers to capture deeper patterns and hierarchies within the data, making them highly effective for tasks that involve high-dimensional, non-linear relationships.

Several notable approaches have emerged in deep learning, including the use of attention mechanisms, CNNs, RNNs, Auto-Encoders (AEs), Deep Belief Networks (DBNs), Generative Adversarial Networks (GANs), Deep Reinforcement Learning (DRL), and Transformer Neural Networks (TNNs). For more details on these and other deep learning architectures, refer to Alom et al. (2019). In this research, we specifically focus on extending AMs by utilizing architectures based on Transformer and RNN models. These architectures are applied to both unimodal and bimodal classification tasks, aiming to improve the performance of models by leveraging the

power of attention in scenarios involving a single modality or the combination of multiple modalities.

2.4.1. Attention Mechanisms

One of the most widely used approaches in deep learning is the application of attention mechanisms (AMs). The core idea behind AMs is to equip classification models with the ability to focus selectively on a subset of inputs or features, thereby prioritizing features based on their relevance to the context. This selective attention allows the model to better handle the varying importance of different features within the data. Due to their exceptional performance in numerous NLP and CV tasks, several attention mechanisms have been developed in recent years (Chaudhari et al., 2021). These mechanisms can generally be divided into two main categories based on how they compute similarity: Self-Attention (SA) (Vaswani et al., 2017) and Contextual Attention (CA) (Yang et al., 2016).

SA focuses on capturing the relationships among features within the same sequence, making it highly effective for modeling long-range dependencies in sequential data. In contrast, CA selectively emphasizes features with respect to an external query vector, which is dynamically adjusted based on the specific training task. The more important the feature is in determining the answer to that query, the more focus it is given, allowing the model to effectively weigh contextual relevance. Figure 2.4 illustrates these two approaches.

In recent years, several variants of the SA mechanism have been proposed, ranging from multi-head perspectives to its adaptation for multimodal feature alignment (Niu, Zhong, and Yu, 2021). In the following subsections, we will delve deeper into the foundational principles behind the attention mechanisms used in this research. We begin with the Self-Attention mechanism, exploring its multi-head extension and cross-modal adaptation. Finally, the last subsection will cover the Contextual Attention mechanism in detail.

Self-Attention Mechanism

The *self-attention mechanism*, introduced by Vaswani et al. (2017) in their paper “*Attention is All You Need*”, is a core component of modern DL architectures, particularly in the Transformer model. It has proven to be highly effective in tasks

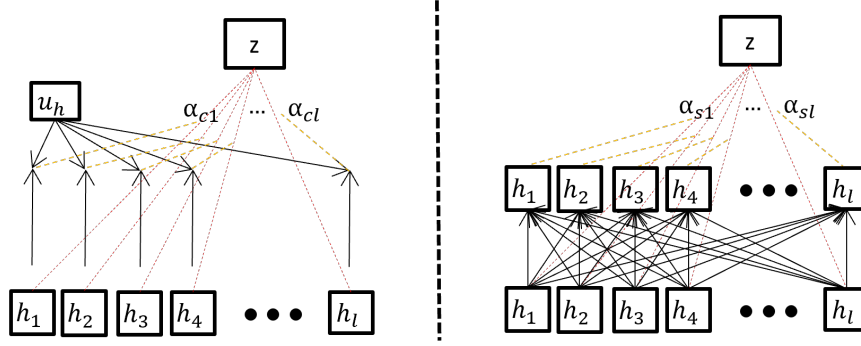


Figure 2.4: Contextual attention vs. self-attention representations, where each h_i represents the encoding of a feature, typically corresponding to a word or token. In contextual attention (left figure), the relevance score α_i of each feature is computed through a dot product with a context vector u_h . In contrast, in self-attention (right figure), the relevance is calculated based on the relationships between elements within the same sequence. Figure inspired by (Yang et al., 2016).

involving NLP and CV, due to its ability to capture long-range dependencies within sequences. The self-attention mechanism is based on the idea of allowing each element in a sequence to focus on, or “attend” to, other elements of the sequence. Figure 2.5 illustrates the self-attention process, below, we describe this process in detail.

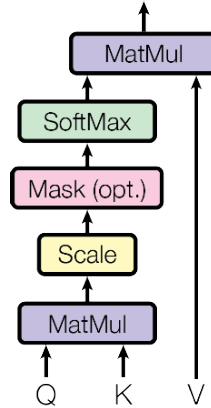


Figure 2.5: Illustration of the self-attention mechanism, Figure taken from (Vaswani et al., 2017).

Step 1: Input Sequence Representation

The process begins with an input sequence consisting of multiple elements, such as words in a sentence or pixels in an image. Each element is represented as a vector, resulting in a matrix $X \in \mathbb{R}^{n \times d}$, where n is the number of elements in the sequence

and d is the dimensionality of the feature vector for each element.

For instance, in NLP, X might represent a sentence with each word encoded as a vector (i.e., word embeddings), and in CV, it might represent an image with each pixel encoded as a feature vector.

Step 2: Linear Projections to Query, Key, and Value

The SA mechanism requires three sets of vectors for each element in the sequence: *Query* (Q), *Key* (K), and *Value* (V). These vectors are derived by linearly projecting the input matrix X into three different subspaces using learned weight matrices W_Q , W_K , and W_V . Mathematically, this can be expressed as:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \quad (2.4.1)$$

Here, $W_Q \in \mathbb{R}^{d \times d_q}$, $W_K \in \mathbb{R}^{d \times d_k}$, and $W_V \in \mathbb{R}^{d \times d_v}$ are the learned weight matrices for the Query, Key, and Value, respectively. These projections allow the model to represent the input data in three different spaces that will be used to compute attention scores.

Step 3: Computation of Attention Scores

Once the Query, Key, and Value vectors are obtained, the next step is to compute the *attention scores*, which measure the similarity between the Query and the Key vectors. These scores indicate how much focus each element should place on other elements within the sequence. The similarity between the Query and Key vectors is calculated using the dot product, followed by a scaling factor to ensure numerical stability. This is expressed as:

$$\text{Attention Scores} = \frac{QK^T}{\sqrt{d_k}} \quad (2.4.2)$$

Here, d_k is the dimensionality of the Key vectors, and the scaling factor $\frac{1}{\sqrt{d_k}}$ is used to prevent excessively large values in the dot product, which could otherwise destabilize the softmax function in the following step.

Step 4: Softmax Normalization

The attention scores are then passed through a *softmax function* to normalize them into probabilities. This ensures that the attention weights are non-negative and sum to 1. The softmax operation converts the raw attention scores into a distribution over the different elements of the sequence:

$$\text{Normalized Attention Scores} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (2.4.3)$$

Step 5: Weighted Sum of Values

Once the attention weights are computed, the next step is to compute the *weighted sum* of the Value vectors. This is done by multiplying the attention weights with the corresponding Value vectors V . The resulting matrix is the output of the self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.4.4)$$

This operation effectively allows each element in the sequence to aggregate information from other elements based on the computed attention weights. In other words, the model dynamically focuses on the most relevant parts of the sequence when generating the output for each element.

Multi-Head Attention

To enhance the model's capacity to capture diverse relationships between sequence elements, the self-attention mechanism is often implemented as *multi-head attention*. In this approach, the self-attention mechanism is applied multiple times in parallel, each with a different set of learned weight matrices. The outputs from each “head” are concatenated and then projected back into the original feature space. This process enables the model to gather a more comprehensive understanding of the input sequence by considering multiple perspectives simultaneously. The multi-head attention mechanism is formalized in Equation 2.4.5.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \quad (2.4.5)$$

Here, W_O is the final learned projection matrix, and each head independently captures different types of relationships between the sequence elements. Multi-head attention improves the model's ability to attend to information from different subspaces and perspectives. Figure 2.6 illustrates the structure of the multi-head self-attention mechanism.

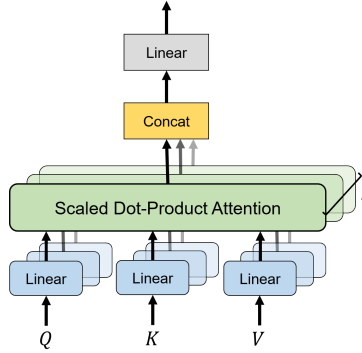


Figure 2.6: Illustration of the multi-head self-attention mechanism, Figure inspired by (Vaswani et al., 2017).

Cross-Modal Attention Mechanism

The *cross-modal attention mechanism* was introduced by Ye et al. (2019). This mechanism extends the principles of self-attention to scenarios where information from different modalities, such as text and images, must be integrated. While self-attention allows elements within the same modality to attend to one another, cross-modal attention facilitates the interaction between elements from two or more distinct modalities. This mechanism is particularly effective in tasks where multimodal data is used, such as image captioning, visual question answering, or multimodal retrieval. Below, we describe the cross-modal attention mechanism step by step. Figure 2.7 illustrates the cross-modal attention process, below, we describe this process in detail.

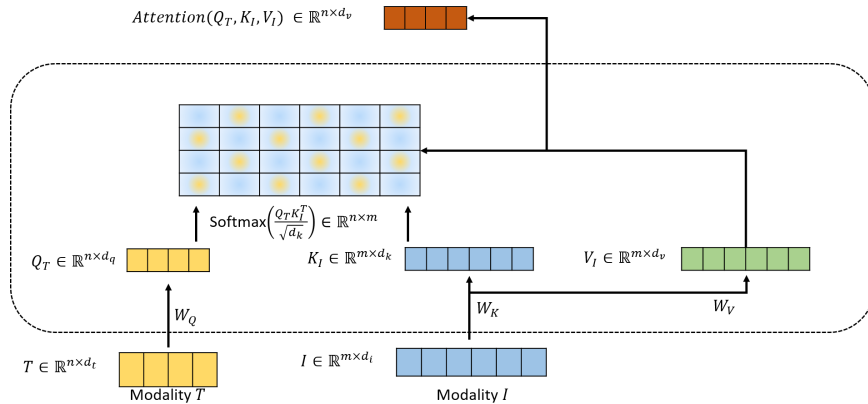


Figure 2.7: Illustration of the cross-modal attention mechanism. Figure inspired by (Ye et al., 2019).

Step 1: Input Modality Representations

The process begins with two different input modalities, such as text and images. Each modality is represented as a sequence of feature vectors. For example, let $T \in \mathbb{R}^{n \times d_t}$ represent the sequence of text features (with n as the number of words and d_t the dimensionality of the word embeddings), and let $I \in \mathbb{R}^{m \times d_i}$ represent the sequence of image features (with m as the number of visual regions and d_i the dimensionality of the image embeddings). The goal of cross-modal attention is to model the relationships between these two sets of representations.

Step 2: Linear Projections to Query, Key, and Value

Similar to the self-attention mechanism, cross-modal attention requires projecting the inputs into Query (Q), Key (K), and Value (V) vectors. However, in cross-modal attention, the Query typically comes from one modality (e.g., text), while the Key and Value come from the other modality (e.g., image). Mathematically, this can be expressed as:

$$Q_T = TW_Q, \quad K_I = IW_K, \quad V_I = IW_V \quad (2.4.6)$$

where $W_Q \in \mathbb{R}^{d_t \times d_q}$, $W_K \in \mathbb{R}^{d_i \times d_k}$, and $W_V \in \mathbb{R}^{d_i \times d_v}$ are learned weight matrices that project the text features into Queries and the image features into Keys and Values, respectively.

Step 3: Attention Score Calculation

The next step is to compute the attention scores, which measure the relevance between elements in the Query modality (text) and elements in the Key modality (image). These scores are computed using the dot product between the Query vectors and the Key vectors, scaled by the dimensionality of the Key vectors to ensure numerical stability:

$$\text{Attention Scores} = \frac{Q_T K_I^T}{\sqrt{d_k}} \quad (2.4.7)$$

These attention scores indicate how much focus each text element should place on each image element based on their computed similarity.

Step 4: Softmax Normalization and Weighted Sum of Values

The attention scores are passed through a softmax function to normalize them into probabilities:

$$\text{softmax}\left(\frac{Q_T K_I^T}{\sqrt{d_k}}\right) \quad (2.4.8)$$

Once the attention weights are computed, they are used to perform a weighted sum over the Values, which originate from the image modality. This is achieved by multiplying the attention weights with the corresponding value vectors V_I . As a result, a new representation is generated for each element in the text modality, enriched with relevant information from the image modality. This process is formalized in Equation 2.4.9.

$$\text{Attention}(Q_T, K_I, V_I) = \text{softmax}\left(\frac{Q_T K_I^T}{\sqrt{d_k}}\right) V_I \quad (2.4.9)$$

Contextual Attention Mechanism

The *contextual attention mechanism* was introduced by Yang et al. (2016) in their proposal of a hierarchical attention network for document classification. This network operates at two different levels: the first level extracts the importance of individual words with respect to sentences, and the second level extracts the importance of sentences with respect to the entire document. Since this research does not employ a hierarchical attention approach, we will focus on describing the first level of the contextual attention mechanism in detail.

Step 1: Encoding the Sequence

Before extracting the importance of elements in a sequence using contextual attention, it is crucial to first apply a sequence encoding process to capture the context of each element of the sequence. This can be achieved using either an RNN architecture or a Transformer-based neural network. For each word x_i in the sequence, the encoding network generates a hidden state h_i . This hidden state is then passed through a fully connected layer (or multilayer perceptron) to produce a hidden representation u_i for the word. This process is applied to all elements of the sequence, as formalized in Equation 2.4.10.

$$u_i = \tanh(W_h \cdot h_i + b_h) \quad (2.4.10)$$

Here, W_h and b_h are learnable parameters of the neural network. The hidden representation u_i captures a non-linear transformation of the hidden state h_i , which

provides a richer representation of the word in context.

Step 2: Calculating Word Importance

Once the hidden representation u_i is computed for each word, the next step is to calculate the importance of each word in the sequence. This is done by measuring the similarity between the word's representation u_i and a context vector u_c , using a dot product. The context vector serves as a global indicator of word importance in the text. The importance score is then normalized using the softmax function to produce the attention weight α_i . This process is formalized in Equation 2.4.11.

$$\alpha_i = \frac{\exp(u_i^T \cdot u_c)}{\sum_j \exp(u_j^T \cdot u_c)} \quad (2.4.11)$$

The context vector u_c is randomly initialized and its parameters are learned during the training process. It acts as a global reference vector that helps the model determine which words or features in a document are more relevant to the overall task. During training, u_c is learned through an optimization process alongside the model's other parameters using backpropagation. As the model learns, u_c is gradually fine-tuned to improve its ability to identify relevant patterns in the data. The backpropagation process updates the values of u_c , optimizing it so that the model can effectively highlight the most important words based on the downstream task.

Step 3: Calculating the Overall Message Representation

After calculating the attention weights α_i for each word, the final step is to compute a weighted sum of the encoded-word representations h_i , which results in a general representation of the message, denoted by z . This weighted sum allows the network to focus on the most important words in the text, providing a more meaningful global representation of the sequence. This is formalized in Equation 2.4.12.

$$z = \sum_j \alpha_j h_j \quad (2.4.12)$$

Here, z represents the overall context-aware representation of the sequence, which is a weighted combination of the hidden states h_j produced by the encoding architecture, and the attention weights.

Figure 2.8 illustrates the structure of the contextual attention mechanism based on an encoding sequence. The mechanism dynamically assigns importance to different words within a sentence, ultimately producing a contextually enhanced representation that captures the most relevant information for text classification.

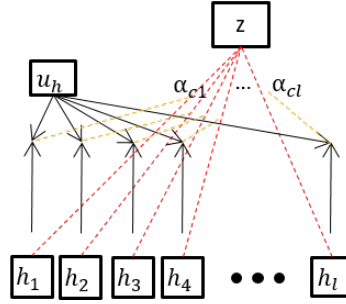


Figure 2.8: Illustration of the contextual attention mechanism, Figure inspired by: (Yang et al., 2016).

2.4.2. Recurrent Neural Network

A *Recurrent Neural Network* (RNN) is a class of ANNs with connections between nodes form either directed or undirected graphs along a temporal sequence (Alzubaidi et al., 2021). This architecture enables the network to exhibit dynamic temporal behavior, making it particularly suitable for processing sequential data. Unlike traditional feed-forward neural networks, RNNs leverage their internal state (also referred to as memory) to process variable-length sequences of inputs (Yu et al., 2019a). By maintaining information from previous inputs, RNNs are especially effective in tasks that require sequence analysis, as they can extract contextual information by defining dependencies across various time steps.

RNNs preserve sequential information within the hidden states of the network, which influences the processing of each new input in the sequence. This allows the network to capture correlations between events that may be separated by time. Similar to how human memory influences behavior without relying on all available information, the hidden states in RNNs carry information that affects decision-making without fully exposing the learned knowledge at each step. The process of preserving memory within these networks is represented mathematically by the following equation:

$$h_t = \phi(Wx_t + Uh_{t-1}) \quad (2.4.13)$$

Here, h_t is the hidden state at time step t , x_t represents the input at the same time step, and W and U are weight matrices that determine how the input and previous hidden state interact. The hidden state from the previous time step h_{t-1} is multiplied

by the matrix U , and this product is added to the weighted input. The function ϕ is typically a non-linear activation function (e.g., tanh or ReLU) that introduces non-linearity into the model. Figure 2.9 illustrates a simple example of a recurrent neural network unit.

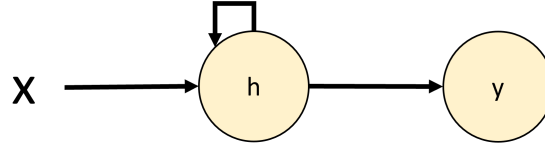


Figure 2.9: Example of an RNN unit.

While RNNs are highly effective at capturing dynamic dependencies in sequential data, they face challenges when dealing with long sequences. Specifically, the issue of vanishing gradients arises during backpropagation, where the gradients shrink with each time step and eventually vanish after many steps, making it difficult to maintain long-term dependencies (LeCun, Bengio, and Hinton, 2015). To address this limitation, specialized architectures that incorporate explicit memory mechanisms were developed. The two most prominent architectures are the *Long Short-Term Memory* (LSTM) networks and the *Gated Recurrent Unit* (GRU) networks (Asakawa, 2016). Both of these neural network variants utilize specialized hidden units, often referred to as memory cells, that are capable of learning to retain information over extended periods.

LSTM and GRU networks introduce memory cells that can regulate the flow of information through a gating mechanism. These cells have a self-loop at the next time step, which allows them to carry forward their state and accumulate new information. Additionally, the memory cells feature multiplicative gates that learn to decide whether to retain or discard information from the memory. This enables the network to selectively preserve important information while discarding irrelevant details. The gated structure helps mitigate the vanishing gradient problem, allowing LSTM and GRU networks to capture long-term dependencies in the data (Yu et al., 2019a).

For more comprehensive details on the structure and operation of LSTM and GRU networks, we refer the reader to the foundational works of (Hochreiter and Schmidhuber, 1997; Bahdanau, Cho, and Bengio, 2015).

2.4.3. Transformer Deep Neural Network

The *Transformer Neural Network* (TNN) is a neural network architecture built upon the self-attention mechanism, forgoing the traditional use of recurrence and convolutions typically employed in sequence modeling. The Transformer was introduced by Vaswani et al. (2017) in the context of sequence-to-sequence learning, particularly in the Neural Machine Translation (NMT) task. Prior to the Transformer, RNNs were the dominant architecture in sequence modeling tasks, especially within the Encoder-Decoder framework for NMT. However, subsequent research (Devlin et al., 2019; Brown et al., 2020; Cohen and Gokaslan, 2020) demonstrated that the Transformer architecture not only surpassed RNNs in NMT but also improved performance across a variety of sequence-related tasks, including sentence classification and other NLP tasks.

The Transformer model comprises two primary components: the *encoder* and the *decoder*, both of which are composed of identical layers that can be stacked N_x times. Figure 2.10 provides a visual representation of the Transformer architecture, illustrating the encoder and decoder stacks. Notably, both the encoder and decoder share the same number of layers N_x .

Encoder Process

The encoder in the Transformer architecture plays a crucial role in processing the input sequence and transforming it into a continuous representation that captures the contextual information across the entire sequence. The encoder is composed of a stack of N_x identical layers, each containing two essential sub-layers: a multi-head self-attention mechanism and a position-wise feed-forward network. Below, we describe these components in detail:

1. Input Embedding and Positional Encoding: Before the input sequence enters the encoder, each token in the sequence is mapped to a fixed-dimensional vector via an embedding layer. However, unlike RNNs and CNNs, the Transformer does not inherently capture the sequential order of the tokens. To address this, the Transformer adds *positional encodings* to the token embeddings. The positional encoding vector, denoted as PE , is designed to inject information about the position of each token in the sequence.

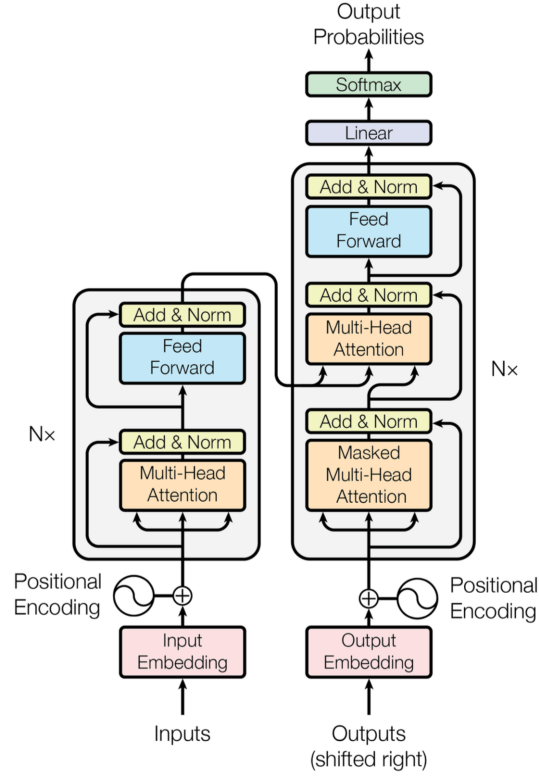


Figure 2.10: Transformer DNN architecture. Figure taken from Vaswani et al. (2017).

As described in Vaswani et al. (2017), the positional encoding is computed using sine and cosine functions of different frequencies:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \quad (2.4.14)$$

Here, pos is the position of the token in the sequence, i is the dimension index, and d is the dimensionality of the embedding. These encodings are added element-wise to the token embeddings, ensuring that each token now has a unique representation that reflects both its content and its position within the sequence.

2. Multi-Head Self-Attention Mechanism: The first sub-layer of the encoder is the multi-head self-attention mechanism, which allows the model to attend to all other tokens in the sequence when processing each token. Given a sequence of token embeddings, the multi-head attention mechanism first generates Query (Q), Key (K), and Value (V) vectors for each token by applying learned weight matrices.

For each token, attention scores are computed between the Query vector of that token and the Key vectors of all tokens in the sequence, as described previously. The attention scores are used to weight the Value vectors, and the final output is a weighted sum of these Value vectors. This enables the model to capture contextual information by focusing on the most relevant parts of the sequence for each token. The multi-head attention mechanism allows the model to attend to different aspects of the sequence in parallel, which improves its ability to capture complex dependencies.

The output of the multi-head attention sub-layer is passed through a residual connection, followed by a normalization layer, as shown in Equation 2.4.15.

$$\text{Attention Output} = \text{LayerNorm}(\text{MultiHeadAttention}(Q, K, V) + \text{Input}) \quad (2.4.15)$$

3. Position-Wise Feed-Forward Network: The second sub-layer of each encoder layer is a position-wise Feed-Forward Network (FFN). This fully connected network is applied independently to each position in the sequence and consists of two linear transformations with a ReLU activation function in between, this process is defined in Equation 2.4.16.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2.4.16)$$

Here, W_1 and W_2 are learned weight matrices, and b_1 and b_2 are biases. This network introduces non-linearity and further refines the representation of each token. Like the multi-head attention sub-layer, the output of the FFN sub-layer passes through a residual connection and a normalization layer:

$$\text{FFN Output} = \text{LayerNorm}(\text{FFN}(x) + \text{Input}) \quad (2.4.17)$$

After processing through all N_x layers, the encoder outputs a continuous representation of the input sequence that encapsulates its contextual information.

Decoder Process

The decoder is responsible for generating the output sequence, which could be in a different modality or language, depending on the task. Similar to the encoder, the decoder is composed of a stack of N_x identical layers, but with an additional sub-layer

compared to the encoder. The following details the components and process of the decoder:

1. Input Embedding and Positional Encoding: As in the encoder, each token in the target sequence is mapped to a fixed-dimensional vector through an embedding layer. The target sequence embeddings are also combined with positional encodings using the same sinusoidal function to provide positional information.

2. Masked Multi-Head Self-Attention Mechanism: The first sub-layer in the decoder is a *masked* multi-head self-attention mechanism. The masking ensures that the decoder can only attend to earlier positions in the output sequence when generating a prediction for a given token. This is essential for autoregressive tasks, such as translation, where the model should not have access to future tokens during training.

The masked multi-head attention works similarly to the encoder’s multi-head attention, except that it prevents information flow from future tokens by applying a mask that blocks certain positions from being attended to. The output of this sub-layer is passed through residual connections and layer normalization.

3. Multi-Head Cross-Attention Mechanism: The second sub-layer in the decoder is a multi-head cross-attention mechanism, which allows the decoder to attend to the encoder’s output. Here, the Query vectors are derived from the decoder’s previous sub-layer, while the Key and Value vectors come from the encoder’s final output. This allows the decoder to focus on relevant parts of the encoded input sequence while generating each token in the output sequence. As before, residual connections and layer normalization are applied after this sub-layer.

4. Position-Wise Feed-Forward Network: The final sub-layer in the decoder is a position-wise feed-forward network, identical to the one used in the encoder. This network processes each token’s representation individually and further refines the output. As with the other sub-layers, residual connections and layer normalization follow this sub-layer.

5. Final Linear and Softmax Layer: After passing through all N_x decoder layers, the output is transformed by a linear layer followed by a softmax function to produce

a probability distribution over the target vocabulary. This distribution is used to predict the next token in the sequence. The decoding process continues iteratively until the entire sequence is generated.

Summary of the Encoder-Decoder Interaction

The interaction between the encoder and decoder is crucial for sequence-to-sequence tasks, where the input and output sequences may differ in length or modality. The encoder processes the entire input sequence to produce a fixed-length representation, which the decoder attends to during its autoregressive generation of the output sequence. The use of multi-head attention mechanisms in both the encoder and decoder, along with the cross-attention between them, enables the Transformer to capture complex dependencies and relationships across sequences, making it a powerful architecture for a wide range of tasks.

Transformer Architecture and Applications

The Transformer’s architecture, characterized by its parallelizable structure and the ability to capture long-range dependencies more efficiently than RNNs, has revolutionized sequence modeling in NLP and beyond. Its encoder-decoder design allows it to handle complex sequence-to-sequence tasks such as machine translation, text summarization, and question answering, while the self-attention mechanism enables rich context modeling across entire input sequences.

Recent works (Devlin et al., 2019; Brown et al., 2020) have extended the Transformer model to tasks beyond translation. Notable examples include the development of large pre-trained language models like BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), which have achieved state-of-the-art performance on a wide range of NLP tasks. These models leverage the Transformer’s capacity for contextual understanding and have set new benchmarks in language modeling, question answering, and text generation. The following section provides a general overview of BERT’s architecture and its pre-training tasks¹.

¹Pre-training tasks refer to unsupervised learning objectives that allow the model to learn general language representations from large amounts of data before being fine-tuned on specific tasks.

2.5. BERT: Bidirectional Encoder Representations from Transformers

Building upon the foundation of the Transformer architecture (Vaswani et al., 2017), BERT (Bidirectional Encoder Representations from Transformers) represents a significant advancement in pre-trained language models. BERT's architecture consists solely of the Transformer encoder stack, allowing it to fully capture bidirectional context from text, in contrast to previous models that were limited to unidirectional or shallow bidirectional contexts (Mohammed and Ali, 2021).

BERT employs a multi-layer bidirectional Transformer encoder, where each layer is composed of self-attention and feed-forward sub-layers. The model comes in two variants: BERT_{BASE}, with 12 layers (transformer blocks), 768 hidden units per layer, and 12 attention heads; and BERT_{LARGE}, with 24 layers, 1024 hidden units, and 16 attention heads. The input to BERT consists of a sequence of tokens, each represented by a sum of token embeddings, segment embeddings, and positional embeddings. Additionally, the model processes text in a manner that allows it to attend to all words in the sequence during both pre-training and fine-tuning.

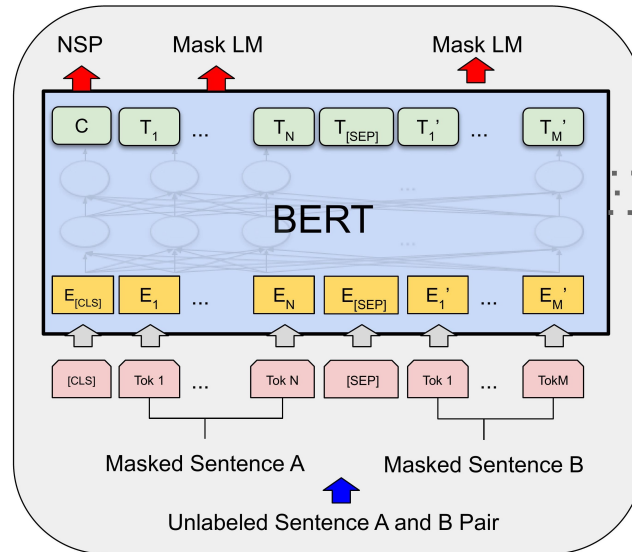


Figure 2.11: Architecture of the BERT model. Figure taken from Devlin et al. (2019).

BERT is pre-trained using two primary unsupervised tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, 15% of the tokens

in each input sequence are randomly masked, and the model is trained to predict these masked tokens based on their bidirectional context. This contrasts with previous models that could only predict tokens from a unidirectional context. The NSP task trains BERT to predict whether two given sentences follow each other in a document, further enhancing its ability to understand relationships between sentences. Figure 2.11 illustrates the overall architecture of the BERT model. The model takes as input a pair of unlabeled sentences, which are first tokenized and segmented using the [SEP] token as a separator. These tokenized inputs are then passed through multiple layers of Transformer encoders, and the outputs from these encoders are used for the pre-training tasks.

BERT was pre-trained on large corpora, namely the BooksCorpus (800M words) and English Wikipedia (2,500M words). This extensive pre-training enables BERT to capture a wide range of linguistic nuances and syntactic features, making it highly effective across a variety of tasks, including question answering, text classification, and language inference. For a more detailed explanation of the architecture, the reader is referred to the original paper by Devlin et al. (2019).

2.6. Evaluation Metrics and Statistical Test

To evaluate the performance of the proposed methods, ensure a fair comparison with state-of-the-art approaches, and comprehensively analyze the obtained results, this section introduces the various evaluation metrics and the statistical test used. The considered metrics include: precision, recall, F1 score, weighted F1 score, macro-average F1 score, the maximum possible accuracy, and diversity of coincidental failures. These metrics were chosen to provide a robust evaluation of the proposed methods and to ensure meaningful comparisons with other approaches.

In order to compare the performance of the different proposed methods, we employed the Bayesian Wilcoxon Signed-Rank Test, which is presented at the end of this section.

The used evaluation metrics rely on the following terms:

- TP (True Positive): A case where the classifier correctly predicts a positive instance.
- FP (False Positive): A case where the classifier incorrectly predicts a positive

instance.

- TN (True Negative): A case where the classifier correctly predicts a negative instance.
- FN (False Negative): A case where the classifier incorrectly predicts a negative instance.

2.6.1. Accuracy

The value is calculated as the number of correctly classified positive and negative elements, divided by the total number of correctly and incorrectly classified elements. The formula for obtaining accuracy is shown below.

$$A = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.6.1)$$

2.6.2. Precision

The value is calculated as the number of correctly classified positive elements, divided by the total number of classified positive elements. The formula for obtaining precision is shown below.

$$P = \frac{TP}{TP + FP} \quad (2.6.2)$$

2.6.3. Recall

The value is calculated as the number of correctly classified positive elements, divided by the total number of actual positive elements. The formula for obtaining recall is shown below.

$$R = \frac{TP}{TP + FN} \quad (2.6.3)$$

2.6.4. F_1 Score

The F_1 score is a measure that combines precision and recall. These measures help understand the performance of the classifier when there are more elements of

one class than another. The formula for obtaining the F_1 score is shown below.

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad (2.6.4)$$

2.6.5. Weighted F_1 Score

The weighted F_1 score is employed to evaluate the performance of a multi-class classifier, particularly in situations where class imbalance is present. To compute this metric, the F_1 score for each class is first calculated independently. Each F_1 score is then multiplied by a weight W_i , which corresponds to the proportion of instances belonging to that class. The weighted F_1 scores are subsequently summed, providing a performance metric that accounts for the distribution of classes within the dataset.

$$\text{weighted } F_1 = W_1 \cdot F1_{\text{class1}} + W_2 \cdot F1_{\text{class2}} + \dots + W_n \cdot F1_{\text{classn}} \quad (2.6.5)$$

2.6.6. Macro-average F_1 Score

The macro-average F_1 score is utilized to assess the performance of a multi-class classifier without taking class imbalance into account, ensuring that no class is given preferential treatment. To compute this metric, the F_1 score for each class is calculated independently, and then an average is taken across the n classes. This results in a single performance measure that treats each class equally, regardless of the number of instances in each class.

$$\text{macro-average } F_1 = \frac{F1_{\text{class1}} + F1_{\text{class2}} + \dots + F1_{\text{classn}}}{n} \quad (2.6.6)$$

2.6.7. Maximum Possible Accuracy

The use of *Maximum Possible Accuracy* (MPA) is introduced to evaluate the complementarity between different classifiers in ensemble or multi-classifier systems (Hossin and Sulaiman, 2015). This metric is particularly useful in understanding how the combined efforts of multiple classifiers can enhance overall performance. MPA is defined as the ratio of correctly classified instances to the total number of instances in the dataset. An instance is deemed correctly classified if at least one of the classifiers within the ensemble is able to assign the correct label to it. By analyzing MPA, we can

gain valuable insights into the extent to which different classifiers complement each other, identifying cases where one classifier compensates for the weaknesses of another. This metric is especially useful in diverse ensembles where individual classifiers may have different strengths and weaknesses, thus enabling a better understanding of the potential benefits of combining multiple classifiers to improve classification accuracy.

2.6.8. Coincident Failure Diversity

The *Coincident Failure Diversity* (CFD) is a metric designed to quantify the diversity of errors among different classifiers (Tang, Suganthan, and Yao, 2006). This metric is particularly important for evaluating the robustness of ensemble methods or multi-classifier systems by assessing how differently each classifier behaves when faced with challenging patterns. The CFD metric ranges from 0 to 1. A value of 0 indicates minimal diversity, meaning all classifiers either correctly classify or simultaneously misclassify the same instances, exhibiting identical behavior. On the other hand, a value of 1 signifies maximum diversity, where all classifiers make distinct classification errors, ensuring that every misclassified instance is unique across the classifiers.

CFD is valuable in ensemble learning because diversity among classifiers is often associated with improved generalization. When classifiers make different mistakes, the ensemble can capitalize on this diversity to potentially correct those errors, leading to better overall performance. The formal calculation of the CFD metric is presented in Equation 2.6.7, where L represents the total number of classifiers, p_0 is the probability that all L classifiers correctly classify a randomly selected instance, and p_i denotes the probability that i randomly selected classifiers fail to classify a randomly selected instance correctly.

$$CFD = \begin{cases} 0, & p_0 = 1 \\ \frac{1}{1 - p_0} \sum_{i=1}^L \frac{L - i}{L - 1} p_i, & p_0 < 1 \end{cases} \quad (2.6.7)$$

2.6.9. Inter-Annotator Agreement

In many ML and NLP tasks, datasets must be labeled by human annotators. However, the subjective nature of labeling can introduce variability among different annotators (Nowak and Rüger, 2010). To ensure the reliability and consistency of

labeled data, it is essential to quantify the level of agreement between annotators. A high level of agreement indicates that the annotations are reliable and can be used confidently in downstream applications, whereas a low agreement may suggest the need for further clarification of annotation guidelines or re-evaluation of the dataset. In the following subsections, we introduce Cohen’s Kappa and Fleiss’Kappa values, two widely used statistical measures for assessing inter-annotator agreement.

Cohen’s Kappa

Cohen’s Kappa (κ) is a statistical measure that quantifies the agreement between two raters who classify items into mutually exclusive categories, this measure was originally proposed by Cohen (1960). It accounts for the agreement occurring by chance. The formula for Cohen’s Kappa is presented in Equation 2.6.8.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (2.6.8)$$

where:

- p_o is the observed agreement between the two raters, calculated as:

$$p_o = \sum_i P_{ii} \quad (2.6.9)$$

where P_{ii} represents the proportion of items classified in category i by both raters.

- p_e is the expected agreement due to chance, computed as:

$$p_e = \sum_i P_{i+} P_{+i} \quad (2.6.10)$$

where P_{i+} and P_{+i} are the marginal probabilities of category i for the first and second rater, respectively.

The Kappa coefficient ranges from -1 to 1: where $\kappa = 1$ indicates perfect agreement, $\kappa = 0$ suggests agreement equivalent to chance, and $\kappa < 0$ implies disagreement worse than chance. Table 2.6.9 presents the interpretation of Kappa values.

Kappa	Interpretation
< 0,00	Poor agreement
0,00 – 0,20	Slight agreement
0,21 – 0,40	Fair agreement
0,41 – 0,60	Moderate agreement
0,61 – 0,80	Substantial agreement
0,81 – 1,00	Almost perfect agreement

Table 2.1: Interpretation ranges of Kappa values.

Fleiss' Kappa

Fleiss' Kappa is an extension of Cohen's Kappa for multiple raters, this measure was introduced by Fleiss and others (1971). It measures the reliability of agreement among N raters classifying items into k categories. Equation 2.6.11 presents the formula for Fleiss' Kappa.

$$\kappa = \frac{P - P_e}{1 - P_e} \quad (2.6.11)$$

where:

- P is the observed agreement, computed as:

$$P = \frac{1}{N} \sum_{i=1}^N \left[\sum_{j=1}^k p_{ij}^2 - 1 \right] \quad (2.6.12)$$

where p_{ij} is the proportion of raters who classified item i into category j .

- P_e is the expected agreement, defined as:

$$P_e = \sum_{j=1}^k p_{.j}^2 \quad (2.6.13)$$

where $p_{.j}$ is the overall proportion of ratings assigned to category j .

Similar to Cohen's Kappa, Fleiss' Kappa values range between -1 and 1, with similar interpretations regarding the level of agreement. For a more detailed explanation of the calculation of Cohen's Kappa and Fleiss' Kappa measures, we refer the reader to the following papers (Cohen, 1960; Fleiss and others, 1971).

2.6.10. Bayesian Comparison of Classifiers Using the Wilcoxon Signed-Rank Test

In order to evaluate the performance of different machine learning classifiers across multiple datasets, it is essential to utilize robust statistical methods that account for the inherent variability of the data. Traditional methods, such as the frequentist paired t -test or the Wilcoxon signed-rank test, can help determine whether there are significant differences in classifiers' performance. However, these approaches only provide p-values, which offer limited information regarding the magnitude and uncertainty of the differences between classifiers (Gardner and Brooks, 2017).

To address these limitations, we employ the *Bayesian Wilcoxon signed-rank test* as a means of comparing the performance of the proposed approaches. The Bayesian framework allows for richer probabilistic interpretations of the differences, including credible intervals that reflect uncertainty. By utilizing the Bayesian Wilcoxon signed-rank test, we can infer the probability that one classifier outperforms another and estimate the extent of this superiority. This method is particularly useful when the assumption of normality is not met or when we wish to incorporate prior knowledge about the performance of the classifiers.

Bayesian Wilcoxon Signed-Rank Test

The classical Wilcoxon signed-rank test is a non-parametric test used for comparing two related samples, which, in the context of this research, refers to the performance of two classifiers on the same set of datasets (Rey and Neuhäuser, 2011). Given that the test does not assume normality of the differences between the paired samples, it is particularly well-suited to handling performance metrics such as accuracy, precision, or F1-score, which may not follow a normal distribution.

The Bayesian adaptation of the Wilcoxon signed-rank test enhances this classical approach by replacing hypothesis testing with probability modeling. Instead of determining whether there is sufficient evidence to reject a null hypothesis, it estimates the posterior distribution of the difference between the performances of the two classifiers, providing a more detailed insight into the nature of these differences (Benavoli et al., 2014).

Let d_i denote the difference in performance between two classifiers for dataset i , where $d_i = X_i - Y_i$, and X_i and Y_i represent the performance of classifiers A and

B on the same dataset, respectively. The classical Wilcoxon test ranks the absolute differences $|d_i|$, assigns signed ranks, and computes a test statistic. The Bayesian approach, however, treats the differences d_i as random variables and models them using a probability distribution.

The aim of the Bayesian Wilcoxon signed-rank test is to estimate the posterior distribution $p(\theta|D)$, where θ represents the central tendency (e.g., median) of the differences, and D is the observed data. This is done by combining a prior distribution $p(\theta)$ with the likelihood of observing the data given the model parameters. The resulting posterior distribution is given by Bayes' theorem:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \quad (2.6.14)$$

where $p(D|\theta)$ is the likelihood of the data given the parameter θ , and $p(D)$ is the marginal likelihood.

Priors and Likelihood

In the Bayesian framework, prior distributions reflect our beliefs about the parameters before observing the data. For example, if we have no strong prior information about the performance differences between classifiers, we may use a non-informative prior such as a uniform distribution. If, on the other hand, we have reasons to believe that one classifier generally performs better, we may use an informative prior that reflects this belief.

The likelihood function $p(D|\theta)$ in the Bayesian Wilcoxon signed-rank test is constructed based on the ranks of the differences d_i . Since the test is non-parametric, it does not assume a specific distribution for the differences but rather models the likelihood based on the signed ranks.

Posterior Distribution and Credible Intervals

Once the prior and likelihood are defined, the posterior distribution of θ is computed. This posterior distribution provides a full probabilistic description of the parameter, allowing us to make statements such as “there is a 95 % probability that the difference in performance between classifiers A and B lies within a certain range”.

The posterior distribution also allows us to compute credible intervals, which are the Bayesian equivalent of confidence intervals in frequentist statistics. A credible

interval provides a range within which the true value of θ is likely to lie with a specified probability, say 95 %.

For example, if the 95 % credible interval for θ is $[-0,02, 0,05]$, we can state with 95 % probability that the performance difference between classifiers A and B lies within this range. If the interval includes zero, it suggests that there is a significant probability that the classifiers perform similarly.

Posterior Probability of Superiority

In addition to estimating credible intervals, the Bayesian Wilcoxon signed-rank test allows us to compute the posterior probability that one classifier outperforms the other. Let $P(\theta > 0)$ denote the posterior probability that the difference in performance is positive, meaning that classifier A outperforms classifier B . If $P(\theta > 0) = 0,90$, we can state that there is a 90 % probability that classifier A is superior to classifier B .

This type of probabilistic interpretation provides a more intuitive understanding of the performance differences compared to classical p-values, which only give a measure of the likelihood of observing the data under the null hypothesis.

The Bayesian Wilcoxon signed-rank test provides a powerful and flexible tool for comparing the performance of classifiers, particularly when the data do not meet the assumptions of traditional parametric tests. By incorporating prior information and providing a full posterior distribution of the differences, this method allows for richer inferences about the superiority of one classifier over another. Specifically, it returns probabilities that, based on the measured performance, one classifier is better than another, or that they are within the region of practical equivalence. A concise way to view this is that the test produces a *posterior distribution* over the possible differences in performance between the two classifiers. In practice, one draws random samples from this posterior distribution typically via a Markov Chain Monte Carlo (MCMC) routine (Benavoli et al., 2014), where each draw corresponds to one plausible “scenario” in which the difference $A - B$ is a specific value.

Assuming we compare two classifiers, A and B , the test then returns the probability that the first classifier is better than the second $P(A > B)$, the probability of a tie $P(\text{rope})$, and the probability that the second classifier is better than the first $P(B > A)$. As proposed by Benavoli et al. (2014), to help visualize these results, n MCMC samples are usually mapped in barycentric coordinates, where each vertex of the triangle is associated with one of the Bayesian test scenarios, and each point

represents a statistical comparison between the two classifiers. As an example, Figure 2.12 compares two different classifiers. As can be observed, there is a strong tendency for classifier B to outperform classifier A, along with a smaller tendency toward a tie between the two classifiers. In this example, 150,000 samples were drawn, resulting in 150,000 points in the barycentric plot. Each point's location is determined by whether the sampled difference favors A , favors B , or falls within the region of practical equivalence.

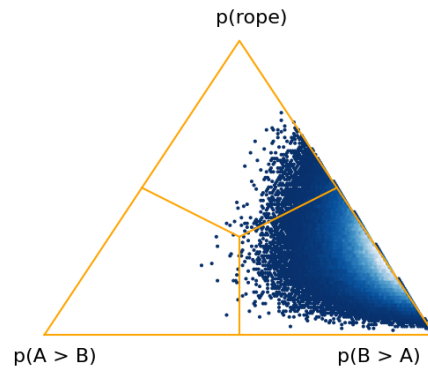


Figure 2.12: Example of the visualization of the Bayesian Wilcoxon signed-rank test when comparing two classifiers. The Figure was generated using the library provided by Benavoli et al. (2014).

For more details on the implementation and theory behind the Bayesian Wilcoxon signed-rank test, we refer the reader to (Benavoli et al., 2014, 2017).

Chapter 3

Related Work

This section presents a comprehensive description and analysis of previous work related to attention mechanisms, as well as various techniques and approaches used for the detection of AL in social media. The section is divided into four main subsections, each addressing a different aspect of the topic. The first subsection provides an overview of the different attention mechanisms and the various Transformer architectures proposed in the literature. The second subsection focuses on both unimodal and multimodal approaches for the detection of abusive language in social media. The third subsection presents an overview of some of the most significant evaluation campaigns and benchmarks related to AL detection in social media. Finally, the fourth subsection discusses the current limitations of the main approaches in the detection of abusive language within both unimodal and multimodal scenarios.

3.1. Attention Mechanisms and Transformer-based Approaches

Several variants of attention mechanisms have been proposed, each contributing to significant advancements in the state of the art across a wide range of tasks, including machine translation (Vaswani et al., 2017), text classification and representation (Chakrabarty, Gupta, and Muresan, 2019), image captioning (Xu et al., 2015), video captioning (Pu et al., 2018), visual question answering (Kanakamedala et al., 2021), and generative modeling (Zhang et al., 2019a). The effectiveness of attention mechanisms has been substantiated by extensive empirical evidence (Niu, Zhong, and Yu, 2021), which has motivated the research community to explore and refine these mechanisms further.

According to Chaudhari et al. (2021), attention mechanisms can be categorized into four primary categories, which are not mutually exclusive. These categories include:

- 1.- The use of single or multiple input *sequences* simultaneously (Bahdanau, Cho, and Bengio, 2015; Lu et al., 2016), where the attention weights are jointly learned to capture interactions between these input sequences. This approach allows for the integration of information across different sequences, enhancing the model’s ability to capture complex relationships.
- 2.- The use of single or hierarchical levels of *abstractions*, where attention weights are computed either for the original input sequence alone or across multiple levels of abstraction within an input sequence (Yang et al., 2016; Zhao and Zhang, 2018). This enables the model to focus on different levels of detail, which can be crucial for tasks requiring nuanced understanding, such as hierarchical text classification.
- 3.- The application of the attention mechanism at different *positions* within the input sequence. This can be divided into soft and hard attention mechanisms. In soft attention, a weighted average of all hidden states in the input sequence is used to build the context vector (Bahdanau, Cho, and Bengio, 2015), whereas in hard attention, the context vector is computed from stochastically sampled hidden states within the input sequence (Xu et al., 2015). This distinction allows models to either focus broadly on the entire sequence or selectively concentrate on specific parts.
- 4.- The use of single or multiple input sequence *representations*, where the latter is employed to assign importance weights to different representations. This helps in identifying the most relevant aspects of the input while filtering out noise and redundancies (Maharjan et al., 2018; Kiela, Wang, and Cho, 2018). By focusing on the most salient features, this approach can enhance the model’s ability to make accurate predictions, especially in complex, multi-modal tasks.

Among single- and multi-sequence attention mechanisms, the most widely adopted are the contextual attention mechanism (Yang et al., 2016) and the self-attention mechanism (Vaswani et al., 2017). These mechanisms have been predominantly applied in tasks such as document classification, text representation, and neural machine

translation (Hu, 2020; Chaudhari et al., 2021). The key distinction between these two lies in how they utilize the query vector or matrix: in self-attention, the query is derived from a linear projection of the same sequence, whereas in contextual attention, it is jointly learned during the training process of the neural network. This distinction enables the contextual attention mechanism to calculate the similarity of the elements of the sequence with respect to the context learned during training.

In this doctoral research, we introduce a novel *Dual Attention* (DA) mechanism that unifies both self-attention and contextual attention mechanisms. The proposed DA mechanism generates a contextualized representation by leveraging the strengths of these approaches, thereby preserving the relevance of each element within the sequence relative to both the entire sequence and the context learned during the training process. This dual perspective allows for a more nuanced understanding of the relationships between elements, enhancing the model’s ability to capture complex patterns in data.

It is important to note that the term DA has been previously employed in various studies to describe the weighted combination of different modalities or sources of information through attention mechanisms. For instance, (Fu et al., 2019) combined features from image regions across different channels for the task of scene segmentation. Similarly, (Li et al., 2019b) introduced a dual attention mechanism for Dialogue Act classification, where the mechanism integrates information extracted from dialogues with the different topics addressed within them. Furthermore, (Xiao et al., 2019) proposed a dual attention mechanism that combines features of objects and actions for reasoning about human-object interactions. More recently, in (Li et al., 2023b), a dual attention mechanism was proposed, combining representations obtained by independently applying attention mechanisms to two feature sets that measured temporality and readings from a sensor array for Remaining Useful Life (RUL) detection.

Unlike these existing approaches, our proposed DA mechanism integrates representations from two distinct attention mechanisms. To the best of our knowledge, our proposed DA mechanism is the first to integrate both self-attention and contextual attention mechanisms across single-modal and cross-modal settings. This combination allows the mechanism to focus on the relationships between each pair of elements in the sequence and the relevance of each element with respect to the context learned during training. This context is specifically related to the application domain, making the DA mechanism highly adaptable to various tasks where understanding the

interplay between sequence elements and their contextual relevance is crucial.

To provide a clearer overview of the main differences between our proposed DA mechanism and various single- and multi-sequence attention mechanism variants based on self-attention and contextual attention, Table 3.1 presents a comparative analysis. As shown, our work leverages both attention mechanisms, not only by combining attention representations but also by integrating both self-attention and contextual attention mechanisms. The application of our DA mechanism differs based on the input type: for single input sequences (texts), it applies the dual attention mechanism, while for multiple input sequences (memes), it employs the cross-modal dual attention mechanism.

Reference	Single(S)/ Multiple(M) Input Sequence	Self- Attention	Contextual Attention	Single(S)/ Hierarchical(H) Level Attention	Combine Attention Representations	Combine Self and Contextual Attention
(Yang et al., 2016)	S	-	✓	H	✓	-
(Vaswani et al., 2017)	S	✓	-	H	✓	-
(Ye et al., 2019)	M	✓	-	H	✓	-
(Fu et al., 2019)	M	✓	-	S	✓	-
(Li et al., 2019b)	S	✓	-	S	-	-
(Xiao et al., 2019)	M	✓	-	S	✓	-
(Chakrabarty, Gupta, and Muresan, 2019)	S	-	✓	S	-	-
(Yan et al., 2022)	S	✓	-	H	✓	-
(Li et al., 2023b)	M	✓	-	H	✓	-
Dual Attention (ours)	S/M	✓	✓	S/H	✓	✓

Table 3.1: Comparative table of our dual attention mechanism versus various single- and multi-sequence attention mechanism variants, based on self-attention and contextual attention mechanisms.

The use of self-attention has gained significant popularity in recent years, largely due to its integration into Transformer-based neural network models (Lin et al., 2022). The Transformer model is a prominent DL architecture that has been widely adopted across various fields, including NLP (Gulati et al., 2020; Ramprasath et al., 2022) and Computer Vision (CV) (Khan et al., 2022). One of the primary applications of the Transformer models in NLP is text classification, which has become popular in recent years due to its outstanding performance in a diverse array of domains. These domains include sentiment analysis (Durairaj and Chinnalagu, 2021; Tabinda Kokab, Asghar, and Naz, 2022), depression detection (Haque et al., 2020; Malviya, Roy, and Saritha, 2021), deception detection (Wawer and Sarzyńska-Wawer, 2022), sentence

pair classification (Devlin et al., 2019; Ding et al., 2021), and the identification of AL content (Mutanga, Naicker, and Olugbara, 2020; Bindra, Sharma, and Bansal, 2022), among others.

As detailed in Section 2.4.3, the Transformer architecture consists of both encoder and decoder layers. Among the most widely used pre-trained NLP models that utilize only the encoder layers are BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ERNIE (Zhang et al., 2019b), and DistilBERT (Sanh et al., 2019). On the other hand, models that leverage the decoder for text sequence generation include GPT-2 (Cohen and Gokaslan, 2020), GPT-3 (Brown et al., 2020), BART (Lewis et al., 2020), and LLaMA (Touvron et al., 2023). These pre-trained models are distinguished by two key factors: 1) the datasets used for training, and 2) the pre-training tasks, which refer to specific tasks designed to train a model on large amounts of unlabeled data before fine-tuning it for a particular downstream task. The goal of pre-training is to enable the model to learn general and useful representations of language (or other modalities, such as images) that can be transferred to various downstream tasks with less labeled data (Zhang et al., 2023).

In the context of CV, pre-trained models have adapted the Transformer architecture originally proposed by Vaswani et al. (2017), where each token is now represented by a region of the image. Some of the most popular pre-trained models in this area include ViT (Dosovitskiy et al., 2020), Swin Transformer (Liu et al., 2022), and the BLIP model (Li et al., 2023a). In addition to these unimodal models, in recent years, multimodal pre-trained models have been developed by integrating both text and image modalities. Among the most widely used models are ViLBERT (Lu et al., 2019), VisualBERT (Li et al., 2019a), VL-BERT (Su et al., 2020), CLIP (Radford et al., 2021), Gemini 1.5 (Gemini-Team et al., 2024), and GPT-4 (OpenAI et al., 2024).

These multimodal models represent a significant advancement in the field, as they harness the strengths of both text and image representations, enabling the creation of more robust and contextually aware models. The integration of multiple modalities within a single model architecture not only enhances performance but also broadens the range of applications, making these models highly versatile and effective for tasks that require a deep understanding of both visual and textual information.

3.2. Abusive language detection in social media

A wide variety of research related to the detection of AL focuses on the detection of sexist, racist, hateful, aggressive, and offensive content on social media platforms (MacAvaney et al., 2019; Wenjie and Arkaitz, 2021). Much of this research has been conducted from a supervised learning perspective, utilizing various data preprocessing techniques, a range of text representations, and a diverse array of machine learning algorithms, including both traditional and deep learning approaches (Schmidt and Wiegand, 2017). These studies often focus on optimizing the detection performance through the careful selection of features and model architectures, tailored to the specific characteristics of the dataset and the type of AL being targeted.

The following subsections provide a detailed overview of the most relevant approaches to the detection of AL, categorized by their focus on unimodal (text-based) and multimodal (meme-based) content. This categorization highlights the differences in methodologies and challenges encountered when dealing with pure text versus the more complex, multimodal nature of memes, where both textual and visual elements contribute to the overall meaning and potential harmfulness of the content.

3.2.1. Detection of abusive language in text

Various methods have been proposed for the detection of AL using textual information. These approaches range from traditional NLP techniques to more advanced deep learning-based models, which currently constitute the state of the art in this domain (Poletto et al., 2021; Jahan and Oussalah, 2023). The range of features employed to address this challenge is broad, reflecting the evolution of techniques over time.

Initial methods often relied on bag-of-words representations, utilizing word and character n-grams as input features to build classifiers (Burnap and Williams, 2016; Nobata et al., 2016; Zeerak and Dirk, 2016; Gaydhani et al., 2018). Although effective in certain contexts, these approaches sometimes struggle with generalization.

To enhance the generalization capabilities of classifiers, subsequent approaches have integrated word embeddings as input features for their models. Word embeddings, which capture semantic relationships between words in a continuous vector space, have proven to be more effective in representing linguistic nuances compared

to traditional n-gram methods (Nobata et al., 2016; Zhang, Robinson, and Tepper, 2018; Chakrabarty, Gupta, and Muresan, 2019).

More recently, the field has witnessed a shift towards the use of sophisticated text representations generated by pre-trained Transformer-based neural language models. These models, including ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019), provide deep contextualized word representations that capture complex patterns in language. By fine-tuning these pre-trained models for the specific task of AL detection, researchers have achieved significant performance improvements (Liu, Li, and Zou, 2019; Nikolov and Radivchev, 2019; Mozafari, Farahbakhsh, and Crespi, 2019a; Mutanga, Naicker, and Olugbara, 2020; Shrivastava, Pupale, and Singh, 2021; Yigezu et al., 2023). These Transformer-based models have become the benchmark for AL detection tasks, offering robust and adaptable solutions across a wide range of datasets and contexts.

Regarding the classification stage, various approaches and techniques have been proposed to enhance the detection of AL. These approaches can be broadly categorized into two main groups. The first group comprises traditional classification algorithms, such as Support Vector Machines (SVMs), Naive Bayes, Logistic Regression, and Random Forest. These methods have been widely used in earlier studies due to their effectiveness in handling structured data and their relatively straightforward implementation (Zeeraak and Dirk, 2016; Burnap and Williams, 2016; Davidson et al., 2017; Schmidt and Wiegand, 2017; Gaydhani et al., 2018; MacAvaney et al., 2019; Abro et al., 2020).

The second group includes deep learning-based approaches, which have gained prominence in recent years due to their ability to automatically extract and learn complex patterns from data. These approaches typically employ Convolutional Neural Networks (CNNs) for feature extraction at the word and character levels (Badjatiya et al., 2017; Gambäck and Sikdar, 2017; Ashwin, Irina, and Dominique, 2020; Roy et al., 2020), and RNNs for capturing word and character dependencies, thereby enhancing the model’s ability to understand sequential information (Badjatiya et al., 2017; Saksesi, Nasrun, and Setianingsih, 2018; Pitsilis, Ramampiaro, and Das, 2018; Chakrabarty, Gupta, and Muresan, 2019; Ashwin, Irina, and Dominique, 2020). Additionally, some approaches combine CNNs and RNNs to create hybrid architectures that effectively capture both spatial and temporal features, leading to more powerful models for AL detection (Zhang, Robinson, and Tepper, 2018; Huynh et al., 2019;

Duwairi, Hayajneh, and Quwaider, 2021).

Recent advancements in AL detection have increasingly incorporated deep learning architectures enhanced with attention mechanisms. These mechanisms enable models to automatically weigh the importance of different features, thereby improving their ability to focus on the most relevant aspects of the input data (Chaudhari et al., 2021). One of the pioneering works in this area employed the self-attention mechanism to detect abusive language in portal news and Wikipedia, marking a significant step forward in the field (Pavlopoulos, Malakasiotis, and Androutsopoulos, 2017). Following this, the contextual attention mechanism, first introduced by Yang et al. (2016), has shown promising results in enhancing sentence representations for AL detection tasks (Chakrabarty, Gupta, and Muresan, 2019; la Peña Sarracén et al., 2018; Jarquín-Vásquez, Montes-y Gómez, and Villaseñor-Pineda, 2020).

Moreover, the use of Transformer-based models has become increasingly popular in recent years, representing the current state of the art in AL detection. These approaches range from fine-tuning pre-trained Transformer models (Rani et al., 2020; Kovács, Alonso, and Saini, 2021), to employing ensembles of Transformers to boost performance further (Mnassri et al., 2022; Mazari, Boudoukhani, and Djeflal, 2023). Recent approaches have focused on retraining pre-trained Transformer models with social media data containing offensive content, aiming to better adapt these models to the specific domain of AL detection (Caselli et al., 2021; Jarquín-Vásquez, Escalante, and Montes-y Gómez, 2023).

3.2.2. Detection of abusive language in memes

The detection of AL has predominantly been addressed through the use of textual resources (Schmidt and Wiegand, 2017; Naseem et al., 2019). However, in recent years, there has been growing interest in expanding AL detection to a multimodal perspective. This shift is particularly evident in the detection of abusive memes on social media, where both textual and visual information are combined to create context (Afridi et al., 2020). The classification of memes represents a Vision & Language (V&L) multimodal problem, where diverse approaches have been developed to tackle the challenges posed by this task.

These approaches can be broadly categorized into two main strategies: 1) the fusion of multimodal features, and 2) the use of pre-trained multimodal models (Bal-

trusaitis, Ahuja, and Morency, 2019; Kiela et al., 2020). The first strategy involves extracting features separately from the text and visual components and then combining them to form a unified representation, which is subsequently used for classification. This fusion process is critical, as it enables the model to leverage the complementary information present in both modalities. The second strategy leverages pre-trained multimodal models, which have been trained on large-scale datasets to understand the interplay between text and images. These models have demonstrated significant improvements in the accuracy of AL detection in memes (Hermida and Santos, 2023).

Figure 3.1 illustrates a general framework for the detection of AL in memes, based on the aforementioned approaches. In this framework, unimodal and multimodal pre-trained models can be employed directly for classification through a fine-tuning strategy or can undergo a feature fusion process before classification. This flexibility allows for the integration of various techniques to optimize performance, depending on the specific characteristics of the dataset and the task at hand.

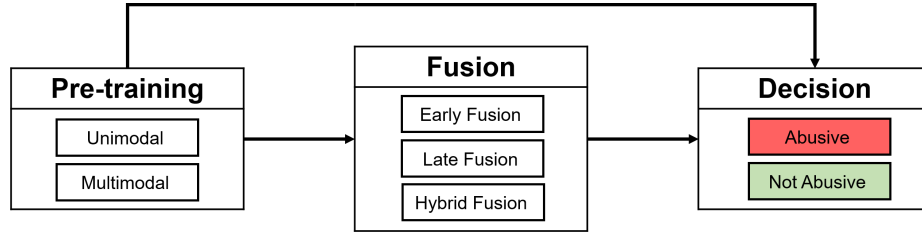


Figure 3.1: General scheme for the detection of AL in memes.

Regarding the proposed approaches for the detection of AL in memes, a significant number of them adhere to the classification scheme illustrated in Figure 3.1. Among these approaches, those based on feature fusion have been widely explored, utilizing a variety of techniques including early, late, and hybrid fusion of image and textual features (Oriol, Canton-Ferrer, and i Nieto, 2019; Keswani et al., 2020; Gomez et al., 2020; Constantin et al., 2021; Kirk et al., 2021). Early fusion, particularly in the form of concatenation or cross-modal fusion, has been extensively employed due to its ability to achieve better alignment between textual and visual features (Hermida and Santos, 2023; Yang et al., 2024).

The features used in these fusion approaches are often extracted from unimodal pre-trained models. For linguistic features, models such as BERT (Devlin et al., 2019), ELMo (Peters et al., 2018), GPT-2 (Radford et al., 2019), RoBERTa (Liu et al., 2019), and ERNIE (Zhang et al., 2019b) are commonly used. For visual features, popular

models include AlexNet (Krizhevsky, Sutskever, and Hinton, 2012), VGG (Simonyan and Zisserman, 2015), GoogLeNet (Szegedy et al., 2015), ViT (Dosovitskiy et al., 2020), and Swin Transformer (Liu et al., 2022). Recent studies have also incorporated image captions as an additional modality, with most of these captions generated by the BLIP model (Li et al., 2023a) and integrated through an early fusion process (Maqbool and Fersini, 2024).

On the other hand, some approaches have opted for the use of pre-trained multi-modal models, which leverage joint multimodal information (image and text) for AL detection (Suryawanshi et al., 2020; Lee et al., 2021; Kirk et al., 2021; Zhou, Chen, and Yang, 2021; Kumar and Nandakumar, 2022; Arya et al., 2024). These multimodal models have demonstrated superior performance in AL detection, as they are trained to process both modalities simultaneously, allowing for a more effective alignment between text and image (Sharma et al., 2020; Yu et al., 2019b). Among the multimodal models, transformer-based representations such as ViLBERT (Lu et al., 2019), VisualBERT (Li et al., 2019a), VL-BERT (Su et al., 2020), CLIP (Radford et al., 2021), Gemini 1.5 (Gemini-Team et al., 2024), and GPT-4 (OpenAI et al., 2024) have been the most widely used.

3.3. Evaluation campaigns for abusive language detection in social media

Considering the well-acknowledged rise of AL on social media platforms, a substantial number of datasets, workshops, and evaluation campaigns have been developed to mitigate the impact of such content (Poletto et al., 2021). The majority of these efforts have centered around the detection of abusive messages, with a predominance of resources created in the English language.

In 2018, the first workshop on *Trolling, Aggression, and Cyberbullying (TRAC-1)*¹ was held at the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). This workshop aimed to classify text data into categories such as Overtly Aggressive, Covertly Aggressive, and Non-aggressive, providing a focused approach to addressing varying levels of aggression in online communication (Kumar et al., 2018).

¹<https://sites.google.com/view/trac1/shared-task>

Similarly, the *Automatic Misogyny Identification (AMI)*² task was introduced as part of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018). This task specifically targeted the automatic identification of misogynistic content in English and Italian tweets, expanding the focus of abusive language to include gender-based hate speech (Fersini, Nozza, and Rosso, 2018).

In addition to these efforts, the OffensEval shared tasks on *Identifying and Categorizing Offensive Language in Social Media*³⁴ were presented at the International Workshop on Semantic Evaluation (SemEval) in 2019 and 2020. These tasks emphasized not only the identification of offensive language but also the automatic categorization of offense types and the identification of offense targets, contributing significantly to the broader understanding and handling of offensive content (Marcos et al., 2019; Zampieri et al., 2020).

In parallel with these efforts, the HatEval shared task on *Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter* was presented at SemEval 2019 (Basile et al., 2019). This task focused on detecting hate speech in both English and Spanish, specifically targeting immigrants and women. Additionally, it aimed to identify the presence of aggressive attitudes and to classify the nature of the target, distinguishing between individuals and groups.

Additionally, the shared task on *Hate Speech and Offensive Content Identification in Indo-European Languages*⁵ (HASOC) has focused on detecting hate speech and offensive content, encompassing a wide range of subtasks. These subtasks include binary classification and fine-grained classification, distinguishing whether the texts target an individual, a group, or are untargeted. This shared task has been conducted annually since 2019, organized within the framework of the annual meeting of the Forum for Information Retrieval Evaluation (FIRE) (Satapara et al., 2024).

Furthermore, the shared task *sEXism Identification in Social neTworks*⁶ (EXIST) has been dedicated to detecting sexism, incorporating various subtasks focused on binary and multi-class classification of different types of sexism, in both English and Spanish. This task has been held since 2021, with its last three editions presented at

²<https://amievalita2018.wordpress.com/>

³⁴<https://competitions.codalab.org/competitions/20011>

⁴<https://competitions.codalab.org/competitions/22917>

⁵<https://hasocfire.github.io/hasoc/2024/index.html>

⁶<https://nlp.uned.es/exist2024/>

the Conference and Labs of the Evaluation Forum (CLEF) (Plaza et al., 2024).

In recent years, shared tasks focused on the Spanish language have also been developed, primarily within the context of the Iberian Languages Evaluation Forum (IberLEF). Notable among these is the MEX-A3T shared task⁷, which focuses on the detection of fake news and aggression in Mexican Spanish tweets (Aragón et al., 2020). Another relevant task is MeOffendEs⁸, centered on the detection of offensive content in various Spanish variants (del Arco et al., 2021). Furthermore, the HOMO-MEX task⁹ has been dedicated to the detection of LGBTQ+ phobic content in Spanish tweets, highlighting the importance of addressing homophobic and transphobic language in social media (Bel-Enguix et al., 2023). Similarly, the HODI shared task was proposed in the context of EVALITA 2023 (Nozza et al., 2023). This task focuses on detecting hate speech targeting the LGBTQIA+ community in Italian, as well as identifying the specific tokens within the sequence that contribute to the hateful nature of the message.

Regarding the detection of AL in memes, in recent years, there has been a growing interest from both the research community and industry in addressing this complex issue. One notable example is the 2020 *Hateful Memes Challenge*¹⁰ organized by Facebook, which focused on the detection of hate speech within memes. This challenge utilized a manually annotated dataset of 10k memes, providing a valuable resource for the development of robust detection approaches (Kielbaso et al., 2020).

In the 14th International Workshop on Semantic Evaluation (SemEval-2020) was introduced the *Memotion Analysis*¹¹ shared task, which aimed to classify memes based on their sarcastic, humorous, and offensive content. This task highlighted the multifaceted nature of meme analysis, where the boundary between humor and offense can be particularly challenging to delineate (Sharma et al., 2020).

Further advancing the field, in the 16th International Workshop on Semantic Evaluation (SemEval-2022) was presented the *Multimedia Automatic Misogyny Identification (MAMI)*¹² shared task. This task was divided into two main sub-tasks: Sub-task A focused on the identification of misogynistic memes, while Sub-task B required par-

⁷<https://sites.google.com/view/mex-a3t/home>

⁸<https://competitions.codalab.org/competitions/28679>

⁹<https://codalab.lisn.upsaclay.fr/competitions/10019>

¹⁰<https://ai.facebook.com/blog/hateful-memes-challenge-and-data-set/>

¹¹<https://competitions.codalab.org/competitions/20629>

¹²<https://competitions.codalab.org/competitions/34175>

ticipants to recognize the specific type of misogyny portrayed, including categories such as stereotyping, shaming, objectification, and violence. This nuanced approach underscores the importance of understanding the diverse forms of misogynistic content in online media (Fersini et al., 2022).

Finally, within the framework of IberLEF, the *Detection of Inappropriate Memes from Mexico (DIMEMEX)*¹³ task was proposed. This task focuses on detecting inappropriate content and hate speech in Mexican Spanish memes, reflecting the growing need to develop culturally and linguistically specific tools for meme analysis (Jarquín-Vásquez et al., 2024).

3.4. Discussion

A wide array of approaches has been proposed for the detection of AL in text and memes, ranging from traditional fusion and classification techniques (Schmidt and Wiegand, 2017; MacAvaney et al., 2019; Poletto et al., 2021; Wenjie and Arkaitz, 2021) to more advanced approaches based on DL (Kiela et al., 2020; Zhou, 2020; Rani et al., 2020; Kovács, Alonso, and Saini, 2021; Wenjie and Arkaitz, 2021). Due to their powerful representational capabilities and ability to capture multiple levels of abstraction, DL-based approaches have gained significant traction in recent years (Guo, Wang, and Wang, 2019; Li et al., 2019a). In particular, V&L representations have shown remarkable promise. The TNN architecture (Vaswani et al., 2017) is built upon many of these models and has been pivotal in creating pre-trained V&L representations. By leveraging the self-attention mechanism, these models effectively capture the relationships between pairs of image regions and words, leading to state-of-the-art results across a wide range of V&L tasks, including the detection of AL in memes (Su et al., 2020; Lu et al., 2019; Li et al., 2019a).

In the context of AL detection in text, DL-based approaches also represent the cutting edge of current research (Zhou, 2020; Wenjie and Arkaitz, 2021). Specifically, pre-trained language models such as BERT, RoBERTa, ERNIE, and GPT-2, which are based on the TNN architecture, have become increasingly popular. This popularity is largely due to two factors: 1) the fine-tuning strategy, which simplifies the training process by adapting pre-trained models to specific tasks, and 2) the use of the self-attention mechanism, which allows these models to effectively capture relationships

¹³<https://codalab.lisn.upsaclay.fr/competitions/18118>

between pairs of words (Mozafari, Farahbakhsh, and Crespi, 2019a; Mutanga, Naicker, and Olugbara, 2020).

Despite the encouraging results achieved with Transformer-based pre-trained representations in both unimodal and multimodal scenarios, several areas for improvement have been identified in these models (Mohammed and Ali, 2021; Khan et al., 2021, 2022; Zhang et al., 2023). Among the most pressing issues are: 1) the challenge of handling missing data, such as out-of-vocabulary words; 2) the lack of contextual information in domain-specific tasks, such as AL detection, where non-vulgar words can be used offensively to target individuals or groups; and 3) the growing number of parameters in pre-trained Transformer models, particularly in multimodal models, which significantly increases the computational cost of inference, pre-training, and retraining.

The first two challenges arise primarily because these pre-trained models are trained on general-purpose datasets and pre-training tasks (Devlin et al., 2019; Li et al., 2019a; Su et al., 2020), which limits the effectiveness of the self-attention mechanism in extracting relevant relationships between different features. Some approaches have attempted to address these limitations by training models using domain-specific data (Beltagy, Lo, and Cohan, 2019; Mohammed and Ali, 2021; Aragon et al., 2023). For example, in the context of AL detection in text, the HateBERT pre-trained model was introduced in Caselli et al. (2021), which was retrained using potentially offensive social media data. However, creating these domain-specific pre-trained models requires vast amounts of data and incurs high computational costs (Radford et al., 2019; Lu et al., 2019; Mohammed and Ali, 2021).

This doctoral research introduces the DA Mechanism to address these specific challenges while maintaining low computational costs by avoiding retraining. This mechanism incorporates contextual information specific to the training task into the self-attention mechanism during the fine-tuning stage. The central idea of this mechanism extends to address the third challenge by proposing an extension of the DA Mechanism to a cross-modal perspective. This extension aims to achieve a more effective alignment between image and text features by creating an architecture that replaces multimodal pre-trained models. It does so by combining the unimodal representations of text and image, thereby significantly reducing the number of parameters typically required by multimodal pre-trained models, all without compromising performance.

Proposed Dual Attention Mechanism

This chapter introduces the proposed DA mechanism, as well as its extension to a multi-level perspective to leverage the multiple encoding levels present in current Transformer-based architectures. As an initial step in this doctoral research, we previously proposed an initial version of the DA mechanism, which focused on the early fusion of features obtained from the SA and CA mechanisms. This mechanism was named the Self-Contextualized Attention (SCA) mechanism. Unlike the SCA mechanism, the DA mechanism performs a cross-level combination of SA and CA representations derived from an encoded matrix. For more details on the initial version of this mechanism, we refer the reader to the following paper (Jarquín-Vásquez, Escalante, and Montes, 2021).

This chapter is divided into five sections. Section 1 introduces the DA mechanism and the adapted architectures used for its evaluation. Section 2 presents the multi-level DA architectures along with their respective adapted approaches for its evaluation. Following this, Section 3 describes the evaluation datasets and implementation details. Section 4 presents the quantitative results of the proposed approaches. Finally, Section 5 provides a qualitative analysis of the results obtained with the proposed DA mechanism.

4.1. Dual Attention Mechanism

This section is divided into the following subsections: in the first subsection, we introduce the proposed DA mechanism for the incorporation of distinctive (contextual)

and local (self) sequential information in encoding features. The second subsection presents the adapted architectures for the evaluation of the DA mechanisms in detecting AL in text and memes.

4.1.1. Construction of the Dual-Attention Mechanism

This subsection introduces the proposed DA mechanism. This mechanism can be applied to any sequence of encoding features H . For the sake of explanation, each element of the sequence is represented by the word/token/image region encoding features h_i , which are extracted from a deep neural network, e.g. either from the hidden states of a RNN or the encoding representations of a Transformer neural network.

Given a sequence of encoding features $H = \{h_1, h_2, \dots, h_n\}$, where $H \in \mathbb{R}^{d \times n}$, where d represents the dimensionality of the encoding features, n is the number of elements in the sequence and h_i refers to the i -th encoding element (either a word, a token, or an image region) of H , the purpose of our proposed DA mechanism is to generate a global *context-aware* representation $G \in \mathbb{R}^{d \times n}$, that considers both the internal (self) and external (contextual) relationships between the encoding features of H . Figure 4.1 illustrates the general architecture of our proposed DA mechanism. This architecture is divided into three major stages, each of them is illustrated by the top 3 rectangles in Figure 4.1, corresponding to the *SA*, *CA* and *DA* stages. In order to unify the proposed mechanism with the current literature Vaswani et al. (2017), as a first step, different linear projections of the encoding features H were obtained, with the intention of capturing different representations of H , while maintaining their same dimensionalities. Specifically, the following matrices were obtained: a matrix of queries Q , two matrices of keys (K_s and K_c), and one matrix of values V . As in Vaswani et al. (2017), the intuition behind the use of these matrices is inspired by the information retrieval systems, where a similarity matrix is obtained through the use of a dot product between the matrix of keys K and queries Q , this to obtain the relevance between the pairs of elements of the sequence, finally, a new representation is obtained with the multiplication of the similarity matrix and a matrix of values V . Unlike Vaswani et al. (2017), in our proposed DA mechanism we use two different key matrices (K_s and K_c) due to the integration of the SA and CA mechanisms. The aforementioned linear-projected matrices are used as input to the different stages of

the proposed DA mechanism, each of the following stages is described in detail below.

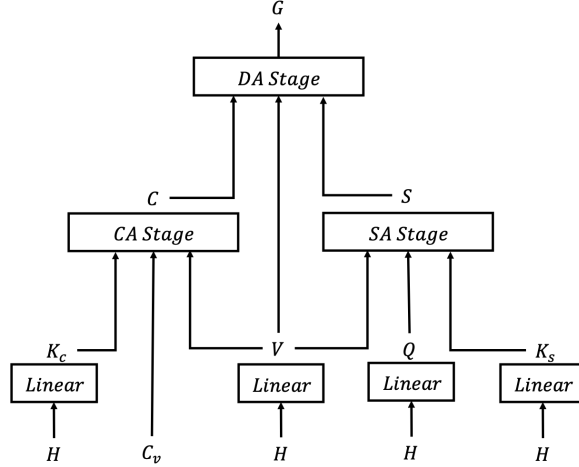


Figure 4.1: General Visualization of our Proposed Dual Attention Unit.

The first stage is the **CA stage**, it uses a *context* vector $C_v \in \mathbb{R}^d$, this vector is randomly initialized and jointly learned during the training process, C_v is used as a query vector in order to obtain the attention values $\alpha_c \in \mathbb{R}^n$ by measuring the similarity between the projected elements of the sequence K_c and the application domain represented by C_v . This similarity, defined in Equation 4.1.1, is calculated by the scalar dot product of C_v^T and K_c ; the resulting values are smoothed with the use of a softmax function. Contrasting the CA mechanism proposed by Yang et al. (2016), instead of using a weighted sum between each attention value and its corresponding encoding features for the final sequence representation, our context-aware representation $C \in \mathbb{R}^{d \times n}$ shown in Equation 4.1.2, takes all the information of the attention values, by doing an element-wise multiplication \odot , within each scalar of α_c and its corresponding projected encoding feature V_i . The use of element-wise multiplication allows for the generation of a matrix representation, enabling the combination of the CA and SA stages, in comparison to the original mechanism, which generates an output vector through a weighted sum.

$$\alpha_c = \text{softmax}(C_v^T \cdot K_c) \quad (4.1.1)$$

$$C = \alpha_c \odot V \quad (4.1.2)$$

The second stage is the **SA stage**, as in Pavlopoulos, Malakasiotis, and Androtsopoulos (2017); Vaswani et al. (2017) the main purpose of SA is the building of connections within the elements of the same sequence, but at different positions. The use of SA allows the modeling of both long-range and local dependencies, this is captured by the attention filter $\alpha_s \in \mathbb{R}^{n \times n}$ defined in the Equation 4.1.3. This attention filter is calculated by the dot product similarity between all the pairs of projected elements of Q and K_s , later these values are smoothed with the use of a softmax function. Finally, the context-aware representation $S \in \mathbb{R}^{d \times n}$ shown in the Equation 4.1.4, is calculated with the matrix multiplication of α_s^T and V , where α_s is used to highlight and filter out the most and less relevant projected encoding features, respectively.

$$\alpha_s = \text{softmax}(Q \cdot K_s^T) \quad (4.1.3)$$

$$S = \alpha_s^T V \quad (4.1.4)$$

The third stage corresponds to the **DA stage**, whose purpose is to merge these representations in order to create a global context-aware representation $G \in \mathbb{R}^{d \times n}$ that integrates both, the internal and external relationships. These relationships are captured with the global attention filter $\alpha_g \in \mathbb{R}^{n \times n}$, which is calculated by the smoothed dot product similarity between C and S , as shown in Equation 4.1.5. This attention filter can be seen as a high-level attention representation, since its calculation is based on the relevance of local and contextual extracted features, of both previously defined attention mechanisms. Finally, the global context-aware representation G is calculated in Equation 4.1.6 with the matrix multiplication of V and the attention filter matrix α_g ; the resulting matrix is normalized by multiplying it by the scalar $\frac{1}{\sqrt{d}}$, as proposed in Vaswani et al. (2017).

$$\alpha_g = \text{softmax}(C^T \cdot S) \quad (4.1.5)$$

$$G = \frac{V \alpha_g^T}{\sqrt{d}} \quad (4.1.6)$$

Figure 4.2 presents the extended visualization of the previously explained DA mechanism. As can be seen, the three different stages are connected by the linear projection V obtained from the initial encoding sequence H , this with the intention

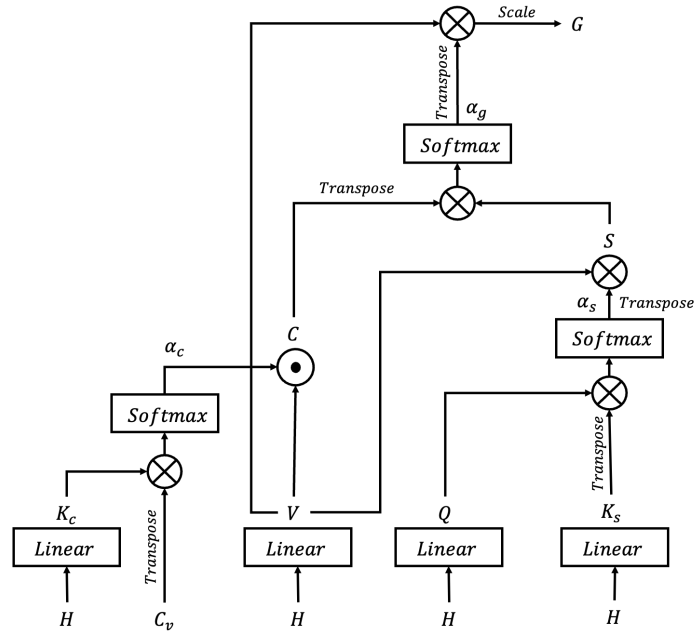


Figure 4.2: Extended Visualization of our Proposed Dual Attention Unit, where the \otimes symbol denotes matrix multiplication and the \odot symbol denotes element-wise multiplication.

of unifying the obtained representation in the CA, SA, and DA stages. The general equation of the proposed DA mechanism is presented in Equation 4.1.7. Since all the operations in our proposed DA mechanism are differentiable, this model can be easily coupled with other neural network architectures.

$$G = \frac{V \text{softmax}((\text{softmax}(C_v^T \cdot K_c) \odot V)^T \cdot (\text{softmax}(Q \cdot K_s^T)^T V))^T}{\sqrt{d}} \quad (4.1.7)$$

4.1.2. Adapted Architectures for the Evaluation of the DA Mechanism

Since the proposed DA mechanism can be applied to any sequence of encoded features, two neural network architecture approaches were adapted according to its use and outstanding performance in the AL detection task (Yang et al., 2016; Chakrabarty, Gupta, and Muresan, 2019; Nikolov and Radivchev, 2019; Mozafari, Farahbakhsh, and Crespi, 2019b). The first architecture, based on Recurrent Neural Networks (RNNs), is used for AL detection in text, while the second architecture, based on Transformers, is employed for detecting AL in both text and memes. These archi-

tectures are illustrated in Figure 4.3, where the left-hand side figure illustrates the RNN-based architecture, and the right-hand side figure illustrates the Transformer-based architectures, respectively.

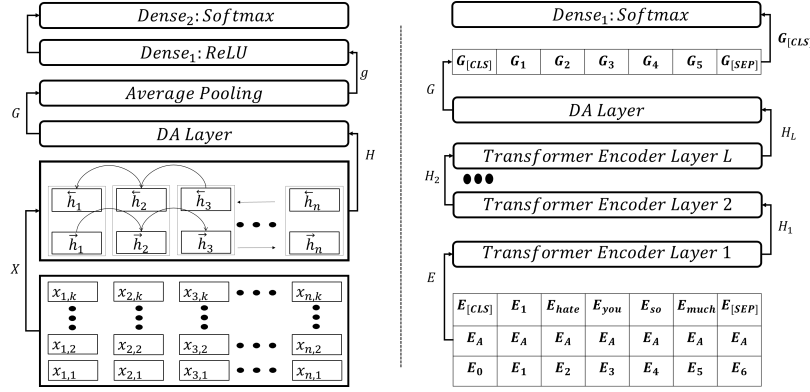


Figure 4.3: Adapted architectures for the integration of the proposed DA mechanism, both architectures integrate the proposed DA mechanism at the last encoding level. The left-hand sided architecture is based on the Bi-GRU network, on the other hand, the right-hand sided one is based on Transformer NNs.

As illustrated, the RNN-based architecture receives as input an embedding matrix $X \in \mathbb{R}^{k \times n}$, which is represented by a sequence of n k -dimensional word vectors x_i . Subsequently, the embedding matrix X passes as input to the encoding layer, which is conformed by a Bidirectional RNN layer, specifically, we used a Bidirectional Gated Recurrent Unit (Bi-GRU) layer Chung et al. (2014). The Bi-GRU layer accomplishes the sequence encoding task by summarizing the information of the whole sequence X centered around each word annotation; the producing encoding layer generates a sequence of encoding features $H \in \mathbb{R}^{d \times n}$, where d denotes the encoding dimensions. Since not all words contribute equally to the meaning and representation of a sequence, the sequence encoded features H are passed as input to the proposed DA mechanism, which generates a global context-aware representation G ; since the next layers of this architecture are conformed by the classification layers, the matrix G is reduced with an average pooling layer, generating a high-level representation vector $g \in \mathbb{R}^d$, which summarizes the most relevant information from G . Finally, the classification layers receive the representation vector g as input, specifically two layers handle the final classification, a dense layer with a Rectified Linear Unit (ReLU) activation function, and a fully-connected softmax layer to obtain the class probabilities and get the final classification.

Regarding the Transformer-based architecture, we chose to adapt the pre-trained BERT_{BASE}¹ model (with 12 layers, 768 hidden units, and 12 attention heads per layer) for the task of detecting AL in text. This model was selected due to its strong performance across a wide range of NLP tasks, including AL detection (Nikolov and Radivchev, 2019; Mozafari, Farahbakhsh, and Crespi, 2019b). For the task of detecting AL in memes, we adapted the pre-trained VisualBERT² model, which also consists of 12 layers, 768 hidden units, and 12 attention heads per layer. VisualBERT was chosen based on its effective performance in multimodal classification tasks, including meme classification (Kiela et al., 2020).

Both adapted architectures receive a sequence of n elements as input. In each model, the first token in the sequence represents the classification token ([CLS]). In the BERT model, the sequence elements are represented by tokens derived from textual input. In contrast, for VisualBERT, the sequence elements are composed of text tokens followed by encoded image regions, allowing for a multimodal representation. In both pre-trained models, the input sequence is initially processed by the embedding layer of the Transformer neural network, generating an embedding matrix $E \in \mathbb{R}^{k \times n}$, where k represents the dimensions of the embeddings. The embedding matrix E , passes as input to the Transformer encoding layers, where we obtain the last encoding layer H_L and pass it as input to the DA layer, in order to contextualize all the sequence representation into the matrix G . Finally, the first column vector of G ($G_{[CLS]}$) passes as input to a fully-connected softmax layer to obtain the class probabilities and get the final classification. Section 4.3.4 provides the implementation details for all the proposed and adapted architectures, including the hyperparameters and the size of all architectures.

4.2. Multi-Level Dual Attention

Inspired by the outstanding results obtained from the multiple levels of encoding representations in deep neural networks (Alzubaidi et al., 2021; Abdel-Jaber et al., 2022), we propose the extension of our proposed DA mechanism from a multi-level perspective architecture. This section is divided two-folded, the first subsection introduces our proposed Gated Hierarchical Attention (GHA) architecture, which in-

¹https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-12_H-768_A-12/1

²https://huggingface.co/docs/transformers/model_doc/visual_bert

tegrates the DA mechanism into different levels of encoding features from a weighted perspective. Correspondingly, the second subsection presents the adapted architectures for the evaluation of the GHA architecture.

4.2.1. Gated Hierarchical Attention Architecture

Recent studies have shown that the contributions of the different encoding levels in deep neural networks, specially the Transformer-based ones contribute differently depending on the specific instance inputs and tasks (Clark et al., 2019). We hypothesize that the relevance of words and image regions in certain contexts may have a better interpretation at certain encoding levels, thus aiming to capture these patterns, we propose a weighted fusion scheme that combines the DA mechanism obtained representations from a multi-level perspective.

In order to combine in a weighted manner the representations obtained from the use of multiple DA mechanisms, we proposed the adaptation of the Gated Multimodal Unit (GMU) (Arevalo et al., 2020) into a multi-level architecture. Originally, the GMU was designed to fuse information coming from distinct data modalities (e.g., text and images) to produce an intermediate representation. However, in our adaptation, rather than fusing data from different modalities, the GMU is adapted to combine the intermediate representations produced by the DA mechanism at multiple encoding levels of a deep neural network. In other words, these multi-level features are treated as if they were separate modalities, allowing the GMU to learn a weighted fusion of the hierarchical representations. This design enables our architecture to capture richer contextual information by integrating features from various layers. Figure 4.4 illustrates the proposed GHA architecture.

As illustrated in Figure 4.4, the GHA architecture is designed to be coupled with deep neural networks with multiple levels of encoding features, specifically, we will focus our adaptation on stacked-RNN (Chakrabarty, Gupta, and Muresan, 2019; Lan et al., 2020) for detecting AL in text, and Transformer-based neural networks (Vaswani et al., 2017; Devlin et al., 2019; Li et al., 2019a) for detecting AL in both text and memes.

This architecture receives as input a pre-trained embedding layer $E \in \mathbb{R}^{k \times n}$, where k represents the dimensionality of the embeddings and n represents the number of elements in the input sequence, either represented by words or by words and image

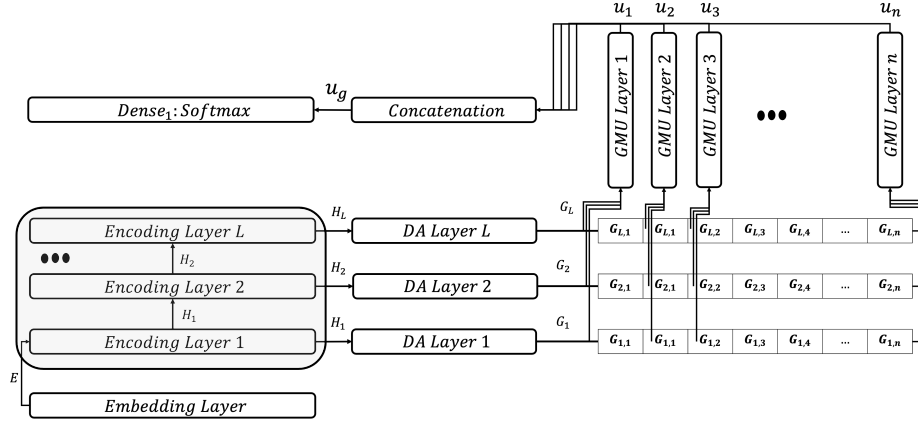


Figure 4.4: Illustration of the Proposed GHA Architecture, this architecture is designed to be applied to any stacked-based encoding architecture (e.g. RNN and Transformer-based architectures).

regions for AL detection in memes. This embedding matrix E passes as input to the multi-level encoding-based architecture, specifically L levels encode the input matrix E , this generates L different encoding representations ($H_1, H_2, H_3, \dots, H_L$). Each generated encoding representation $H_i \in \mathbb{R}^{d \times n}$ is contextualized by our DA mechanism, specifically, an arrangement of L different DA mechanism layers matched its corresponding encoding layer, that is, the H_i encoding representation passes as input to the i -th DA layer, this generates L different context-aware representations ($G_1, G_2, G_3, \dots, G_L$), where $G_i \in \mathbb{R}^{d \times n}$ maintains the same dimensionalities of the previous codification.

As shown in Figure 4.4, each matrix G_i captures the contextual information of each sequence element across the L encoding levels. In order to create an intermediate representation of these contextualized representations, we use an arrangement of n different GMU layers, where each layer learns an intermediate representation $u_i \in \mathbb{R}$ for the i -th sequence element at the L different encoding levels, these intermediate representations are concatenated in order to create an intermediate context-aware representation $u_g \in \mathbb{R}^n$, as shown in Equation 4.2.1. Finally, the vector u_g passes as input to a fully-connected softmax layer to obtain the class probabilities and get the final classification.

$$u_g = [u_1, u_2, u_3, \dots, u_n] \quad (4.2.1)$$

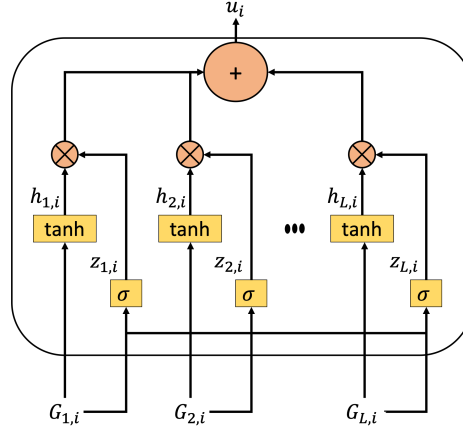


Figure 4.5: Illustration of the adapted Gated Multimodal Unit (GMU) for the intermediate representation of the encoding levels in RNN and Transformer-based neural networks. Each GMU layer is applied to an individual word/token/image region, generating an intermediate representation with respect to its L different encoding layers. This Figure is inspired by Arevalo et al. (2020).

Figure 4.5 presents the overall architecture of the adapted GMU layer, as illustrated, each GMU layer receives as input the different encoding levels of a specific word/token/image region $(G_{1,i}, G_{2,i}, \dots, G_{L,i})$, where $G_{j,i} \in \mathbb{R}^d$ represents the j -th encoding level features of the i -th sequence element. To obtain the intermediate representation u_i , the GMU extracts hidden features for each encoding level $G_{j,i}$, as shown in Equation 4.2.2, where $W_i \in \mathbb{R}^d$ are learnable weights, \tanh is the default activation function and, $h_{j,i} \in \mathbb{R}$ is the resultant hidden representation of the $G_{j,i}$ encoding representation. Aiming to capture the relevance of the different encoding levels, the GMU contains a third internal feature $z_{j,i} \in \mathbb{R}$, this relevance is captured through equation 4.2.3, where $[\cdot]$ denotes the concatenation operator, $W_{z_i} \in \mathbb{R}^{d_1+\dots+d_L}$ are the learnable weights and σ represents the sigmoid activation function. Finally, the intermediate representation u_i (defined in Equation 4.2.4) is obtained through a weighted sum between the product of each hidden representation $h_{j,i}$ and its corresponding relevance $z_{j,i}$. For more details of the GMU, we refer the reader to the following paper (Arevalo et al., 2020).

$$h_{j,i} = \tanh(W_i G_{j,i}) \quad (4.2.2)$$

$$z_{j,i} = \sigma(W_{z_i} [G_{1,i}, G_{2,i}, \dots, G_{L,i}]) \quad (4.2.3)$$

The different encoding levels of the adapted architecture can be obtained with an arraignment of stacked layers of an RNN or with the use of a Transformer-based approach. Specifically, this architecture receives an embedding matrix $X \in \mathbb{R}^{k \times n}$ as input, where k represents the dimensions of the embedding and n represents the number of elements of the input sequence, this embedding matrix passes as input to the first encoding layer. Regarding the obtained encoding representations of all the encoding layers (H_1, H_2, \dots, H_L), each output representation H_i passes as input to: 1) the next encoding layer H_{i+1} (with the exception of the last encoding layer), and 2) to the DA_i layer, where we obtain a context-aware representation G_i for each encoding layer. All contextualized representations (G_1, G_2, \dots, G_L) pass through the Addition and Normalization (Add and Norm) layer to obtain an intermediate representation G of all the contextualized representations, while maintaining the same encoding dimensions. Unlike the Add and Norm layer proposed by Vaswani et al. (2017), instead of having a residual connection, our implementation sums all the representations G_i . Finally, the matrix G is reduced with the average pooling layer, generating a high-level representation vector $g \in \mathbb{R}^d$, which summarizes the most relevant information from the previous matrix G . Vector g is received as input to a fully-connected softmax layer to obtain the class probabilities and get the final classification.

4.3. Evaluation and Implementation Details

This section presents the implementation details for the proposed DA mechanism and GHA architecture. This section is divided into four subsections, the first subsection introduces the evaluation datasets for the detection of AL in text and memes; the second subsection discusses the evaluation metrics for evaluating our proposed approaches in the detection of AL in text and memes. The third subsection presents the adapted baseline architectures used to evaluate the proposed DA mechanism in AL detection for text and memes. Finally, the fourth subsection provides the implementation details, regarding the text pre-processing phase, the model hyperparameters, and the used libraries for the implementation of the proposed approaches.

4.3.1. Evaluation Datasets for the Detection of Abusive Language

AL can be of different types, its main divisions are distinguished by the target and severity of the insults (Mandl et al., 2019). Accordingly, different collections and evaluation campaigns have considered different kinds of AL for their study (Schmidt and Wiegand, 2017; MacAvaney et al., 2019). The following two subsections provide a brief description of the six English evaluation datasets used in our experiments for AL detection in text and the three datasets used in our experiments for AL detection in memes.

Datasets for Abusive Language Detection in Text

The first three datasets: *Waseem* (Zeeraak and Dirk, 2016), *Davidson* (Davidson et al., 2017) and *Golbeck* (Golbeck et al., 2017) were some of the first large-scale datasets for abusive tweet detection. Specifically, the *Davidson* dataset focuses on the identification of offensive language and hate speech in tweets, the *Waseem* dataset focuses on the identification of racist and sexist tweets; whereas the *Golbeck* dataset focuses on the detection of harassment in tweets.

On the other hand, the *SE 2019 T 6* (Marcos et al., 2019) and *AMI 2018* (Fersini, Nozza, and Rosso, 2018) datasets were presented at the *SemEval-2019 Task 6*, and at the *Evalita 2018* Task on Automatic Misogyny Identification (AMI) respectively. The *SE 2019 T 6* dataset focuses on identifying offensive tweets, whereas the *AMI 2018* dataset focuses on identifying misogyny in tweets. Finally, the *HASOC 2019* (Mandl et al., 2019) dataset was presented at the *11th Forum for Information Retrieval Evaluation (FIRE)*, in the Hate Speech and Offensive Content Identification (HASOC) shared-task, where the main goal is the classification of Hate Speech and non-offensive online content in Indo-European Languages. Although these shared tasks encompass a variety of evaluation subtasks and languages, our experiments focus solely on binary classification in English.

The selection of these datasets is based on their well-established annotation guidelines and/or moderate to strong inter-annotator agreement. The *Waseem* dataset has an inter-annotator agreement of 0.57 across its three classes, while the *Golbeck* dataset shows a Cohen’s Kappa of 0.84 between harassment and non-harassment. The *AMI 2018* dataset reports an agreement of 0.81 for misogyny, and the *SemEval*

2019 task 6 dataset has a Fleiss' kappa of 0.83 for offensive and non-offensive classes in a trial set. The *HASOC 2019* dataset shows an agreement of 0.77 for hate speech detection in English. Although the *Davidson* dataset lacks specific agreement metrics, it is widely recognized for its large size and clear distinction between hate speech and offensive language.

Figure 4.7 presents the classes distributions of the six evaluation datasets; as shown, there is a great class imbalance, where in most cases the least abundant class is the one containing some form of AL.

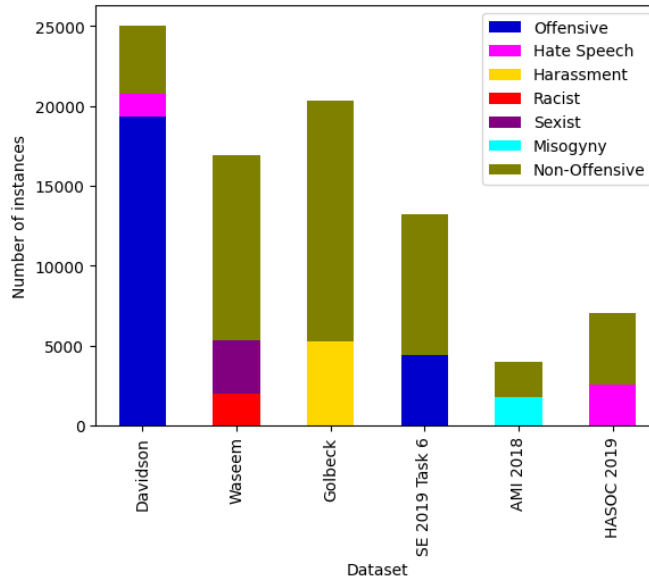


Figure 4.7: Classes distribution of the evaluation datasets for the detection of AL in text, this Figure presents the distribution of all the training sets.

Datasets for Abusive Language Detection in Memes

Regarding the datasets focused on AL detection in memes, we utilized three different collections. The first collection is the dataset from the Hateful Memes Challenge³ (HMC), organized by Meta⁴ (Kiel et al., 2020). This dataset targets hate speech detection in English-language memes and comprises a collection of 10K instances.

The second collection is the Multimedia Automatic Misogyny Identification⁵ (MA-

³<https://ai.meta.com/blog/hateful-memes-challenge-and-data-set/>

⁴<https://ai.meta.com/>

⁵<https://competitions.codalab.org/competitions/34175>

MI), introduced in Task 5 of SemEval 2022 (Fersini et al., 2022). This dataset focuses on detecting misogynistic content in English-language memes and includes two subtasks. The first subtask involves basic classification, where a meme is labeled as either misogynistic or non-misogynistic. The second subtask is more advanced, requiring the identification of specific types of misogyny within overlapping categories such as stereotype, shaming, objectification, and violence. This dataset contains a total of 11K instances, with 10K instances for training and 1K instances for testing.

The third dataset, Detection of Inappropriate Memes from Mexico⁶ (DIMEMEX), is a Spanish-language meme dataset developed by us and introduced as a shared-task at IberLEF 2024 (Jarquín-Vásquez et al., 2024). This shared task comprises two subtasks: the first subtask is a three-way classification to distinguish between hate speech, inappropriate content, and harmless content; the second subtask involves a fine-grained classification to distinguish different types of hate speech, including sexism, racism, classism, and others. For a detailed description of this dataset’s construction, we refer readers to (Jarquín-Vásquez et al., 2024). Figure 4.8 presents the class distributions for the three evaluation datasets. As illustrated, there is a significant class imbalance in the HMC and DIMEMEX datasets, where, in most cases, the least frequent class is the one containing some form of AL.

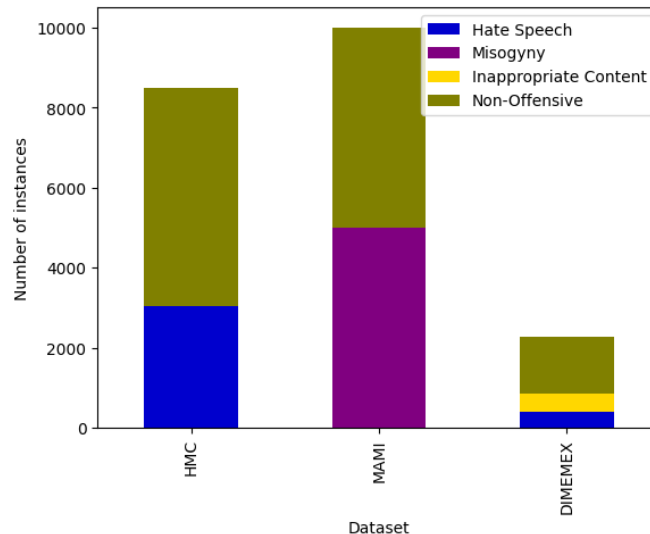


Figure 4.8: Classes distribution of the evaluation datasets for the detection of AL in memes, this Figure presents the distribution of all the training sets.

⁶<https://codalab.lisn.upsaclay.fr/competitions/18118>

To evaluate our proposed approaches on these datasets, we conducted binary classification experiments on the first two datasets. For the DIMEMEX dataset, however, we applied both a three-way classification approach and a finer-grained classification scheme based on the two subtasks. The first two evaluation datasets were chosen due to their extensive use and recognition in the literature (Hermida and Santos, 2023).

4.3.2. Evaluation Metrics for the Detection of Abusive Language

A wide variety of metrics have been employed to evaluate the performance of classifiers in the task of AL detection, both in unimodal and multimodal scenarios (Kumar et al., 2018; Kiela, Wang, and Cho, 2018). In order to make a fair comparison and evaluate the performance of our proposed DA mechanism, GHA architecture, and the adapted architectures, we used several evaluation metrics inspired by the aforementioned shared tasks and standard metrics in the AL detection domain. Specifically, the metrics used to evaluate our proposed approaches included accuracy, as well as the weighted and macro-average F_1 scores. In all experiments, we report the mean and standard deviation across five training runs, each initialized with different random seeds.

For AL detection in text datasets, such as the *Waseem*, *Davidson*, and *Golbeck* datasets, we evaluated performance using the weighted-average F_1 score to ensure fair comparison with state-of-the-art (SOTA) approaches. Regarding shared task datasets, we adhered to the evaluation metrics specified by the task organizers. Specifically, the *SemEval 2019 Task 6* and *HASOC 2019* datasets were evaluated using the macro-average F_1 score, while the *AMI 2018* dataset was evaluated using accuracy. For AL detection in memes, we used the evaluation metrics specified in the shared tasks. For the MAMI and DIMEMEX datasets, we used the macro-average F_1 score, as the objective of these shared tasks was to assign equal importance to all classes to ensure a balanced evaluation. In contrast, accuracy was used as the evaluation metric for the HMC dataset.

The *Waseem*, *Davidson*, and *Golbeck* datasets were split into 80% for training, 10% for validation, and 10% for testing. Results were reported on the test partition to maintain consistency with the partitioning practices of SOTA approaches discussed in the results section. For shared task datasets in both text and memes (*SE 2019 T6*,

AMI 2018, *HASOC 2019*, *HMC*, *MAMI*, and *DIMEMEX*), we used the test partition results provided by the task organizers to ensure a fair comparison.

To test for statistical significance in the performance improvements of our DA mechanism and GHA architecture for AL detection in text and memes, we utilized macro-average F_1 scores in a Bayesian Wilcoxon signed-rank test (Benavoli et al., 2017).

4.3.3. Adapted baselines for the Evaluation of the DA Mechanism

To compare the robustness of integrating the proposed DA mechanism into the *RNN*-based architectures and the $\text{BERT}_{\text{BASE}}$ model in the detection of AL in text, we consider three baseline architectures: the first is a simple Bi-GRU network that receives words as input but does not use any attention layer; the second employs a three-layer Bi-GRU stack without adding any attention layer; and the third is a fine-tuned BERT model without using any DA and GMU layers. As described in Devlin et al. (2019), we take the last encoding layer of the classification token $\langle \text{CLS} \rangle$ and use it as input for the softmax classification layer. These three baseline architectures are referred to in the experiment results as Bi-GRU, Bi-GRU_S, and $\text{BERT}_{\text{BASE}}$, respectively. It is important to mention that the first two baseline architectures used the same hyperparameter settings for the RNN-based architectures, while the third one uses the same settings for Transformer-based architectures.

To assess the robustness of integrating the DA mechanism into the VisualBERT model for AL detection in memes, we considered four baseline architectures. The first two baselines were adapted to measure the impact of the text and vision modalities in detecting AL in memes. The first baseline is a fine-tuned pre-trained language model without using any DA or GMU layers. Specifically, we used the $\text{BERT}_{\text{BASE}}$ model for English datasets (HMC and MAMI) and the BETO model⁷, a Spanish-adapted version of BERT, for the DIMEMEX dataset. The second baseline involves fine-tuning the pre-trained Vision Transformer⁸ (ViT) model. As with the text-based baseline, we take the last encoding layer of the classification token $\langle \text{CLS} \rangle$ and use it as input to the softmax classification layer.

⁷<https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased>

⁸https://huggingface.co/docs/transformers/model_doc/vit

The third baseline integrates both modalities (visual and textual) using an Early Fusion (EF) technique, which concatenates the classification vectors obtained from the BERT_{BASE}/BETO and ViT models. Finally, the fourth baseline is a fine-tuned VisualBERT model without employing any DA or GMU layers. Similar to other Transformer-based baselines, the last encoding of the classification token <CLS> is used as input to the softmax classification layer. These four baseline architectures are referred to in the experimental results as BERT_{BASE}/BETO, ViT, EF(BERT_{BASE}/BETO, ViT), and VisualBERT, respectively.

4.3.4. Implementation Details

This subsection provides the implementation details, including the neural network architecture configurations and hyperparameter settings. This subsection is divided into two parts: the first part presents the implementation details of the approaches for detecting AL in text, and the second part focuses on the implementation details for detecting AL in memes.

Implementation Details for Detecting AL in Text

Regarding the text preprocessing phase, different operations were applied: in order to avoid biases, user mentions and links were replaced by the default tokens: <user> and <url>; in order to enrich the vocabulary, all hashtags were segmented by words (*e.g.* #BuildTheWall - build the wall) with the use of the ekphrasis library (Baziotis, Pelekis, and Doukeridis, 2017); in addition to this, all emojis were converted into words using the demoji⁹ library. All text was lowercased and non-alphabetical characters, as well as consecutive repeated words, were removed. For the RNN-based approaches, we used pre-trained fastText embeddings (Mikolov et al., 2018) as word representation, trained with subword information on Common Crawl, which has been recognized as useful for this task (Corazza et al., 2020). All the text preprocessing steps were applied consistently across all evaluation datasets. NOTE: for some instances from the HASOC 2019 dataset that originated from Facebook, the user normalization operation was not applied due to the absence of the '@' symbol.

Concerning the hyperparameter settings of the adapted RNN-based architectures, Table 4.1 presents their configurations; all these architectures are based on a Bi-GRU

⁹<https://pypi.org/project/demoji/>

neural network, due to its great performance in the short-text encoding task (Yang et al., 2016; Chakrabarty, Gupta, and Muresan, 2019). The first version of this architecture uses one Bi-GRU layer, and one DA layer on top of the encoded representation of the Bi-GRU layer. The hierarchical version of this architecture uses three different Bi-GRU and DA layers, which maintain the same input and output sizes (as shown in Table 4.1). On the other hand, for the weighted fusion of all the contextualized levels extracted from the three DA mechanisms, via de GHA architecture, we used an arrangement of n GMU layer on top of each word multi-level representation, the three aforementioned approaches are referred in the evaluation results as DA, AHA, and GHA, respectively. These architectures were trained for a total of 15 epochs, with a learning rate of $1e-4$, using the Adam optimizer (Kingma and Ba, 2015), a Dropout rate of 15 %, and a batch size 32.

Vectors, Matrices and Variables		Size
	n	50
	k	300
	d	128
	C_v	128
	Q, K_c, K_s, V	50x128
Layer	Input size	Output size
Embedding	50	50x300
Bi-GRU _{i}	50x300	50x128
DA _{i}	50x128	50x128
GMU _{i}	3x128	1
Avg Pooling	50x128	128
Dense ₁	128	64
Dense ₂	64	#Classes

Table 4.1: Hyperparameters of the RNN-based encoding architectures.

The hyperparameter settings of the Transformer-based architectures are presented in Table 4.2. Regarding the text pre-processing phase, we kept the steps described above, in addition to this, we used the WordPiece¹⁰ tokenizer in order to tokenize each instance sentence, as described in Devlin et al. (2019). Regarding the integration of the DA mechanism into the last encoding layer on the pre-trained BERT_{BASE} model, no GMU, Avg Pooling, and Add-and-Norm layers were used; instead, we only used one DA layer on top of the last encoding layer of BERT. On the other

¹⁰<https://huggingface.co/course/chapter6/6?fw=pt>

hand, for the hierarchical integration of the DA mechanism at all the encoding levels of BERT, we used an arrangement of 12 DA mechanisms, an Add-and-Norm layer for the integration of all the previously contextualized encoding levels, and an Avg Pooling layer. For the weighted fusion of all the contextualized levels extracted from the DA mechanisms, we used an arrangement of n GMU layers on top of each token multi-level representation, as previously described in the RNN-based architectures, the three aforementioned approaches are referred in the evaluation results as DA, AHA, and GHA, respectively. All these architectures were trained for a total of 3 epochs, with a learning rate of $5e-5$, using the Adam optimizer, and a batch size of 32.

Vectors, Matrices and Variables		Size
n		70
d		768
C_v		768
Q, K_c, K_s, V		70x768
Layer	Input size	Output size
Embedding (BERT)	70	70x768
BERT _{i}	70x768	70x768
DA _{i}	70x768	70x768
Add-and-Norm	12x70x768	70x768
Avg Pooling	70x768	768
GMU _{i}	12x768	1
Concatenation	1,70	70
Dense ₁	70/768	#Classes

Table 4.2: Hyperparameters of the Transformer-based encoding architecture.

Implementation Details for Detecting AL in Memes

Regarding text preprocessing in memes, we applied all the operations described in the previous subsection across the three evaluation datasets. For the baseline that relies solely on textual information (using either the BERT_{BASE} model for English datasets or the BETO model for the Spanish dataset), a maximum token length n of 40 was used. This is in contrast to the 70 tokens used in baselines and proposed Transformer-based architectures for AL detection in text. The reduction in token length was due to the fact that the text in memes did not exceed 40 tokens across all training instances in the three evaluation datasets. Both models were trained for a

total of 2 epochs, with a learning rate of $5e-5$, using the Adam optimizer, a dropout rate of 15 %, and a batch size of 24.

For the ViT baseline, which uses only the visual modality, we employed the model configured with a patch size of 16 and an image resolution of 224×224 pixels. To utilize this model, the following preprocessing steps were applied to each image: all images were converted to RGB scale, and all images were resized to a resolution of 224×224 pixels. Since the ViT model processes all regions of the image based on the patch size, a total of 196 visual tokens were generated by dividing the 224×224 pixel image into regions of 16×16 pixels (patch size). These visual tokens were then used to create the input embedding matrix for the Transformer model, including the classification token [CLS] followed by the 196 visual tokens, resulting in an input sequence n of size 197. This baseline was trained for a total of 2 epochs, with a learning rate of $1e-5$, using the Adam optimizer, a dropout rate of 15 %, and a batch size of 24.

For the third baseline, based on the early fusion of text features (obtained from either BERT_{BASE} or BETO) and image features (obtained from the ViT model), the same text and image preprocessing steps as in the previous baselines were applied. A fine-tuning process was performed in parallel for the BERT_{BASE}/BETO model and the ViT model. During this process, the [CLS] classification vectors from both models were extracted and concatenated into a new vector of size 1536, which was then used as input to the softmax classification layer. This baseline was trained for a total of 3 epochs, with a learning rate of $1e-5$, using the Adam optimizer, a dropout rate of 15 %, and a batch size of 24.

The hyperparameter settings of the VisualBERT model configurations are presented in Table 4.3. Regarding the text pre-processing phase, we kept the steps described above. Regarding the integration of the DA mechanism into the last encoding layer on the pre-trained VisualBERT model, no GMU, Avg Pooling, and Add-and-Norm layers were used; instead, we only used one DA layer on top of the last encoding layer of VisualBERT. On the other hand, for the hierarchical integration of the DA mechanism at all the encoding levels of VisualBERT, we used an arrangement of 12 DA mechanisms, an Add-and-Norm layer for the integration of all the previously contextualized encoding levels, and an Avg Pooling layer. For the weighted fusion of all the contextualized levels extracted from the DA mechanisms, we used an arrangement of n GMU layers on top of each token multi-level representation. The three aforementioned approaches are referred in the evaluation results as DA, AHA, and

GHA, respectively.

Vectors, Matrices and Variables		Size
n		54
d		768
C_v		768
Q, K_c, K_s, V		54x768
Layer	Input size	Output size
Embedding (VisualBERT)	54	54x768
VisualBERT _i	54x768	54x768
DA _i	54x768	54x768
Add-and-Norm	12x54x768	54x768
Avg Pooling	54x768	768
GMU _i	12x768	1
Concatenation	1,54	54
Dense ₁	54/768	#Classes

Table 4.3: Hyperparameters of the VisualBERT encoding configurations.

For all configurations based on the VisualBERT model, a maximum input length of 54 tokens was used. This length was determined by allocating 40 tokens to the textual inputs, including the classification token, while the remaining 14 tokens corresponded to image regions associated with objects detected by the VisualBERT model. To determine the optimal number of objects, we experimented with various configurations for the number of detected objects. The best results were achieved when up to 14 objects were recognized in the images. All these architectures were trained for a total of 3 epochs, with a learning rate of $1e-5$, using the Adam optimizer, and a batch size of 24. For the DIMEMEX dataset, since the text was in Spanish, automatic translation was performed using GPT-3.5 Turbo¹¹.

Note: All hyperparameters for the various proposed configurations for detecting AL in text and memes were selected through a grid search, taking the best values from three random training runs. The following values were considered: for the learning rate $\{1e-4, 5e-5, 1e-5\}$; for the number of epochs, Transformer-based architectures were tested with $\{1, 2, 3, 4\}$, and RNN-based architectures with $\{13, 14, 15, 16\}$; finally, for the batch size, the configurations $\{16, 24, 32, 40\}$ were evaluated. All baseline models, along with the proposed configurations of the DA mechanism and its hierarchical adaptation for the detection of AL in text and memes, were evaluated

¹¹<https://platform.openai.com/docs/models#gpt-3-5-turbo>

on a computer equipped with an NVIDIA GTX 1080 Ti graphics card, 32 GB of DDR4 RAM, and an Intel Core i7-7820X processor.

4.4. Quantitative Results

This section presents the quantitative results of the evaluation of the proposed approaches across six evaluation datasets for detecting AL in text, and three datasets for detecting AL in memes. The section is divided into four subsections. The first subsection presents the results of integrating the proposed DA mechanism into different encoding architectures, as described in Subsection 4.1.2. The second subsection reports the evaluation results of the proposed multi-level DA architectures, including the GHA architecture, using different encoding architectures and compares the obtained results with state-of-the-art (SOTA) approaches. To gain insights into the effectiveness of the proposed DA mechanism, the third subsection provides an analysis of complementarity and error diversity of the DA mechanism compared to the use of SA and CA mechanisms. Finally, the fourth subsection provides a statistical significance analysis of the proposed approaches.

4.4.1. Effectiveness of the Proposed Dual Attention Mechanism

To analyze the effectiveness of the proposed DA mechanism in the detection of AL in text, as a first experiment we proposed the evaluation of its integration on the last encoding layer of the following encoding architectures: Bi-GRU, Bi-GRU_S and BERT_{BASE}. As baselines, we considered these architectures without the integration of the DA mechanism; Table 4.4 reports the mean and standard deviation of the evaluation results of this first experiment.

Focusing on the obtained results with the Bi-GRU architecture and its integration with the DA mechanism (rows 2 and 3), better results are obtained in all datasets with the integration of the DA mechanism into the Bi-GRU architecture, obtaining an improvement of up to 8.4%. Concerning the obtained results by the Bi-GRU_S architecture and its integration with the DA mechanism (rows 4 and 5), a consistent increase in performance is obtained in all the evaluation datasets, with the integration of the DA mechanism into the Bi-GRU_S architecture, obtaining an improvement of

AM	EA	Waseem	Davidson	Golbeck	SE 2019 T 6	AMI 2018	HASOC 2019
-	Bi-GRU	0.813±0.0078	0.904±0.0093	0.682±0.0091	0.735±0.0105	0.582±0.0086	0.679±0.0097
DA	Bi-GRU	0.853±0.0065	0.912±0.0061	0.715±0.0074	0.762±0.0054	0.631±0.0061	0.726±0.0075
-	Bi-GRU _S	0.824±0.0063	0.914±0.0058	0.675±0.0081	0.742±0.0070	0.596±0.0068	0.691±0.0064
DA	Bi-GRU _S	0.862±0.0083	0.919±0.0082	0.715±0.0084	0.751±0.0071	0.630±0.0081	0.717±0.0097
-	BERT _{BASE}	0.853±0.0073	0.921±0.0081	0.707±0.0094	0.772±0.0079	0.696±0.0063	0.752±0.0077
DA	BERT _{BASE}	0.864±0.0074	0.926±0.0086	0.726±0.0090	0.783±0.0071	0.714±0.0065	0.763±0.0074

Table 4.4: Comparison results from our three baseline architectures, and our proposed Dual Attention mechanism variants in six datasets for the AL detection task in text. The *Waseem*, *Davidson*, and *Golbeck* datasets were evaluated with the weighted-average F_1 score, the *SemEval 2019 task 6* and *HASOC 2019* datasets were evaluated using the macro-average F_1 score, finally, the *AMI 2018* dataset was evaluated using the accuracy. Note that “AM” and “EA” refer to Attention Mechanism and Encoding Architecture, respectively.

up to 5.9%. Regarding the analysis of results in the BERT_{BASE} architectures (rows 6 and 7), better results are obtained with the integration of the DA mechanism into the last encoding layer of the BERT_{BASE} model, by obtaining an improvement of up to 2.6%. The overall improvement with the integration of the DA mechanism in all baseline architectures is consistent in all datasets (row 2 vs 3, row 4 vs 5, and row 6 vs 7).

Centering the analysis of results on the three baseline architectures (rows 2, 4, and 6), the results indicate that the use of Transformer models outperforms the use of RNN-based architectures in all datasets by a wide margin of up to 13.1%; this may be due to the large number of parameters and pre-trained information of the BERT model. On the other hand, regarding the addition of the proposed DA mechanism into the three encoding architectures (rows 3, 5, and 7), the greatest improvement is obtained with the integration of the DA mechanism in the Bi-GRU architecture, obtaining a maximum improvement of 8.4%; this may be due to the fact that the BERT model integrates the SA mechanism, in contrast to the Bi-GRU architecture; which produces a greater improvement in its integration with the DA mechanism. Finally, the best results are obtained with the integration of the DA mechanism into the BERT_{BASE} model (row 7 vs. rows 3, and 5).

To analyze the effectiveness of the proposed DA mechanism in detecting AL in memes, we conducted an initial experiment evaluating its integration into the final encoding layer of the VisualBERT model. As a baseline, we considered the fine-tuning of the VisualBERT model without integrating our DA mechanism. Additionally, we evaluated the exclusive use of the text modality by fine-tuning the BERT and BETO models and the exclusive use of the image modality by fine-tuning the ViT model. To measure the complementarity between these two modalities, we also considered the early fusion of the classification vectors obtained from the BERT/BETO and ViT

models. Table 4.5 presents the mean and standard deviation of the evaluation results for this first experiment.

AM	EA	HMC	MAMI	DIMEMEX ST1	DIMEMEX ST2
-	BERT/BETO	0.601±0.0084	0.623±0.0091	0.467±0.0063	0.272±0.0081
-	ViT	0.564±0.0079	0.616±0.0086	0.431±0.0094	0.286±0.0109
-	EF	0.621±0.0063	0.654±0.0074	0.485±0.0081	0.296±0.0065
-	VisualBERT	0.663±0.0096	0.685±0.0117	0.494±0.0103	0.316±0.0099
DA	VisualBERT	0.695±0.0081	0.729±0.0077	0.528±0.0080	0.342±0.0094

Table 4.5: Comparison results of our four baseline architectures and the proposed integration of the Dual Attention mechanism into VisualBERT across three datasets for AL detection in memes. The MAMI and DIMEMEX datasets were evaluated using the macro-average F_1 score, while the HMC dataset was evaluated using accuracy. Note that “AM” and “EA” refer to Attention Mechanism and Encoding Architecture, respectively.

Regarding the results obtained when evaluating the exclusive use of the text and image modalities (rows 2 and 3), better performance was observed in 3 out of 4 datasets with the text modality, achieving improvements of up to 6.5%. This indicates that the text modality is more effective in distinguishing instances of AL in memes. For the DIMEMEX dataset in Subtask 2, better results were achieved with the vision modality; however, the margin of improvement was low. When early fusion was used, consistent improvements were observed across all datasets, surpassing the results obtained with the independent use of the text and image modalities (row 4 vs. rows 2 and 3), with an improvement of up to 10.1%. This demonstrates the complementarity of both modalities and highlights the necessity of utilizing both for effective AL detection in memes.

Table 4.5 also compares the use of VisualBERT, a vision & language model trained with multimodal data, against the early fusion approach of unimodal models (row 5 vs. row 4). Consistent improvements were observed with VisualBERT across all datasets, with performance gains of up to 6.7%. This highlights the advantage of using multimodal models over the early fusion of unimodal models. Finally, the results of integrating the proposed DA mechanism into the VisualBERT model are presented. Comparing the performance of the DA-enhanced VisualBERT model to the standalone VisualBERT model (row 6 vs. row 5), consistent improvements were achieved across all evaluation datasets, with gains of up to 6.4%. Overall, the best results were obtained with the integration of the DA mechanism into the VisualBERT model.

4.4.2. Effectiveness of the Proposed Multi-Level Dual Attention Architectures

As a second evaluation step, and aiming to evaluate the performance of the hierarchical integration of the DA mechanism into the encoding layers of the Bi-GRU_S, BERT_{BASE}, and VisualBERT encoding architectures, this subsection covers the evaluation of the proposed GHA and AHA architectures. Table 4.6 reports the mean and standard deviation results of this evaluation in the detection of AL in text.

AM	EA	Waseem	Davidson	Golbeck	SE 2019 T 6	AMI 2018	HASOC 2019
AHA	Bi-GRU _S	0.871 ±0.0074	0.924 ±0.0065	0.721 ±0.0059	0.764 ±0.0078	0.641 ±0.0063	0.736 ±0.0070
GHA	Bi-GRU _S	0.876 ±0.0085	0.935 ±0.0094	0.727 ±0.0104	0.771 ±0.0085	0.678 ±0.0080	0.753 ±0.0079
AHA	BERT _{BASE}	0.883 ±0.0073	0.939 ±0.0082	0.731 ±0.0063	0.802 ±0.0080	0.725 ±0.0067	0.776 ±0.0096
GHA	BERT _{BASE}	0.895 ±0.0084	0.942 ±0.0073	0.736 ±0.0085	0.824 ±0.0097	0.732 ±0.0071	0.781 ±0.0089
-	SOTA	0.880 ¹²	0.920 ¹²	0.727 ¹³	0.829 ¹⁴	0.704 ¹⁵	0.788 ¹⁶

Table 4.6: Comparison results from our two baseline architectures, our proposed GHA architecture, and state-of-the-art approaches in six datasets for the detection of AL in text.

Regarding the AHA integration into the multiple levels of the encoding Bi-GRU_S and BERT_{BASE} architectures (rows 2 and 4), an improvement in all evaluation datasets is obtained compared to its counterpart (single-level integration) shown in Table 4.4; in addition to this, the best results are obtained with the integration of the AHA architecture into the BERT_{BASE} model (row 2 vs row 4). On the other hand, the results obtained with the GHA architecture (rows 3 and 5) are superior compared to those with the use of the AHA architecture (row 2 vs 3 and row 4 vs 5), which indicates that the architectures benefit from the weighted fusion between the different encoding levels. Overall, the best results of this evaluation are obtained with the GHA architecture, with the use of the BERT_{BASE} model as encoding representation. These findings are consistent according to the research conducted by Chakrabarty, Gupta, and Muresan (2019), in which the benefit of deep stacked architectures is demonstrated.

Finally, when comparing the results of our best approach (GHA architecture with BERT_{BASE} as encoding representation) against the SOTA approaches (row 5 vs 6), better results are obtained in 4 out of 6 evaluation datasets, showing an overall improvement of up to 3.9%. These results show the improvement in the detection of

¹²(Mozafari, Farahbakhsh, and Crespi, 2019b)

¹³(Chakrabarty, Gupta, and Muresan, 2019)

¹⁴(Liu, Li, and Zou, 2019)

¹⁵(Saha et al., 2018)

¹⁶(Wang et al., 2019)

AL with the use of the proposed DA mechanism and MLDA architectures. Regarding the SOTA results: 1) for the shared-task datasets, we select the results of the best team within the competition, 2) for the unique datasets presented at AL research, we selected the best results which report the dataset partition-split, as well as evaluation metrics. To make a fair comparison with the different SOTA methods and their respective configurations, the SOTA results reported in Table 4.6 were taken from their respective papers.

Specifically, for the *Waseem* and *Davidson* datasets, we compare our results with Mozafari, Farahbakhsh, and Crespi (2019b), where the integration of CNNs at the different encoding levels of the BERT model was proposed. The representations obtained at each level were concatenated and passed through a classification layer. For the *Golbeck* dataset, we compare against Chakrabarty, Gupta, and Muresan (2019), which proposed an architecture based on stacking RNNs and the integration of CA at different stack levels. For the *SE 2019 T6* dataset, we compare our results with Liu, Li, and Zou (2019), where BERT fine-tuning was performed using different preprocessing techniques, such as converting emojis to text and hashtag segmentation. For the *AMI 2018* dataset, we compare against Saha et al. (2018), which used logistic regression as a classification algorithm and combined sentence embeddings, TF-IDF, and Bag of Words representations as input feature vectors. Finally, for the *HASOC 2019* dataset, we compare with Wang et al. (2019), which explored using a k-fold ensemble approach based on the Ordered Neurons LSTM architecture (Shen et al., 2019) coupled with a CA mechanism.

As can be seen, there is a wide variety of approaches among the reported SOTA systems, which incorporate different text preprocessing techniques in various representations, ranging from traditional Bag of Words to Transformer models, and apply different classification algorithms from traditional machine learning ones like logistic regression to deep architectures coupled in Transformer models. Unlike this wide range of approaches, our method is based on the integration of our DA mechanism into RNNs and Transformer-based architectures, as well as its expansion to a multi-level perspective. These proposed approaches achieved good results in AL detection, outperforming SOTA systems in 4 out of 6 evaluation datasets using our best configuration. These results allow us to conclude that the proposed approaches are effective in detecting different types of AL.

The results of evaluating the hierarchical integration of the proposed DA mecha-

nism within the VisualBERT encoding architecture and the proposed baseline for the detection of AL in memes are presented in Table 4.7. Additionally, the table includes a comparison with SOTA approaches. The reported results for the SOTA approaches correspond to those of the winning teams in the shared tasks of the three evaluation datasets. All values were directly extracted from the respective overview papers of the corresponding shared tasks (Kiela et al., 2020; Fersini et al., 2022; Jarquín-Vásquez et al., 2024).

AM	EA	HMC	MAMI	DIMEMEX ST1	DIMEMEX ST2
AHA	VisualBERT	0.716±0.0081	0.739±0.0093	0.541±0.0090	0.363±0.0082
GHA	VisualBERT	0.728±0.0075	0.747±0.0085	0.559±0.0072	0.398±0.0109
-	SOTA	0.765 ¹⁷	0.834 ¹⁸	0.583 ¹⁹	0.447 ¹⁹

Table 4.7: Comparison results from our baseline architecture, our proposed GHA architecture, and state-of-the-art approaches in three datasets for the detection of AL in memes.

As shown in the results obtained from the integration of the baseline AHA (row 2), a consistent improvement was achieved across all datasets compared to its counterpart (single-level integration) presented in Table 4.5. These findings highlight the advantages of leveraging multiple encoding levels of Transformer architectures over exclusively using the last encoding level.

When comparing the results obtained with the weighted integration of encoding levels from the GHA architecture against the baseline AHA (row 3 vs. row 2), an improvement is observed across all evaluation datasets. This demonstrates the benefits of employing a weighted fusion of all encoding levels. The most significant improvement was achieved in the dataset for Subtask 2 of the DIMEMEX shared task, which involves multi-class classification of different types of hate speech.

The results presented in Table 4.7 also compare the performance of the reported SOTA approaches against our proposed hierarchical integration of the DA mechanism. When comparing our best results (obtained using the GHA architecture integrated with VisualBERT) against the SOTA approaches (row 3 vs. row 4), it can be observed that our method did not outperform the SOTA systems on any of the evaluation datasets. This is likely due to the fact that the winning systems rely on ensemble techniques combining diverse pre-trained text, vision, and vision-and-language models, while our approach is based solely on the hierarchical integration of the proposed

¹⁷(Zhu, 2020)

¹⁸(Zhang and Wang, 2022)

¹⁹(Wang and Markov, 2024)

DA mechanism within the VisualBERT model.

Specifically, the winning approach for the HMC shared task (Zhu, 2020) applied an inpainting model to detect and remove text from images, improving object detection and web entity recognition. The cleaned images were then processed for bottom-up-attention feature extraction, web entity detection, and human race identification, enriching the input to transformer models. The team trained VL-BERT, UNITER-ITM/VILLA-ITM, and vanilla ERNIE-Vil models on the extracted information and averaged their predictions. For the MAMI shared task, the winning team (Zhang and Wang, 2022) defined an ensemble model that combined deep multimodal features with Multi-Layer Perceptrons, Extreme Gradient Boosting, and Gradient-Boosted Decision Trees. Finally, the DIMEMEX shared task winner (Wang and Markov, 2024) explored the integration of four SOTA language models (XLM-T, Multilingual-E5, RoBERTa-base-BNE, BETO) with the Swin Transformer-based visual model, and employed a Multilayer Perceptron fusion module to create a robust multimodal classification system.

As described, the winning approaches utilize ensemble techniques and a wide array of sophisticated pre-trained text, vision, and vision-and-language models, resulting in significantly higher computational complexity and resource demands. In contrast, our methodology emphasizes the hierarchical integration of the proposed DA mechanism within a single vision & language model. This approach prioritizes simplicity and computational efficiency, offering a more streamlined alternative. A detailed analysis of the trade-off between performance and complexity is presented in Chapter 5.

4.4.3. On the Relevance of the Dual Attention Mechanism

In order to analyze in more detail the performance of the proposed DA mechanism, we focus on the analysis of complementarity, and diversity over the SA and CA mechanisms. This analysis is conducted using the Bi-GRU and VisualBERT encoding architectures for detecting AL in text and memes, respectively. For measuring complementarity, we utilized the MPA metric, while for assessing diversity, we applied the CFD metric (Tang, Suganthan, and Yao, 2006). Both metrics are formally defined in Subsections 2.6.7 and 2.6.8, respectively.

Table 4.8 presents the results of the MPA and CFD evaluation metrics across the six AL text detection datasets and the three AL meme detection datasets. Additio-

nally, the table reports the accuracy obtained using the SA, CA, and DA mechanisms. When comparing the performance of the SA and CA mechanisms (column 2 vs 3), better accuracy results are obtained in all datasets with the CA mechanism, which shows the advantage of incorporating the general context via C_v in the detection of AL. In order to measure the diversity of both attention mechanisms, we apply the CFD metric in the predictions of the SA and CA mechanisms (shown in column 5), the results show that although the diversity is low, there is complementarity in the predictions of both mechanisms, this motivated us in the creation of the DA mechanism, which seeks to combine and complement the strengths of the SA and CA mechanisms. Table 4.8 also reports the accuracy of the proposed DA mechanism in comparison with the SA and CA mechanisms (column 6 vs columns 2 and 3), as presented, the DA mechanism shows better performance results in all evaluation datasets, showing the advantages of combining both mechanisms. Finally, we report the MPA obtained with SA and CA predictions (column 4), as revealed, a performance improvement is obtained in all the evaluation datasets, which indicates the complementarity of both mechanisms and the existence of a margin of improvement for the future development of novel DA mechanisms.

Dataset	SA	CA	MPA	CFD	DA
Waseem	0.8359	0.8471	0.8821	0.1825	0.8624
Davidson	0.9381	0.9476	0.9679	0.2057	0.9523
Golbeck	0.7379	0.7521	0.7893	0.2079	0.7726
SE 2019 T 6	0.8396	0.8419	0.9085	0.1827	0.8772
AMI 2018	0.6102	0.6317	0.6792	0.2138	0.6478
HASOC 2019	0.7498	0.7604	0.8062	0.2038	0.7830
HMC	0.6723	0.6842	0.7219	0.1794	0.7016
MAMI	0.7495	0.7818	0.8137	0.1952	0.8014
DIMEMEX ST1	0.7248	0.7426	0.7891	0.2018	0.7730
DIMEMEX ST2	0.5892	0.6134	0.6783	0.2192	0.6473

Table 4.8: Comparison results of the complementarity and diversity of the SA and CA mechanisms contrasted with the performance of the DA mechanism. For evaluation, all AMs were integrated with a Bi-GRU encoding architecture for the analysis of text datasets, while the VisualBERT model was used for the analysis of meme datasets. The SA, CA, and DA result columns report accuracy.

4.4.4. Statistical Significance Analysis

We used a Bayesian Wilcoxon signed-rank test to assess the importance of integrating the DA mechanism in the encoding architectures, as well as, the AHA and GHA

architectures. It is advised to directly compare machine learning classifiers using this test, which is a non-parametric Bayesian variant of the Wilcoxon signed-rank test built on the Dirichlet process (Benavoli et al., 2017). The test calculates the posterior probabilities of the null and alternative hypotheses given the observed data, giving a clear probability of one approach being superior to the other (when comparing two treatments).

For this analysis, we define methods A and B, according to the integration of the DA mechanism in the encoding architectures, as well as the AHA and GHA architectures. Table 4.9 displays the findings of this statistical analysis over the macro-F1 scores, where the notation “>” denotes “*better than*”. We can observe that there is a very high probability (> 0.9794) that the integration of the DA mechanism over the Bi-GRU, Bi-GRU_S, and BERT offers better results than the sole use of the encoding architectures (rows 2, 3, and 4). On the other hand, when comparing the hierarchical integration of the DA mechanism vs. its integration into the last encoding layer in text encoding architectures (rows 5 and 6), there is a high probability (> 0.7224) that its hierarchical integration obtains better results, by leveraging all the encoding levels of the encoding architectures.

Encoding Architecture	A	B	p(A > B)	p(rope)	p(B > A)
Bi-GRU	DA	-	0.9932	0.0059	0.0008
Bi-GRU _S	DA	-	0.9933	0.0056	0.0009
BERT	DA	-	0.9794	0.0102	0.0102
Bi-GRU _S	AHA	DA	0.7224	0.2740	0.0034
BERT	AHA	DA	0.8850	0.1123	0.0026
Bi-GRU _S	GHA	AHA	0.6040	0.3669	0.0289
BERT	GHA	AHA	0.7225	0.2741	0.0032
VisualBERT	DA	-	0.8261	0.1594	0.0145
VisualBERT	GHA	AHA	0.7403	0.2368	0.0229

Table 4.9: Bayesian signed-rank test results for each proposed approach. The A and B columns indicate the integration of the proposed DA mechanism, as well as the proposed approaches of their multilevel integration, over the encoding architectures; the ‘-’ symbol over the B column, denotes the absence of the DA mechanism.

For the weighted integration of all encoding layers in text encoding architectures, there is a probability (> 0.6040) of achieving better results compared to the unweighted integration (rows 7 and 8). Regarding the integration of the proposed DA mechanism into the VisualBERT model (row 9), there is a high probability (> 0.8261) of obtaining better results compared to its counterpart that relies solely on fine-tuning. Finally, for the weighted integration of all encoding layers in the VisualBERT model vs. its hierarchical integration in the AHA architecture (row 10), there is a high probability (> 0.7403) of achieving better results with the weighted integration

of all encoding layers in the proposed DA mechanism.

To help visualize this analysis, in Figure 4.9 we map 150,000 Monte Carlo samples in barycentric coordinates as proposed by Benavoli et al. (2014), where each vertex of the triangle is associated with each Bayesian test scenario. For example, using the data provided in Table 4.9, the Bayesian Test concluded that for 148,980 out of 150,000 samples, the integration of the DA mechanism in the Bi-GRU encoding architecture is advantageous to exclusively using the Bi-GRU encoding output.

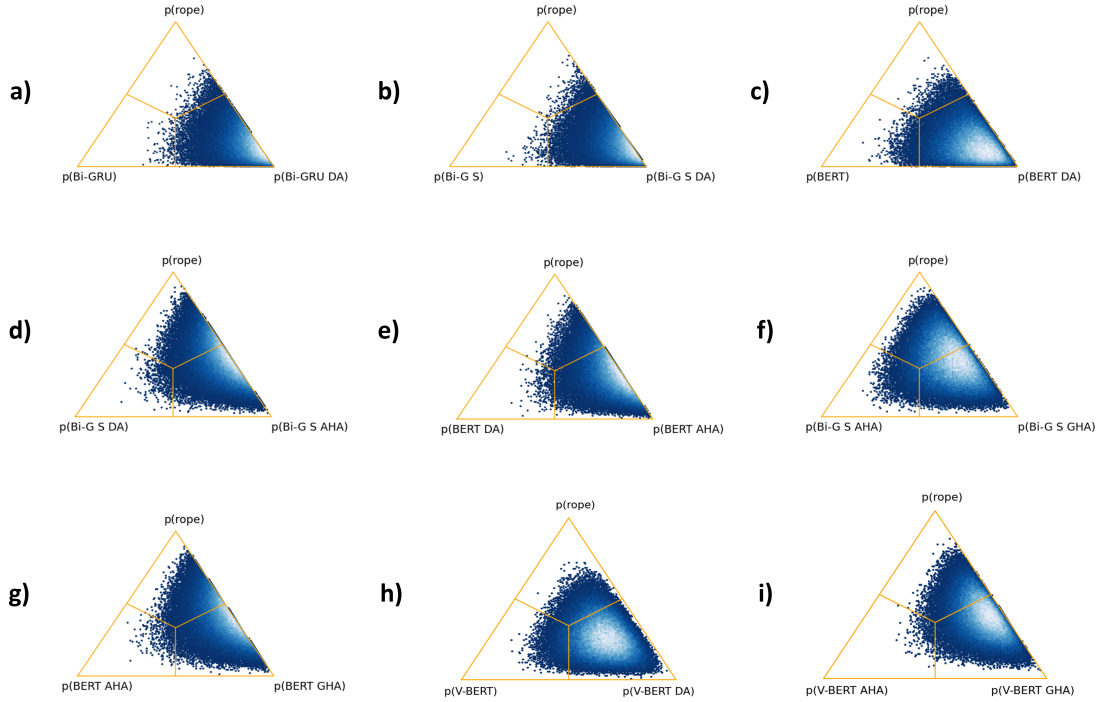


Figure 4.9: Visualization of the Bayesian Test for comparing: 1) the integration of the DA mechanism in the Bi-GRU, Bi-GRU_S, and BERT architectures (subsections a, b, and c); 2) the AHA architectures vs. the sole use of the DA mechanism over the last encoding layer of the Bi-GRU_S and BERT architectures (subsections d and e); 3) the GHA architecture vs. the AHA architecture using the Bi-GRU_S and BERT architectures (subsections f and g); and 4) the integration of the DA mechanism into the VisualBERT model vs. the fine-tuned one, and the GHA architecture vs. the AHA architecture integrated into the VisualBERT model (subsections h and i).

4.5. Analysis of Results

This section provides a qualitative analysis of the results obtained through the integration of the proposed DA mechanism, as well as the proposed GHA architecture. The analysis presented in this section focuses on the detection of AL in text. A detailed qualitative analysis of the results of AL detection in memes is provided in the following chapter. This section is divided into two subsections. The first subsection examines the sigma values across different encoding levels of the BERT model using the GHA architecture. The second subsection provides an in-depth analysis of the attention values generated by the proposed DA mechanism, contrasting its performance against the SA mechanism.

NOTE: This section contains examples of language that may be offensive to some readers, these do not represent the perspectives of the authors.

4.5.1. Relevance Analysis of the BERT Encoding Layers Using the GHA Architecture

In general, the best results were obtained with the GHA architecture, using the BERT_{BASE} model as encoding representation. In order to analyze in detail the outstanding performance of this configuration, in this subsection we propose to analyze the leverage of the different encoding levels obtained through the GHA architecture. This analysis will be performed in the detection of AL at three different levels, namely: dataset, type of AL, and instance samples. In recent studies, a wide variety of research has addressed and analyzed the contribution of the encoding layers of the Tranformer-based architectures for a variety of tasks (Clark et al., 2019; van Aken et al., 2019), these approaches have focused on the analysis of the multiple self-attention heads at the different encoding layers. Unlike these approaches, we plan to analyze them from a different perspective, using the activations captured through the GMU units of the GHA architecture, specifically, we focus on the analysis of the $z_{j,i}$ values obtained with the sigma function (as presented in Equation 4.2.3) which represent the corresponding relevance of the i -th token at the j -th encoding layer.

Figures 4.10 and 4.11 present the activation heat map matrices of the sigma value analysis, with respect to the three aforementioned analysis levels; in all cases, we reported the average score of the $z_{j,i}$ values obtained at each of the 12 encoding

layers of the $BERT_{BASE}$ model. For the sake of interpretation, each row in the heatmap matrix represents the average activation of an encoding layer, the 12 encoding layers are presented from bottom to top, where the stronger color represents a greater activation.

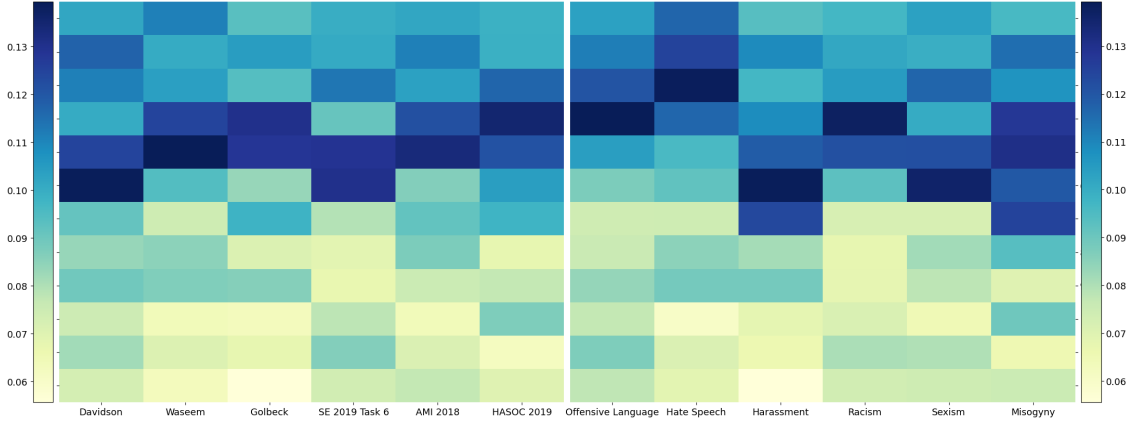


Figure 4.10: Visualization of the relevance of the encoding layers using the GHA architecture; the left-hand side heatmap presents the relevance by dataset, on the other hand, the right-hand heatmap illustrates the relevance by type of abusive language.

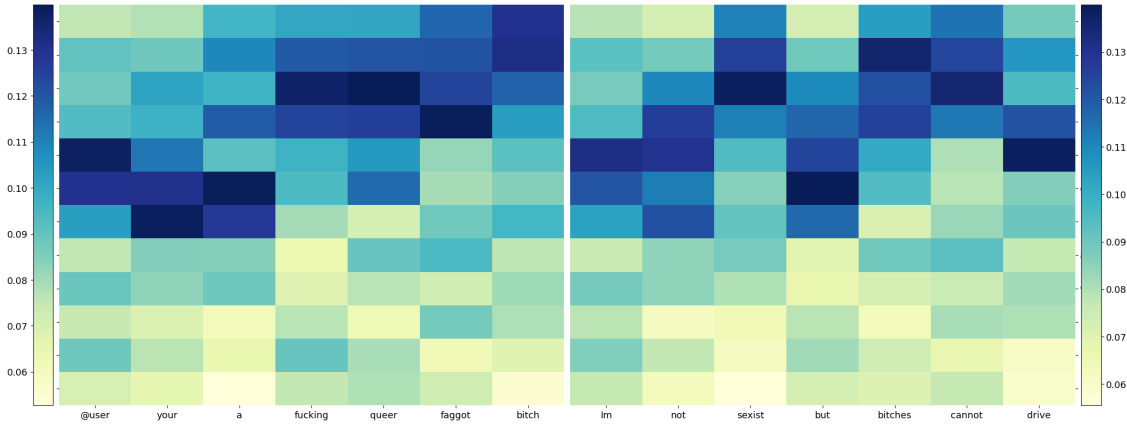


Figure 4.11: Visualization of the relevance of the encoding layers by instance samples; the samples were taken from the Davidson and Waseem dataset, which corresponds to an offensive and sexist sample, respectively. The left-hand heatmap presents the offensive instance: “@user your a fucking queer faggot bitch”, while the right-hand heatmap presents the sexist instance: “Im not sexist but bitches cannot drive”.

Figures 4.10 and 4.11 present the results of the analysis at the three proposed levels. Regarding the sigma analysis of the datasets (left heatmap of Figure 4.10), low

average activation values were captured in the first layers of the Transformer model, on the other hand, the highest average activation values were obtained in the middle-upper layers (specifically in layers 7, 8 and 9), which indicates according to Clark et al. (2019) that the Transformer model focuses on more contextual aspects of the language in the detection of AL. Concerning the analysis of the type of AL (right heatmap of Figure 4.10) similar results were obtained, where low average activation values were captured in the first layers of the Transformer model, on the other hand, the activation of the upper layers in the detection of offensive language and hate speech stands out, in contrast to the detection of harassment, sexism, racism, and misogyny (layers 9-11 vs. 6-8). This shows evidence that the Transformer model focuses on more semantic and contextual aspects of the language in the detection of offensive language and hate speech, in contrast to the detection of harassment, sexism, racism, and misogyny, where it focuses on more syntactic aspects of language. Finally, Figure 4.11 shows the sigma analysis in two AL samples, in which a low average activation stands out in the upper layers, in personal pronouns, prepositions, and non-offensive words (e.g. “not”, “a”, “your”, “@user”, and “but”), in contrast to potentially offensive words (e.g. “fucking”, “faggot”, and “bitch”), which presented high activation values in the upper layers, which evidence that their interpretation depends more on contextual and semantic aspects for the detection of AL.

4.5.2. Analysis of Attention Values in the Proposed Dual Attention Mechanism

To measure the effectiveness of integrating the relevance of local and contextual features through the proposed DA mechanism, in this subsection, we perform an analysis of the attention values. This analysis consists of 1) extracting the *top-k* most relevant words/expressions via the DA mechanisms in the AL detection task, and 2) comparing these obtained words/expressions with the attention values of the DA mechanism and the BERT attention values against a well-acknowledged lexicon. Specifically, we used the *Hatebase*¹² database, which is a specialized lexicon of potentially offensive words and expressions. This comparison aims to evaluate the most relevant words captured by the SA and DA mechanisms and validate the effectiveness of combining the SA and CA mechanisms.

¹²<https://hatebase.org/>

To extract these relevant words/expressions we proposed the *Local Attention Score (LAS)*. This score is designed to analyze the patterns captured by the DA mechanism through attention values in the different evaluation datasets. LAS is inspired by the combination of the Local Mutual Information (LMI) and the attention values. By weighting both values, we take into account the number of word occurrences in the abusive class and the relevance captured through the attention values. For interpretation purposes, obtaining high values with the LAS indicates greater relevance of the word/expression in terms of its number of occurrences in the abusive class and its relevance captured by the attention mechanism for the detection of AL. The LAS is shown in Equation 4.5.1, where $LAS(w, c)$ represent the relevance score of word w in class c (the abusive class), $p(c|w)$ and $p(c)$ are calculated by $\frac{count(w,c)}{count(w)}$ and $\frac{count(c)}{|D|}$. Furthermore, $p(w|c)$ and $\bar{\alpha}(w, c)$ are calculated by $\frac{count(w,c)}{|D|}$ and $\frac{1}{n} \sum_{i=1}^n (w_i, c)$, where n is the number of occurrences of word w in class c , and (w_i, c) is the attention value of the i -th occurrence of word w in class c ; $|D|$ is the number of occurrences of all words in the training set. Table 4.10 presents the results of the *top-20* words/expressions obtained with the proposed LAS metric.

top-word	Waseem	Davidson	Golbeck	SE 2019 T 6	AMI 2018	HASOC 2019
1	sexist	bitch	cunt	shit	bitch	fuck
2	islam	bitches	nigger	fuck	whore	fucking
3	muslims	hoes	cunts	fucking	bitches	shit
4	cunt	pussy	fuck	ass	cunt	ass
5	mohammed	hoe	niggers	bitch	girls	liar
6	bimbos	fuck	muslim	stupid	pussy	idiot
7	prophet	nigga	kill	disgusting	slut	traitor
8	quran	shit	muslims	idiot	women	asshole
9	women	ass	ass	sucks	ladies	bitch
10	bitch	faggot	bitch	liar	girl	bastards
11	religion	fucking	white	crap	vagina	shame
12	drive	niggas	juice	bullshit	whores	damn
13	feminist	cunt	pussy	fucked	fucking	fool
14	feminists	niggah	whites	nigga	ass	racist
15	bitches	fag	cock	ignorant	fuck	idiots
16	islamic	nigger	ugly	dumb	female	stupid
17	jews	fuckin	nigga	racist	woman	moron
18	men	faggots	fucking	disgrace	suck	fck
19	rape	retarded	burn	asshole	skank	bullshit
20	blondes	niccas	stupid	hypocrites	sluts	suck

Table 4.10: Top-20 words obtained with the proposed *Local Attention Score (Equation 4.5.1)* over the abusive class, with the use of the proposed DA mechanism. The words indicated in bold, represent the words contained in the *Hatebase* lexicon.

$$LAS(w, c) = p(w, c).log(\frac{p(c|w)}{p(c)}).\bar{\alpha}(w, c) \quad (4.5.1)$$

As shown in Table 4.10, the top-20 words/expressions captured with the

LAS using the DA values, correspond to potentially offensive words/expressions, stereotype-based words, or specific hate target groups. The proposed score was applied to all evaluation datasets, these results are displayed in the different columns of Table 4.10. Regarding the captured results, a clear trend can be observed in the use of specific words/expressions according to the type of AL addressed in each dataset, for example, the top most relevant words/expressions for the Waseem dataset (dedicated to the detection of sexism and racism) are highly related to offenses against women or stereotypes, or to hate target groups with regard to racism detection. Concerning the datasets dedicated to the detection of offensive language (such as the case of task 6 of SemEval 2019) a greater number of vulgar words/expressions were captured. On the other hand, in the datasets dedicated to the detection of harassment and misogyny (as is the case of the Golbeck and AMI 2018 datasets), the capture of vulgar and pejorative words/expressions referring to women is shown. Finally, some words/expressions that are not potentially offensive were also captured by the proposed *LAS*, such as “women”, “drive”, and “blondes”. To analyze them better, Table 4.11 presents examples of some offensive instances that use some of these words, which are used as a stereotype or as the target of an offense. For example, in the offensive instance “*I’m not sexist but BITCHES CANNOT DRIVE*”, the word *drive* is used as a stereotype, where its contextual interpretation is important to classify the instance as misogynistic. The capture of these words by *LAS* is highly related to the type of AL addressed in the evaluation datasets, as well as any biases they may contain. The words indicated in bold in Table 4.10 are contained in the Hatebase lexicon as potentially offensive words/expressions.

#	Word	Text
1	Drive	RT NathanWassihun I’m not sexist but BITCHES CANNOT DRIVE
2	Drive	HOLY FUCK IM NOT SEXIST BUT ALOT OF WOMEN CANNOT FUCKING DRIVE
3	Blondes	Dumb blondes with pretty faces? You’re definitely right on one of those statements...
4	Feminism	BoycottBrandy I just wanted proof that feminism sheep believe the lie.
5	Hypocrite	@USER GOP, Conservatives, Evangelicals, Traditionalists Catholics are all hypocrites.
6	Ignorant	@USER @USER She is a Sick Corrupt Ignorant Moron!

Table 4.11: Examples of non-offensive words captured with the local attention score, which are used in offensive contexts.

Regarding the second analysis between the *top – k* most relevant words captured via the proposed *LAS* using the BERT attention values and the DA values in contrast with the well-acknowledge *Hatebase* lexicon, taking as our ground-truth; Table 4.12 presents the results of this comparison, between the intersections obtained with the use of the attention values of the BERT model (reported at column 2) and with the use

of the attention values obtained with the proposed DA mechanism coupled with the BERT model (shown in column 3). As reported in Table 4.12, the use of the proposed DA mechanism in the BERT model allows capturing a greater number of potentially offensive words/expressions, compared to the sole use of the SA mechanism in the BERT model, in some cases almost doubling the results obtained without the use of the proposed DA mechanism. This supports the quantitative improvement in the detection of AL with the incorporation of contextual information (CA mechanism) and the relationships between the elements of the sequence (SA mechanism) via the proposed DA mechanism.

Dataset	(-)BERT	(DA)BERT	Possible Words
Waseem	6 (42 %)	11 (78 %)	14
Davidson	12 (46 %)	19 (73 %)	26
Golbeck	9 (37 %)	18 (75 %)	24
SE 2019 T 6	13 (44 %)	21 (72 %)	29
AMI 2018	8 (34 %)	17 (73 %)	23
HASOC 2019	10 (47 %)	16 (76 %)	21

Table 4.12: The intersection percentage of the top-50 words with the highest Local Attention scores and the *Hatebase* lexicon database, (-) indicates the absence of the proposed DA mechanism.

Proposed Cross-Modal Dual Attention

This chapter introduces the Cross-Modal Dual Attention (CMDA) mechanism, designed to adapt the proposed DA mechanism for multimodal classification approaches. Its primary objective is to measure the relevance and interaction between pairs of elements from different modalities. Specifically, the main goal of this mechanism is to bridge the gap between the results obtained in AL detection in memes and state-of-the-art approaches. Additionally, the CMDA mechanism has been adapted to a bi-contextual architecture, extending its applicability to more than two modalities. The evaluation of this proposed mechanism focuses exclusively on AL detection in memes.

The chapter is divided into four sections. Section 1 introduces the CMDA mechanism along with the bi-contextual architecture. Section 2 provides implementation details and outlines the baselines used to compare the performance of the proposed approaches. Section 3 presents the quantitative results obtained from evaluating the proposed approaches against state-of-the-art methods. This section also includes a comparison of the number of parameters in the models being evaluated. Finally, Section 4 offers a qualitative analysis of the CMDA mechanism, focusing on attention value visualizations and error analysis.

5.1. Cross-Modal Dual Attention Mechanism

This section is organized into the following subsections: the first subsection introduces the proposed CMDA mechanism, which incorporates the DA mechanism

into a multimodal perspective with the aim of improving the alignment between elements from two different modalities. The second subsection presents a bi-contextual approach designed to integrate information from more than two modalities.

5.1.1. Construction of the CMDA Mechanism

To further enhance the initially proposed DA mechanism, we aimed to expand its capabilities into a cross-modal approach. In this subsection, we introduce the CMDA mechanism, specifically developed to improve AL detection in memes. CMDA achieves this by integrating information from two distinct modalities in a cross-modal framework. For clarity, these modalities are represented by the text and image components of memes.

The inspiration for CMDA stems from the cross-modal attention mechanism proposed by Ye et al. (2019). The core intuition of the cross-modal attention mechanism involves the interaction between a pair of modalities in the form of sequences by adapting the self-attention mechanism. The cross-modal attention block takes two input modalities, α and β , along with their respective sequences, $X_\alpha \in \mathbb{R}^{T_\alpha \times d_\alpha}$ and $X_\beta \in \mathbb{R}^{T_\beta \times d_\beta}$. The cross-modal attention mechanism aims to adapt the modality β to α by incorporating the contextual information from one modality into the other.

Figure 5.1 illustrates the CMDA mechanism, which processes sequential data from two distinct modalities. Its primary contribution lies in integrating contextual information from a given modality to enhance the latent adaptation of modality β to modality α , thereby achieving improved alignment of relevant features during the latent adaptation process.

Parallel to classical (single modality) self-attention, the CMDA mechanism maps the first modality, α , into a set of queries denoted by $Q_\alpha = X_\alpha W_{Q_\alpha}$. While the set of key-value pairs is obtained from the second modality, given by $K_\beta = X_\beta W_{K_\beta}$ and $V_\beta = X_\beta W_{V_\beta}$. Where $W_{Q_\alpha} \in \mathbb{R}^{d_\alpha \times d_k}$, $W_{K_\beta} \in \mathbb{R}^{d_\beta \times d_k}$, and $W_{V_\beta} \in \mathbb{R}^{d_\beta \times d_v}$. The latent adaptation from β to α in the CMDA mechanism is defined as:

$$CMDA_{\beta \rightarrow \alpha} \in \mathbb{R}^{T_\alpha \times d_v} = \left[\text{softmax} \left(\frac{Q_\alpha K_\beta^T}{\sqrt{d_k}} \right) \odot c_\beta \right] V_\beta \quad (5.1.1)$$

Specifically, in Equation 5.1.1, the portion on the right-hand side enclosed within the brackets calculates the general attention filter of the CMDA mechanism. This is achieved through the element-wise multiplication of the contextualized vector c_β

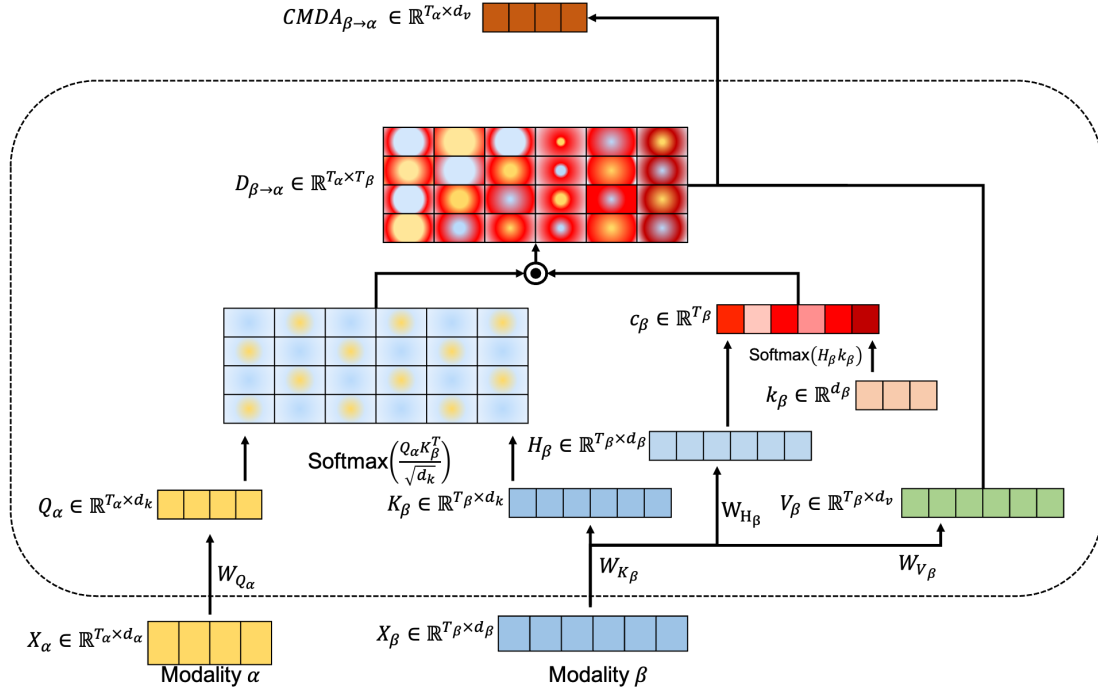


Figure 5.1: Proposed Cross-Modal Dual Attention Mechanism.

and the matrix obtained by calculating the similarity between the features of both modalities using a normalized dot product with the softmax function. The purpose of this process is to combine the relevance between each pair of features from the two modalities while emphasizing the importance of each feature from modality β in solving the task at hand (AL detection in memes). This generated representation serves as an initial step in crafting the dual attention kernel, denoted as $D_{\beta \rightarrow \alpha}$. The purpose of this kernel is to integrate contextual information from modality β into the latent adaptation of modality α .

$$H_\beta = \tanh(X_\beta W_{H_\beta}) \quad (5.1.2)$$

$$c_\beta = \text{softmax}(H_\beta k_\beta) \quad (5.1.3)$$

The contextual information specific to modality β is stored within the attention vector $c_\beta \in \mathbb{R}^{T_\beta}$, which is computed through a normalized dot product (as shown in Equation 5.1.3) between the matrix of hidden states H_β (obtained according to Equation 5.1.2) and the context vector k_β . It's worth noting that the context vector k_β is

initialized randomly and refined through joint learning during the training process. Ultimately, the dual attention kernel $D_{\beta \rightarrow \alpha}$ is employed to adjust the final representation of modality α , based on the similarity of their respective features. The ultimate representation $CMDA_{\beta \rightarrow \alpha}$ is derived by multiplying the matrices $D_{\beta \rightarrow \alpha}$ and V_{β} .

5.1.2. Proposed Bi-contextual CMDA Architecture

To extend the applicability of the CMDA mechanism to more than two modalities, this subsection introduces the Bi-contextual CMDA architecture. Since the CMDA mechanism enables the latent adaptation of one modality to another, it allows for various combinations of inputs and outputs within the proposed CMDA mechanism. The Bi-contextual architecture, depicted in Figure 5.2, leverages two CMDA mechanisms for AL detection in memes.

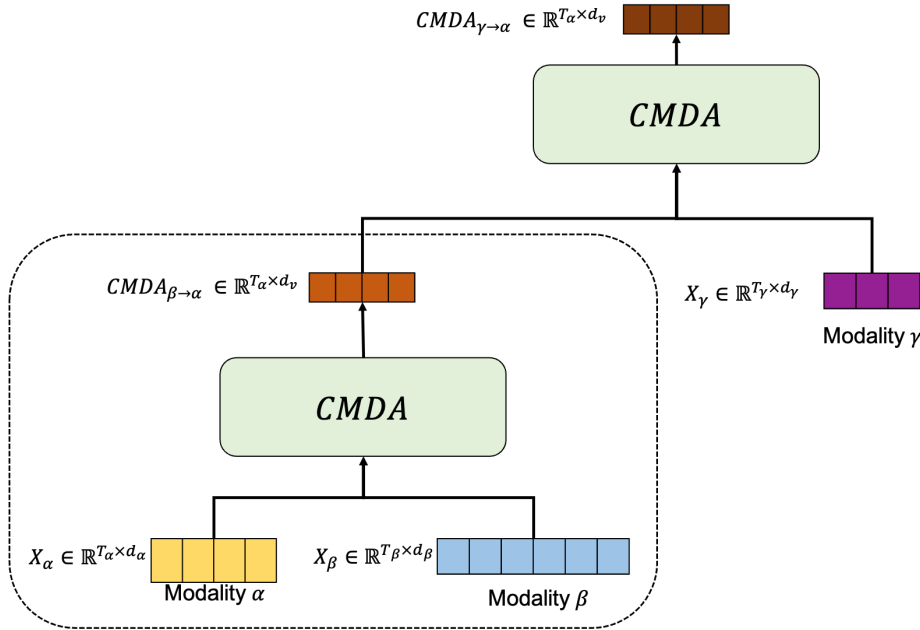


Figure 5.2: Proposed Bi-contextual CMDA architecture.

Specifically, the dashed rectangle in the lower portion of the diagram represents the foundational architecture, which processes two input modalities, denoted as α and β . The resulting representation, $CMDA_{\beta \rightarrow \alpha} \in \mathbb{R}^{T_{\alpha} \times d_v}$, is subsequently treated as a combined input modality for a second CMDA mechanism. This second mechanism integrates the combined representation with a third modality, γ , yielding a new representation, $CMDA_{\gamma \rightarrow \alpha} \in \mathbb{R}^{T_{\alpha} \times d_v}$. This final representation effectively combines the

contextual information from two modalities, γ and β , to enhance the representation of α .

The use of the Bi-contextual CMDA architecture enables flexibility in the ordering of the modalities assigned to α , β , and γ . In the previous chapter, the modalities employed for AL detection in memes included the text extracted from the image and the image of the meme itself. Additionally, to fully exploit the potential of this architecture and to provide the models with richer contextual information, a new modality was incorporated: the description of the meme image. To integrate this modality, we employed an image captioning task.

Figure 5.3 showcases an example generated through the image captioning task, where the descriptions were created using the BLIP¹ model. These three modalities were utilized in our experiments.

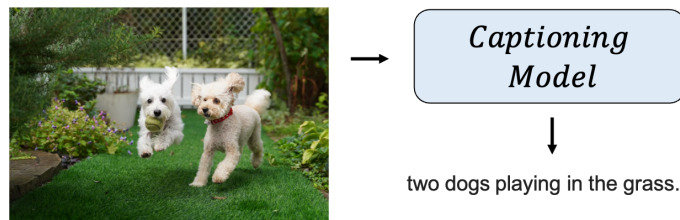


Figure 5.3: Image captioning example.

To perform the classification process using the CMDA mechanism and the Bi-contextual architecture, the output from these architectures is passed through an Average Pooling layer to obtain the classification vector. Finally, the resulting representation is fed into the classification layers.

5.2. Proposed Baselines and Implementation Details

This section provides the implementation details and outlines the baselines used to evaluate the effectiveness of the proposed CMDA mechanism. The evaluation was conducted using the same datasets for AL detection in memes described in the previous chapter. Additionally, the same evaluation metrics were employed to assess the performance of the CMDA mechanism. This section is structured as follows: first,

¹https://huggingface.co/docs/transformers/model_doc/blip

the baselines proposed to measure the effectiveness of the CMDA mechanism and the Bi-contextual architecture are presented. Subsequently, the implementation details are discussed, including the hyperparameters of the various proposed approaches.

5.2.1. Proposed Baselines for the Evaluation of the CMDA Mechanism

To evaluate the effectiveness of the CMDA mechanism, an initial evaluation stage was conducted to measure the impact of the modalities used. The following baselines were proposed:

- 1.- Fine-tuning the BERT_{BASE}/BETO model for AL detection in memes, using only the text modality extracted from the memes. The BETO model is used for the DIMEMEX dataset, while the BERT model is employed for the HMC and MAMI datasets.
- 2.- Fine-tuning the BERT_{BASE}/BETO model, using only the text modality derived from the image captions extracted with the BLIP model.
- 3.- Fine-tuning the Vision Transformer (ViT) model, using only the image modality.

To enable a direct comparison of the latent adaptation of different modality pairs using the CMDA mechanism, the CMA mechanism described in Section 2.4.1 was implemented. Additionally, the CMA mechanism was integrated into the Bi-contextual architecture to assess the impact of the proposed CMDA mechanism within the Bi-contextual architecture. For consistency, all these configurations utilized the same parameter settings, pre-trained models, and classification schemes.

To evaluate the effectiveness of the proposed CMDA mechanism and Bi-contextual architecture against a robust baseline previously demonstrated to be effective in AL detection in memes, we decided to replicate the Hate-CLIPper model proposed by Kumar and Nandakumar (2022). This model leverages the pre-trained CLIP² model to extract image encodings relative to the text and text encodings relative to the image. Both representations are passed through the CMA mechanism to evaluate the interactions between paired elements from both modalities. Subsequently, the diagonal of the resulting representation is concatenated to form a classification vector.

²https://huggingface.co/docs/transformers/model_doc/clip

This vector is then fed into a fully connected layer, followed by the application of the softmax function to produce the final classification.

Finally, to compare the effectiveness of the proposed approaches against modern methods based on Large Language Models (LLMs) and Prompt Engineering, we proposed two baselines following zero-shot and few-shot classification schemes.

In the zero-shot classification scheme, the LLM was provided with a prompt³ that included the meme to be classified, instructions to categorize it according to the dataset’s predefined classes, the corresponding category definitions, the text extracted from the meme, and the image caption generated using the BLIP model.

In the few-shot classification scheme, in addition to the category definitions, the LLM was given three examples per category, each extracted from its respective training set. Each example included the assigned label for a given meme, an explanation justifying the label, the text contained within the meme, and the image caption generated with the BLIP model. *Due to the model’s data processing limitations, it was not possible to load the images for each example in the few-shot configuration, only the image of the meme to be classified.*

Table 5.1 presents an example of the prompt used in the zero-shot configuration with the MAMI dataset. As illustrated, the prompt contains the image to be classified, followed by the classification instructions, category definitions, the text extracted from the meme, the image caption associated with the meme, and the final instruction “The image belongs to the category:”, which signals the part the model is expected to complete. It is important to note that in our few-shot configuration, the only difference compared to the prompt shown in Table 5.1 is that, between the category definitions and the text contained in the image section, we insert the few-shot examples from the training set under the aforementioned input constraints.

It is important to highlight that, as these schemes do not involve a training phase, only the test examples were presented to the LLMs for evaluation in both cases. The prompts were carefully designed to ensure that the language model generated only the predefined labels corresponding to the dataset’s categories. For a more comprehensive explanation of these classification schemes, we refer the reader to the following review (Sahoo et al., 2024).

³A structured and specifically designed input to interact with a language model, guiding and optimizing the generation of desired responses.

Image
Prompt
<p>You are a helpful assistant.</p> <p>Classify the following meme image into one of the two categories: {misogynistic, non-misogynistic}.</p> <p>Category definitions:</p> <p>Misogynistic: The image conveys, promotes, or reinforces negative stereotypes, hatred, devaluation, objectification, or discrimination against women.</p> <p>Non-misogynistic: The image does not express or promote any harmful or derogatory views toward women. It may be neutral, humorous, or unrelated to gender issues.</p> <p>Text contained in the image: "Valentine's Day's coming? Oh crap! I forgot to get a girlfriend again."</p> <p>Image caption associated with the image: A cartoon man with orange hair and a red jacket looks surprised and says, "Valentine's Day's coming? Oh crap! I forgot to get a girlfriend again," while standing next to a woman with purple hair and a white tank top who looks unimpressed.</p> <p>The image belongs to the category:</p>

Table 5.1: Example of the zero-shot classification scheme used to categorize a non-offensive instance. The image was taken from the MAMI dataset (Fersini et al., 2022).

The LLM used for these experiments was Gemini 1.5⁴, as this model is multimodal, enabling the processing of images alongside the textual information provided in the prompt.

5.2.2. Implementation Details

This subsection describes the implementation details of the proposed approaches. Regarding the preprocessing of text and images, we utilized the same steps described in Subsection 4.3.4. The same text preprocessing was applied to both the meme text and the text extracted from the image captions. For the models used in the evaluation of the CMDA mechanism, we selected BERT as the encoding architecture for the text in the English datasets, BETO as the encoding architecture for the text in the DIMEMEX dataset, and ViT as the encoding architecture for the images. Regarding

⁴<https://deepmind.google/technologies/gemini/>

the sequence lengths, a maximum length of 40 tokens was used for the text extracted from memes, while a maximum length of 70 tokens was applied to the text extracted from the image captions.

Table 5.2 presents the hyperparameters of the proposed CMDA mechanism, assuming a latent adaptation of $\beta \rightarrow \alpha$, where β represents the image modality and α represents the text modality extracted from the meme text. It is important to note that these representations can be replaced by others or even represent a combination previously obtained by another CMDA mechanism, as illustrated in the proposed bi-contextual architecture.

Vectors, Matrices and Variables		Size
	$n(\alpha)$	40
	$n(\beta)$	197
	d	768
	k_β	768
	Q_α	40x768
	K_β, V_β	197x768
Layer	Input size	Output size
Embedding (BERT/BETO)	40	40x768
BERT/BETO	40x768	40x768
ViT	197x768	197x768
$CMDA_{\beta \rightarrow \alpha}$	197x768(β), 40x768(α)	40x768
Avg Pooling	40x768	40
Dense ₁	40	#Classes

Table 5.2: Hyperparameters of the proposed CMDA mechanisms, assuming a latent adaptation of $\beta \rightarrow \alpha$, where β represents the image modality and α represents the text modality. The description of the hyperparameters includes details of the classification layers.

Regarding the experimental configurations used for the unimodal baselines, including the fine-tuning of the BERT, BETO, and ViT models, we employed the same experimental setup described in Subsection 4.3.4. For the baselines using the CMA mechanism, these architectures were trained for a total of 3 epochs, with a learning rate of $1e-5$, using the Adam optimizer, a Dropout rate of 10%, and a batch size of 32. The Hate-CLIPper model was trained for a total of 2 epochs, with a learning rate of $5e-5$, using the Adam optimizer, a Dropout rate of 10%, and a batch size of 24. All variants of the proposed CMDA mechanism were trained for a total of 2 epochs, with a learning rate of $1e-5$, using the Adam optimizer (Kingma and Ba, 2015), a Dropout rate of 15%, and a batch size of 24.

All the hyperparameters described above were determined through a grid search using the different sets of values outlined in Subsection 4.3.4. In contrast to the experiments conducted in the previous chapter, all baseline models, as well as the proposed configurations of the CMDA and the Bi-contextual CMDA architecture for the detection of AL in memes, were evaluated on a computer equipped with an NVIDIA RTX 4090 graphics card, 64 GB of DDR5 RAM, and an Intel Core i9-14900K processor.

5.3. Quantitative Results

This section presents the quantitative evaluation results of the different baselines and proposed approaches. This subsection is divided into two parts. The first subsection reports the evaluation results of the CMDA mechanism for AL detection in memes and compares its performance against the proposed baselines and SOTA models. This subsection concludes with a comparison of the number of parameters in the proposed approaches, the considered baselines, and the SOTA models. The second subsection provides a statistical analysis of the obtained results.

5.3.1. Effectiveness of the Proposed Cross-Modal Dual Attention Mechanism

Table 5.3 reports the evaluation results of the proposed baselines and the variants of the proposed CMDA mechanism. As observed, when comparing the performance of the unimodal baselines (rows 2, 3, and 4), the text modality extracted from memes yields better results compared to the exclusive use of images or image captions. When comparing the performance of the pre-trained CLIP model against the unimodal approaches (row 5 vs. rows 2, 3, and 4), a significant improvement is observed with the use of a vision & language pre-trained model, outperforming its counterparts by a wide margin across all evaluation datasets.

Regarding the evaluation of the Hate-CLIPper model, a consistent improvement is observed across all datasets compared to the exclusive use of the CLIP model (row 6 vs. row 5). Additionally, the table reports the evaluation results of the LLM Gemini 1.5 under zero-shot and few-shot classification schemes (rows 7 and 8). As shown, the few-shot setting yields the best performance, demonstrating consistent improvements

across all test datasets. However, it is important to note that the few-shot approach did not outperform the Hate-CLIPper model across the four evaluation datasets. Nevertheless, it achieved better performance compared to the best unimodal results (row 8 vs. row 4).

AM	T. Approach	HMC	MAMI	DIMEMEX ST1	DIMEMEX ST2	#P
-	ViT (I)	0.564±0.0079	0.616±0.0086	0.431±0.0094	0.286±0.0109	86M
-	BERT/BETO (C)	0.571±0.0073	0.608±0.0094	0.434±0.0079	0.249±0.0104	110M
-	BERT/BETO (T)	0.601±0.0084	0.623±0.0091	0.467±0.0063	0.272±0.0081	110M
-	CLIP	0.739±0.0085	0.768±0.0068	0.528±0.0071	0.394±0.0090	151M
-	Hate-CLIPper	0.752±0.0095	0.781±0.0076	0.543±0.0086	0.410±0.0094	151M
-	Gemini 1.5 (ZS)	0.696	0.754	0.486	0.340	≈200B
-	Gemini 1.5 (FS)	0.713	0.771	0.523	0.364	≈200B
CMA	AVG ($T \rightarrow I$)	0.672±0.0071	0.705±0.0060	0.493±0.0091	0.316±0.0113	196.2M
CMA	AVG ($I \rightarrow T$)	0.703±0.0070	0.729±0.0084	0.516±0.0093	0.325±0.0088	196.2M
CMDA	AVG ($T \rightarrow I$)	0.687±0.0079	0.720±0.0086	0.514±0.0092	0.381±0.0118	196.2M
CMDA	AVG ($I \rightarrow T$)	0.725±0.0082	0.752±0.0090	0.547±0.0074	0.392±0.0097	196.2M
CMA	BiC ($I \rightarrow (C \rightarrow T)$)	0.730±0.0067	0.752±0.0084	0.558±0.0095	0.383±0.0082	196.4M
CMDA	BiC ($I \rightarrow (C \rightarrow T)$)	0.759±0.0091	0.804±0.0074	0.617±0.0086	0.452±0.0098	196.4M
-	SOTA	0.765⁵	0.834⁶	0.583 ⁷	0.447 ⁷	

Table 5.3: Evaluation results of the unimodal baseline architectures and the proposed approaches for assessing the performance of the CMDA mechanism in AL detection in memes. We report the mean and standard deviation over 5 runs for each proposed approach, except for the Gemini 1.5 baseline due to request limitations to the server, and for the SOTA models, as we used the results reported on the respective leaderboards. NOTE: The column “#P” indicates the number of parameters of each approach. The letters I , C , and T refer to the use of the image modality, the captions extracted from the image, and the text extracted from the meme, respectively.

Additionally, Table 5.3 reports the evaluation results of the CMA mechanism applied to the latent adaptation of text over an image and vice versa (rows 9 and 10). As observed, better results are achieved when incorporating visual information into the text modality (row 10), which aligns with the findings from the unimodal evaluation (rows 2, 3, and 4).

Subsequently, the table presents the results obtained with the integration of the CMDA mechanism for the latent adaptation of text over image and vice versa (rows 11 and 12). Consistently, better results are achieved when visual information is integrated into the text modality (row 12). Moreover, when comparing the results of the CMDA mechanism against CMA, a consistent improvement is observed across all evaluation datasets (row 12 vs. row 10 and row 11 vs. row 9). The table also reports the evaluation results of the Bi-contextual architecture integrating both the CMA and CMDA mechanisms (rows 13 and 14). As shown, the CMDA mechanism achieves superior results across all evaluation datasets, significantly outperforming

⁵(Zhu, 2020)

⁶(Zhang and Wang, 2022)

⁷(Wang and Markov, 2024)

its counterpart. It is worth noting that all possible permutations of the three considered modalities were tested. The best results were obtained by integrating visual information over the inclusion of captions into the meme text (row 14).

When comparing the best results obtained using the Bi-contextual architecture with the CMDA mechanism against the SOTA approaches, it can be observed that the significant gap identified in the proposed approaches in Chapter 4 was notably reduced. Furthermore, better results were achieved in 2 out of 4 datasets. For the datasets where the results did not surpass those of the SOTA approaches, the performance gap did not exceed 3.7%. Additionally, Table 5.3 compares the number of parameters across the different models. As shown, integrating the CMDA mechanism into the various encoder architectures does not lead to a significant increase in the number of parameters compared to the CMA mechanism within the same configurations. Notably, our best model is over 1,000 times smaller than the Gemini 1.5 model, a multimodal SOTA LLM. Furthermore, our best approach outperformed Gemini 1.5 across all evaluation datasets, further demonstrating the robustness of the proposed CMDA mechanism and its effectiveness when integrated with the Bi-contextual architecture.

5.3.2. On the Relevance of the Cross-Modal Dual Attention Mechanism

To further analyze the performance of the proposed CMDA mechanism, we focus on evaluating the complementarity and diversity of its different configurations, including the Bi-contextual architecture and its counterparts based on the CMA mechanism. As in Subsection 4.4.3, we use the MPA metric to assess complementarity and the CFD metric to evaluate diversity.

Table 5.4 reports the results for both MPA and CFD across the three AL meme detection datasets. As observed, when comparing the MPA scores of the CMA-based configurations (column 2) with those of the CMDA-based ones (column 3), the latter consistently achieve better complementarity, with improvements of up to 4.1% in MPA across all datasets.

An interesting observation is that datasets with a higher number of classes tend to exhibit lower MPA values, likely due to the increased complexity of distinguishing among multiple classes. This trend is evident when comparing the results of Task 1

and Task 2 of the DIMEMEX dataset (row 4 vs. row 5).

Dataset	MPA(CMA)	MPA(CMDA)	CFD(CMA)	CFD(CMDA)
HMC	0.7972	0.8304	0.2764	0.2590
MAMI	0.9080	0.9273	0.3406	0.3104
DIMEMEX ST1	0.8823	0.9141	0.4292	0.3819
DIMEMEX ST2	0.8629	0.8856	0.4870	0.4605

Table 5.4: Comparison of the complementarity and error diversity between the proposed CMDA-based configurations, including the Bi-contextual architecture, and their counterparts based on the CMA mechanism.

Finally, Table 5.4 also presents a comparison of error diversity using the CFD metric (column 4 for CMA vs. column 5 for CMDA). CMA-based configurations consistently exhibit higher CFD scores, suggesting greater error diversity. In contrast, our CMDA mechanism shows more stable behavior when distinguishing between offensive and non-offensive instances. Additionally, error diversity tends to increase with the number of classes in the dataset. For example, CFD values for Tasks 1 and 2 of DIMEMEX are significantly higher than those for the HMC and MAMI datasets, which are based on binary classification. Consequently, the highest CFD values are observed in Task 2 of DIMEMEX.

5.3.3. Statistical Significance Analysis

As in the previous chapter, we used a Bayesian Wilcoxon signed-rank test to evaluate the significance of integrating the CMDA mechanism into the encoding architectures, as well as the Bi-contextual architectures. This test calculates the posterior probabilities of the null and alternative hypotheses given the observed data, providing a clear probability of one approach being superior to the other. For this analysis, we define methods A and B according to the integration of the proposed CMDA variants. Table 5.5 presents the results of this statistical analysis over the macro-F1 scores, where the notation “>” denotes “*better than*”.

We observe a very high probability (> 0.9274) that the CMDA mechanism outperforms the CMA mechanism. Additionally, when comparing the Bi-contextual architecture to the standalone use of the CMDA mechanism, there is a very high probability (> 0.9335) of achieving better results with the Bi-contextual variant, highlighting the advantages of incorporating all three modalities. Finally, Table 5.5 also presents a comparative analysis between the results obtained by our best approach and those achieved by the Hate-CLIPper model, a robust and high-performing system evaluated for AL detection in memes. As shown, there is a probability greater than ($>$

0.8935) that our approach yields superior results, providing further evidence of its effectiveness.

Encoding Architectures	A	B	$p(A > B)$	$p(\text{rope})$	$p(B > A)$
BERT/BETO and ViT	CMDA	CMA	0.9274	0.0723	0.0001
BERT/BETO and ViT	Bi-C(CMDA)	CMDA	0.9335	0.0576	0.0088
BERT/BETO and ViT	Bi-C(CMDA)	Hate-CLIPper	0.8935	0.0021	0.1042

Table 5.5: Bayesian signed-rank test results for each proposed approach. The A and B columns indicate the integration of the proposed CMDA mechanism variants, as well as the proposed baseline approaches, over the encoding architectures; Bi-C(CMDA) denotes the results of the Bi-contextual architecture Bi-C ($I \rightarrow (C \rightarrow T)$).

The visual results of this analysis are presented in Figure 5.4, where each point represents a statistical comparison between two encoding approaches. Each vertex of the triangle corresponds to a potential outcome of the comparison for the following strategies: Subsection a illustrates the comparison between the CMDA mechanism and the CMA mechanism. Subsection b presents the comparison between the proposed Bi-contextual architecture and the standalone use of the CMDA mechanism. Finally, subsection c showcases the results of the comparison between our best approach and the Hate-CLIPper model.

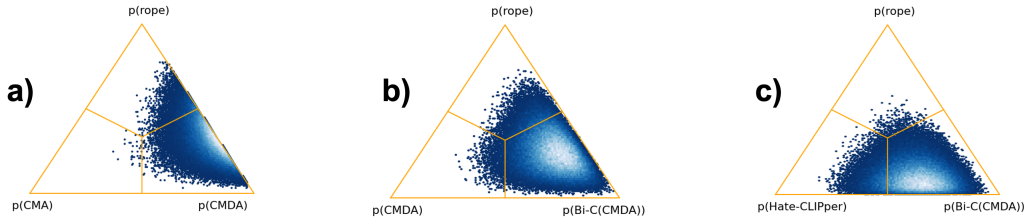


Figure 5.4: Visualization of the Bayesian Test for comparing: 1) the integration of the CMDA mechanism vs. the CMA mechanism (subsection a), 2) the integration of the Bi-contextual architecture with the CMDA mechanism vs. the standalone use of the CMDA mechanism, where the best configurations obtained were compared in both cases (subsection b), and 3) our best approach (the Bi-contextual architecture) vs. the Hate-CLIPper model (subsection c).

5.4. Analysis of Results

This section provides a qualitative analysis of the results to gain a deeper understanding of the strengths and weaknesses of the proposed CMDA mechanism. The

section is divided into two subsections. The first subsection presents a visualization of the attention values for various offensive and non-offensive memes. The second subsection offers an error analysis of the instances that were misclassified using the CMDA mechanism and the Bi-contextual architecture.

NOTE: This section contains examples that may be offensive to some readers, these do not represent the perspectives of the authors.

5.4.1. Visualization of Attention Values in the CMDA Mechanism

This subsection presents the visualization of the attention values generated by the CMDA mechanism. These values were extracted from the attention kernel $D_{\beta \rightarrow \alpha}$. It is important to note that the attention values for each word were calculated by averaging the tokens that composed the word (e.g., if the word “running” was segmented into the tokens “run##” and “##ing”, these were averaged to improve visualization). Similarly, the regions of the images were also averaged to enhance their visualization, providing a clearer representation of the attention distribution. Figures 5.5 and 5.6 present examples of the visualization of attention values for an instance containing hate speech and another non-offensive one, respectively.

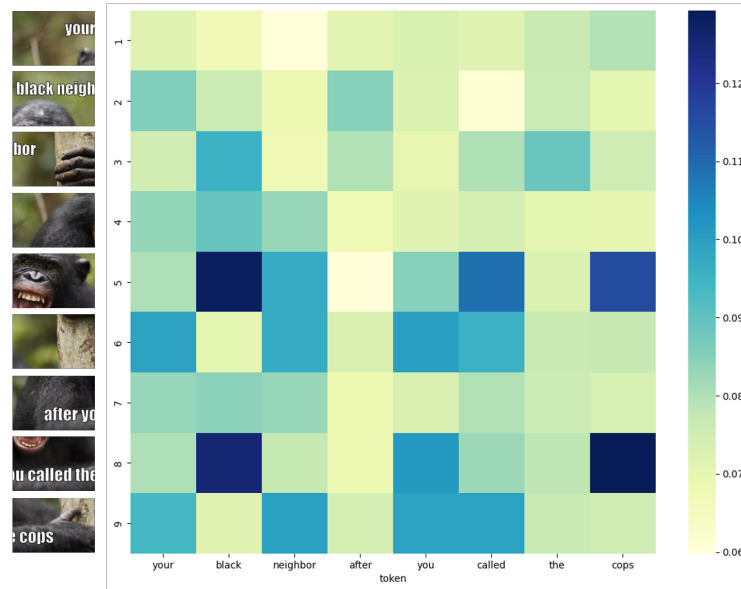


Figure 5.5: Visualization of the attention values for an instance containing hate speech. This example was taken from the HMC dataset (Kiel, Wang, and Cho, 2018).

As observed in the heatmap distributions, particularly in Figure 5.5, words related to the target of an offense, such as the word “*black*”, are highlighted along with the image regions associated with the insult—in this case, the face of the chimpanzee. On the other hand, in Figure 5.6, which contains no offensive content, the most relevant regions are associated with the humorous aspect of the meme.

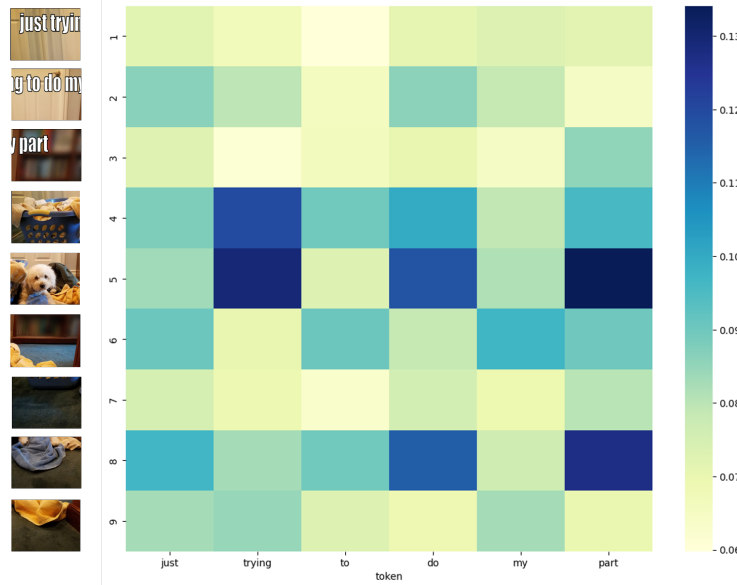


Figure 5.6: Visualization of the attention values for a non-offensive instance. This example was taken from the HMC dataset (Kiel, Wang, and Cho, 2018).

5.4.2. Error Analysis

This subsection presents an error analysis to gain a better understanding of the opportunities for improvement in misclassified instances. For this analysis, we extracted various examples that could not be correctly classified by the CMDA mechanism in its best variant (integrating visual information into text modality) and were also misclassified by our best approach (the Bi-contextual architecture integrated with the CMDA mechanism). Particularly, a manual qualitative analysis was performed over a subset of these memes. In the following paragraphs, we briefly describe the main observed features as well as some samples of these instances.

Table 5.6 shows examples of memes misclassified by the proposed CMDA and Bi-contextual architecture. A common denominator found in most of these memes is the need for extra-linguistic context for correct interpretation. For instance, the second

meme in the upper right corner adds the suffix “*tl*” to certain words, it is commonly used to mock people who speak indigenous languages in Mexico. Another fundamental aspect for detecting inappropriate content and hate speech in memes is the correct interpretation of both image and text. This is exemplified by the third meme in the middle left, where the textual content “*Science has reached the new switches*” seems harmless, but the addition of visual content referencing intimate body parts turns it into vulgar content with sexual connotations.






Category: Hate	Category: Hate
	
Translation: When you see that your parents address is on Insurgentes Sur.	Translation: F**k it life goes on.
Category: Inappropriate	Category: Inappropriate
	
Translation: Science has reached the new switches.	Translation: Life is like a priest, you never know what you're going to get. ⁵
Category: Neither	
	
Translation: - Every time we talk, I end up wet. - Do I turn you on? - No, you spit when you talk.	

Table 5.6: Samples of memes that were incorrectly classified by the proposed CMDA and Bi-contextual architecture. These memes were taken from the DIMEMEX dataset of subtask 1 (Jarquín-Vásquez et al., 2024).

⁵The verb *get* in Spanish has various meanings. In the context of this sentence, it can be interpreted as “touch”.

We also identified some instances that even labeled incorrectly by the proposed approaches may not be considered as mistakes at all. An example is the last meme, which belongs to the Neither category but was classified as inappropriate content by the proposed approaches, likely due to the inclusion of an initial conversation with sexual overtones.

Finally, Table 5.7 shows examples of memes misclassified by the proposed CMDA and Bi-contextual architecture in Subtask 2 of the DIMEMEX dataset. Again, the need for extra-linguistic context is evident for correctly classifying the different types of hate speech in memes. For example, the fourth meme in the lower right corner uses the expression “*Soviet tank*” to mock an overweight person. The correct interpretation of both image and text is also crucial, as seen in the third meme in the lower left corner, where a drawing of a man practicing various strikes, combined with the textual content, promotes violence against women. These characteristics identified in misclassified memes reveal the complexity of this task, as well as the low performance achieved by participating teams in Subtask 2⁶. They also highlight the need for new multimodal models and resources in Spanish.

Category: Classism	Category: Racisms
<p>cuando estas con tu celular en la calle y un niño te pregunta si tienes free fire:</p>	<p>Cuando te dice “Mi chocolatito” pero cuando esta enojada te grita: Cállate Coca cola con ojos.</p>
<p>When you’re on your phone in the street and a kid asks if you have Free Fire - let me guess, public school?</p>	<p>Translation: When she calls you ‘my little chocolate’— when she’s mad she says: shut up, Coke bottle with eyes.</p>
Category: Sexism	Category: Other
<p>cómo que no</p> <p>cocinaste nada</p>	<p>*La morra que pesa más que un tanque soviético dice que le gustan los chicos de color*</p> <p>El chico de color del curso:</p>
<p>Translation: What do you mean you didn’t cook anything?</p>	<p>Translation: The girl who weighs more than a Soviet tank says she likes black guys. The black guy in the class:</p>

Table 5.7: Samples of memes that were incorrectly classified by the proposed CMDA and Bi-contextual architecture. These memes were taken from the DIMEMEX dataset of subtask 2 (Jarquín-Vásquez et al., 2024).

⁶<https://codalab.lisn.upsaclay.fr/competitions/18118#results>

Conclusions and Future Work

This chapter presents the conclusions and future work obtained from this research. This chapter is divided into three sections. The first section addresses the research questions formulated during this doctoral research. The second section outlines the general conclusions drawn from this research. Finally, the third section discusses future work, highlighting opportunities identified from the results obtained through the proposed DA and CMDA mechanisms.

6.1. Addressing the Research Questions

This section provides detailed answers to the research questions posed in this doctoral investigation:

RQ1: Which fusion approach yields the best performance in the integration of SA and CA mechanisms for the task of AL detection?

As an initial experiment in detecting AL in textual data, an early fusion approach was proposed. This approach involved concatenating the SA and CA representations. While this method showed consistent improvements in AL detection for text, it was not the most effective fusion approach for the task. Motivated by the promising results of this preliminary approach, the research progressed to propose more sophisticated mechanisms: Dual Attention and Cross-Modal Dual Attention.

Unlike the early fusion approach, DA and CMDA integrate the representations at the attention computation stage, enabling the model to learn a more nuanced representation of the relevance of elements within one or more input sequences. This deeper integration allows the network to better capture the underlying relationships between the input modalities. Overall, this fusion strategy yielded superior performance across

various datasets.

It is important to note, however, that a direct comparison between the two mechanisms is not entirely fair due to their differing objectives: DA focuses on improving single-sequence input processing, while CMDA aims to capture relationships between elements of two distinct input sequences.

RQ2: Which deep learning architectures are best suited to incorporate the proposed attention mechanisms in terms of maximizing performance for detecting AL in both textual data and memes?

Throughout this doctoral investigation, various baseline models and encoding architectures were evaluated to incorporate the proposed DA and CMDA mechanisms. Among the architectures tested, the transformer-based models consistently demonstrated the best performance across all datasets for AL detection in both text and memes. This conclusion is supported by the results detailed in Sections 4.4 and 5.3.

Transformers, with their ability to model long-range dependencies and capture complex relationships within the data, proved to be highly effective in leveraging the proposed attention mechanisms. The combination of the transformer architecture with DA and CMDA mechanisms enabled the model to achieve state-of-the-art results in most of the tested scenarios, underscoring its suitability for this research.

RQ3: What are the most significant textual and visual features that contribute to the deep representation of text and images in the context of AL detection?

The most effective textual and visual features were those extracted using pre-trained transformer models. Specifically, for AL detection in memes, three distinct feature sets were utilized: the visual features derived from the image, the textual content of the meme, and the image captions describing the visual content.

Among these, the textual features extracted directly from the text of the memes yielded the highest performance. This was followed by visual features processed in grid form by the Vision Transformer (ViT). Finally, features derived from the image captions provided additional, though relatively weaker, contributions.

Importantly, the combination of all three feature sets enabled advanced integration strategies, such as those implemented in the Bi-Contextual architecture. This approach allowed latent adaptation of image features to text and captions, resulting in the best performance overall. The findings emphasize the critical role of textual features while demonstrating the complementary value of visual and caption-based

features.

RQ4: Is the unique integration of textual and visual modalities sufficient for the effective detection of AL in memes, or are additional sources of information required?

The exclusive use of textual and visual modalities for AL detection in memes has shown encouraging results, as both modalities are complementary. However, there remains considerable room for improvement. The sole reliance on these two modalities has proven to be insufficient in several cases, as evidenced by the error analysis presented in Section 5.4. In many instances, accurately distinguishing between offensive and non-offensive content required extralinguistic knowledge beyond what was provided in the text and image. This included understanding the irony and sarcasm embedded in memes, interpreting polysemous words whose meaning varies with context, and recognizing subtle, specific stereotypes associated with different categories of hate speech.

This limitation was further underscored by the significant performance improvements observed when incorporating image captions as an additional modality within the proposed Bi-Contextual architecture. The integration of captions provided the model with enriched contextual information, improving its ability to disambiguate nuanced or complex instances of AL in memes. These findings suggest that, although textual and visual modalities form a solid foundation, the inclusion of supplementary extralinguistic information is often essential to achieve strong performance in this challenging task, leaving ample room for further advancements in AL detection in memes.

6.2. Conclusions

This doctoral research proposed extensions to the SA and CA mechanisms, addressing their complementary limitations. The SA mechanism, while effective, disregards the global context learned during neural network training. In contrast, the CA mechanism overlooks internal relationships between pairs of elements within a sequence. To overcome these limitations and incorporate the strengths of both mechanisms, this research introduced the DA mechanism and its multimodal extension, the CMDA.

Given the significant impact of AL propagation on social media and its strong reliance on contextual understanding for accurate identification, we evaluated the

effectiveness of DA and CMDA mechanisms for AL detection in text and memes. Additionally, the proposed mechanisms were integrated into various encoding architectures, enabling performance comparisons across multiple models.

Performance of the DA Mechanism

The DA mechanism demonstrated promising results in AL detection for textual data, consistently improving performance across all evaluation datasets. The mechanism's ability to dynamically capture relationships within a sequence while leveraging global context proved beneficial in enhancing the detection of AL.

To further extend its capabilities, we proposed a multi-level application of the DA mechanism, resulting in the GHA architecture. This architecture combines weighted representations derived from applying DA at different encoding levels of deep neural networks. The multi-level extension significantly improved AL detection in text by dynamically integrating information from multiple encoding levels, thereby enhancing the model's ability to capture nuanced patterns.

For AL detection in memes, the DA mechanism also yielded encouraging results, consistently improving performance across all datasets. However, there remained substantial room for improvement compared to SOTA models, highlighting areas for further refinement.

Performance of the CMDA Mechanism

Building on these results, we developed the CMDA mechanism to more comprehensively model relationships between elements within a sequence. This mechanism was further extended to incorporate three modalities: the visual features from the image, the textual content of the meme, and the textual captions describing the image.

The results obtained using CMDA were highly encouraging, demonstrating consistent improvements across all evaluation datasets. Notably, the integration of CMDA into the Bi-Contextual architecture significantly enhanced performance. This architecture facilitated better alignment between features from different modalities, enabling the creation of a more sophisticated representation for AL detection in memes. The results underscore the effectiveness of CMDA in capturing cross-modal interactions and improving the interpretability and robustness of the model.

Insights from Error Analysis

The error analysis highlighted several areas of opportunity for future improvements. One notable challenge was the necessity of extralinguistic context for accurately classifying AL in memes. In many instances, the text and image alone were insufficient for disambiguating offensive content from non-offensive content. This observation underscores the need for incorporating additional contextual information, such as cultural or situational knowledge, to further enhance the effectiveness of AL detection models.

Final Remarks

Overall, this research demonstrated the potential of DA and CMDA mechanisms to address critical limitations in existing attention mechanisms and provided novel contributions to the fields of AL detection and multimodal learning. The results achieved with the proposed mechanisms across textual and multimodal datasets reflect significant progress while identifying clear directions for future exploration.

6.3. Future Work

As part of future work, we outline several directions to extend and enhance the contributions of this doctoral research:

- 1.- **Evaluating the DA and CMDA mechanisms in other classification tasks:** One promising avenue for future research involves testing the effectiveness of the DA and CMDA mechanisms in other classification tasks where accurate contextual interpretation is crucial. For instance, sentiment classification, both in textual and multimodal settings, could benefit significantly from these mechanisms. Exploring these applications will help generalize the proposed approaches and assess their robustness in diverse domains where understanding nuanced contextual information is essential.
- 2.- **Incorporating DA and CMDA mechanisms into new training techniques for LLMs:** Another key direction is to integrate DA and CMDA mechanisms into emerging training strategies for LLMs that focus on reducing the number of trainable parameters, such as Low-Rank Adaptation (LoRA). This

integration could extend the applicability of the proposed mechanisms beyond encoding architectures, enabling their adoption in decoding-based architectures as well. By leveraging parameter-efficient techniques, this approach may facilitate scaling to larger datasets and more computationally demanding tasks while maintaining model efficiency.

3.- Combining predictions from LLMs and the proposed architectures:

We plan to explore methods for combining the predictions of state-of-the-art LLMs with those of the proposed DA and CMDA-based architectures. Such ensemble approaches could capitalize on the strengths of both methodologies, leading to improved performance in AL classification for both text and memes. This hybrid strategy has the potential to refine decision-making processes by balancing the contextual depth of DA/CMDA mechanisms with the broader generalization capabilities of LLMs.

4.- Enhancing image descriptions with advanced LLMs for improved AL

detection in memes: A specific area of improvement involves utilizing more advanced LLMs, such as GPT-4, to generate richer and more accurate image descriptions. Enhanced image captions could provide critical contextual insights, ultimately improving the performance of AL detection in memes. In parallel, we plan to investigate new feature fusion approaches that better integrate visual, textual, and caption-based modalities.

These directions not only build on the strengths of this doctoral research but also aim to address its limitations while paving the way for broader applicability and impact in related fields.

Bibliography

- Abdel-Jaber, H.; Devassy, D.; Al Salam, A.; Hidaytallah, L.; and EL-Amir, M. 2022. A review of deep learning algorithms and their applications in healthcare. *Algorithms* 15(2).
- Abro, S.; Shaikh, S.; Khand, Z. H.; Ali, Z.; Khan, S.; and Mujtaba, G. 2020. Automatic hate speech detection using machine learning: A comparative study. *International Journal of Advanced Computer Science and Applications* 11(8).
- Afridi, T. H.; Alam, A.; Khan, M. N.; Khan, J.; and Lee, Y. 2020. A multimodal memes classification: A survey and open research issues. *CoRR* abs/2009.08395.
- Aggarwal, C., and Zhai, C. 2012. A survey of text classification algorithms. *Mining Text Data* 9781461432234:163–222.
- Aggarwal, C. C. 2018. *Neural Networks and Deep Learning: A Textbook*. Springer Publishing Company, Incorporated, 1st edition.
- Alom, M. Z.; Taha, T. M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M. S.; Hasan, M.; Van Essen, B. C.; Awwal, A. A. S.; and Asari, V. K. 2019. A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8(3).
- Alzubaidi, L.; Zhang, J.; Humaidi, A. J.; Al-dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M. A.; Al-Amidie, M.; and Farhan, L. 2021. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data* 8.
- Aragón, M. E.; Jarquín, H.; Montes-y Gómez, M.; Escalante, H. J.; Villaseñor-Pineda, L.; Gómez Adorno, H.; Bel-Enguix, G.; and Posadas-Durán, J. P. 2020. Overview

- of mex-a3t at iberlef 2020: Fake news and aggressiveness analysis case study in mexican spanish. In *Notebook Papers of 2nd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Malaga, Spain, September*.
- Aragon, M.; Lopez Monroy, A. P.; Gonzalez, L.; Losada, D. E.; and Montes, M. 2023. DisorBERT: A double domain adaptation model for detecting signs of mental disorders in social media. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15305–15318. Toronto, Canada: Association for Computational Linguistics.
- Arevalo, J.; Solorio, T.; Montes, M.; and González, F. 2020. Gated multimodal networks. *Neural Computing and Applications* 1433–3058.
- Arya, G.; Hasan, M. K.; Bagwari, A.; Safie, N.; Islam, S.; Ahmed, F. R. A.; De, A.; Khan, M. A.; and Ghazal, T. M. 2024. Multimodal hate speech detection in memes using contrastive language-image pre-training. *IEEE Access* 12:22359–22375.
- Asakawa, S. 2016. Comparison among lstm, gru, and rnn, and their cross products. In *Proceedings of the 14th Conference of the Japanese Society for Cognitive Psychology*, volume 2016, 259–271.
- Ashwin, G. D.; Irina, I.; and Dominique, F. 2020. Classification of hate speech using deep neural networks. *Revue de l'Information Scientifique et Technique* 25(1):1–12.
- Badjatiya, P.; Gupta, S.; Gupta, M.; and Varma, V. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.
- Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- Baltrusaitis, T.; Ahuja, C.; and Morency, L.-P. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 41(2):423–443.

- Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel Pardo, F. M.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In May, J.; Shutova, E.; Herbelot, A.; Zhu, X.; Apidianaki, M.; and Mohammad, S. M., eds., *Proceedings of the 13th International Workshop on Semantic Evaluation*, 54–63. Minneapolis, Minnesota, USA: Association for Computational Linguistics.
- Baziotis, C.; Pelekis, N.; and Doukeridis, C. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 747–754. Vancouver, Canada: Association for Computational Linguistics.
- Bel-Enguix, G.; Gómez-Adorno, H.; Sierra, G.; Vásquez, J.; Andersen, S.-T.; and Ojeda-Trueba, S. 2023. Overview of homo-mex at iberlef 2023: Homo-mex: Hate speech detection in online messages directed towards the mexican spanish speaking lgbtq+ population. *Procesamiento del Lenguaje Natural* 71.
- Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. Hong Kong, China: Association for Computational Linguistics.
- Benavoli, A.; Corani, G.; Mangili, F.; Zaffalon, M.; and Ruggeri, F. 2014. A bayesian wilcoxon signed-rank test based on the dirichlet process. In *Proceedings of ICML*, volume 32 of *Proceedings of Machine Learning Research*, 1026–1034. Beijing, China: PMLR.
- Benavoli, A.; Corani, G.; Demsar, J.; and Zaffalon, M. 2017. Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis. *J. Mach. Learn. Res.* 18(1):2653–2688.
- Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning* 2(1):1–127.
- Bindra, M.; Sharma, B.; and Bansal, N. 2022. Detecting hate speech and offensive language using transformer techniques. In *Micro-Electronics and Telecommunication Engineering*, 703–715. Springer.

- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.
- Burnap, P., and Williams, M. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science* 5:11. <https://doi.org/10.1140/epjds/s13688-016-0072-6>.
- Caselli, T.; Basile, V.; Mitrović, J.; and Granitzer, M. 2021. HateBERT: Retraining BERT for abusive language detection in English. In Mostafazadeh Davani, A.; Kiela, D.; Lambert, M.; Vidgen, B.; Prabhakaran, V.; and Waseem, Z., eds., *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, 17–25. Online: Association for Computational Linguistics.
- Cecillon, N.; Labatut, V.; Dufour, R.; and Linares, G. 2019. Abusive language detection in online conversations by combining content- and graph-based features. *Frontiers in Big Data* 2.
- Chakrabarty, T.; Gupta, K.; and Muresan, S. 2019. Pay “attention” to your context when classifying abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, 70–79. Association for Computational Linguistics.
- Chaudhari, S.; Mithal, V.; Polatkan, G.; and Ramanath, R. 2021. An attentive survey of attention models. *ACM Trans. Intell. Syst. Technol.* 12(5).
- Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 1–9.
- Clark, K.; Khandelwal, U.; Levy, O.; and Manning, C. D. 2019. What does BERT look at? an analysis of BERT’s attention. In Linzen, T.; Chrupała, G.; Belinkov, Y.; and Hupkes, D., eds., *Proceedings of the 2019 ACL Workshop BlackboxNLP*:

- Analyzing and Interpreting Neural Networks for NLP*, 276–286. Florence, Italy: Association for Computational Linguistics.
- Cohen, V., and Gokaslan, A. 2020. Opengpt-2: Open language models and implications of generated text. *XRDS* 27(1):26–30.
- Cohen, J. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1):37.
- Constantin, M. G.; Pârvu, D.-S.; Stanciu, C.; Ionascu, D.; and Ionescu, B. 2021. Hateful meme detection with multimodal deep neural networks. In *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, 1–4.
- Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; and Villata, S. 2020. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology* 20:1–22.
- Davidson, T.; Warmusley, D.; Macy, M. W.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, 512–515. AAAI Press.
- del Arco, F. M. P.; Casavantes, M.; Escalante, H. J.; Martín-Valdivia, M. T.; Montejo-Ráez, A.; y Gómez, M. M.; Jarquín-Vásquez, H.; and Villaseñor-Pineda, L. 2021. Overview of meoffendes at iberlef 2021: Offensive language detection in spanish variants. In *Notebook Papers of 3rd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF)*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.
- Ding, S.; Shang, J.; Wang, S.; Sun, Y.; Tian, H.; Wu, H.; and Wang, H. 2021. ERNIE-Doc: A retrospective long-document modeling transformer. In *Proceedings ACL*, 2914–2927.

- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv* abs/2010.11929.
- Durairaj, A. K., and Chinnalagu, A. 2021. Transformer based contextual model for sentiment analysis of customer reviews: A fine-tuned bert. *International Journal of Advanced Computer Science and Applications* 12(11).
- Duwairi, R.; Hayajneh, A.; and Quwaider, M. 2021. A deep learning framework for automatic detection of hate speech embedded in arabic tweets. *Arabian Journal for Science and Engineering* 46:1–14.
- Fersini, E.; Gasparini, F.; Rizzi, G.; Saibene, A.; Chulvi, B.; Rosso, P.; Lees, A.; and Sorensen, J. 2022. SemEval-2022 task 5: Multimedia automatic misogyny identification. In Emerson, G.; Schluter, N.; Stanovsky, G.; Kumar, R.; Palmer, A.; Schneider, N.; Singh, S.; and Ratan, S., eds., *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 533–549. Seattle, United States: Association for Computational Linguistics.
- Fersini, E.; Nozza, D.; and Rosso, P. 2018. Overview of the evalita 2018 task on automatic misogyny identification (AMI). In Caselli, T.; Novielli, N.; Patti, V.; and Rosso, P., eds., *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it)*, volume 2263 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Fleiss, J., et al. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5):378–382.
- Fortuna, P., and Nunes, S. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys* 51(4).
- Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3141–3149.

- Gambäck, B., and Sikdar, U. K. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, 85–90. Association for Computational Linguistics.
- Gardner, J., and Brooks, C. 2017. A statistical framework for predictive model evaluation in moocs. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale, L@S '17*, 269–272. New York, NY, USA: Association for Computing Machinery.
- Gaydhani, A.; Doma, V.; Kendre, S.; and B B, L. 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. In *IEEE International Advance Computing Conference 2018*.
- Gemini-Team; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; Mariooryad, S.; Ding, Y.; and Geng, X. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.
- Golbeck, J.; Ashktorab, Z.; Banjo, R. O.; Berlinger, A.; Bhagwan, S.; Buntain, C.; Cheakalos, P.; Geller, A. A.; Gergory, Q.; Gnanasekaran, R. K.; Gunasekaran, R. R.; Hoffman, K. M.; Hottle, J.; Jienjiltter, V.; Khare, S.; Lau, R.; Martindale, M. J.; Naik, S.; Nixon, H. L.; Ramachandran, P.; Rogers, K. M.; Rogers, L.; Sarin, M. S.; Shahane, G.; Thanki, J.; Vengataraman, P.; Wan, Z.; and Wu, D. M. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci '17*, 229–233. New York, NY, USA: Association for Computing Machinery.
- Gomez, R.; Gibert, J.; Gómez, L.; and Karatzas, D. 2020. Exploring hate speech detection in multimodal publications. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1459–1467.
- Guberman, J., and Hemphill, L. 2017. Challenges in modifying existing scales for detecting harassment in individual tweets. *Proceedings of 50th Annual Hawaii International Conference on System Sciences (HICSS)*.
- Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; and Pang, R. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, 5036–5040.

- Guo, W.; Wang, J.; and Wang, S. 2019. Deep multimodal representation learning: A survey. *IEEE Access* 7:63373–63394.
- Haque, F.; Un Nur, R.; Jahan, S.; Mahmud, Z.; and Shah, F. 2020. A transformer based approach to detect suicidal ideation using pre-trained language models. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, 1–5.
- Hermida, P. C. D., and Santos, E. 2023. Detecting hate speech in memes: a review. *Artificial Intelligence Review* 56:1–19.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Hossin, M., and Sulaiman, M. N. 2015. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process* 5(2):1.
- Hu, D. 2020. An introductory survey on attention mechanisms in nlp problems. In Bi, Y.; Bhatia, R.; and Kapoor, S., eds., *Intelligent Systems and Applications*, 432–448. Cham: Springer International Publishing.
- Huynh, T.; Nguyen, D.-V.; Nguyen, K.; Nguyen, N.; and Nguyen, A. 2019. Hate speech detection on vietnamese social media text using the bi-gru-lstm-cnn model. In *Proceedings of the Sixth International Workshop on Vietnamese Language and Speech Processing (VLSP 2019)*.
- Jahan, M. S., and Oussalah, M. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing* 546:126232.
- Jarquín-Vásquez, H.; Escalante, H. J.; and Montes-y Gómez, M. 2023. Improving the identification of abusive language through careful design of pre-training tasks. In Rodríguez-González, A. Y.; Pérez-Espinosa, H.; Martínez-Trinidad, J. F.; Carrasco-Ochoa, J. A.; and Olvera-López, J. A., eds., *Pattern Recognition*, 283–292. Cham: Springer Nature Switzerland.
- Jarquín-Vásquez, H.; Escalante, H. J.; and Montes, M. 2021. Self-contextualized attention for abusive language identification. In *Proceedings of the Ninth Internatio-*

- nal Workshop on Natural Language Processing for Social Media*, 103–112. Online: Association for Computational Linguistics.
- Jarquín-Vásquez, H. J.; Montes-y Gómez, M.; and Villaseñor-Pineda, L. 2020. Not all swear words are used equal: Attention over word n-grams for abusive language identification. In Figueroa Mora, K. M.; Anzurez Marín, J.; Cerda, J.; Carrasco-Ochoa, J. A.; Martínez-Trinidad, J. F.; and Olvera-López, J. A., eds., *Pattern Recognition*, 282–292. Cham: Springer International Publishing.
- Jarquín-Vásquez, H.; Tlelo-Coyotecatl, I.; Casavantes, M.; Hernández-Farías, D. I.; Escalante, H. J.; Villaseñor-Pineda, L.; and y Gómez, M. M. 2024. Overview of dimemex at iberlef 2024: Detection of inappropriate memes from mexico. In Jiménez-Zafra, S. M.; Chiruzzo, L.; Rangel, F.; Corrêa, U. B.; Jover, A. B.; Gómez-Adorno, H.; Barba, J. Á. G.; Farías, D. I. H.; Ráez, A. M.; Moral, P.; Abellán, C. R.; Rodríguez, M. E. V.; Taulé, M.; and Valencia-García, R., eds., *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2024)*, Valladolid, Spain, September, 2024, CEUR Workshop Proceedings. Valladolid, Spain: CEUR-WS.org.
- Kanakamedala, D.; Veeranki, T.; Bitla, R.; Vangalapudi, S.; and T, S. 2021. Visual question answering using deep learning. In *2021 Innovations in Power and Advanced Computing Technologies*, 1–5.
- Keswani, V.; Singh, S.; Agarwal, S.; and Modi, A. 2020. Iitk at semeval-2020 task 8: Unimodal and bimodal sentiment analysis of internet memes. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*.
- Khan, S.; Naseer, M.; Hayat, M.; Zamir, S. W.; Khan, F. S.; and Shah, M. 2021. Transformers in vision: A survey. *ACM Comput. Surv.* Just Accepted.
- Khan, S.; Naseer, M.; Hayat, M.; Zamir, S. W.; Khan, F. S.; and Shah, M. 2022. Transformers in vision: A survey. *ACM Comput. Surv.* 54(10s).
- Kiela, D.; Firooz, H.; Mohan, A.; Goswami, V.; Singh, A.; Ringshia, P.; and Testugine, D. 2020. The hateful memes challenge: Detecting hate speech in multimodal

- memes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Kiela, D.; Wang, C.; and Cho, K. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of EMNLP*, 1466–1477.
- Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In Bengio, Y., and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kiritchenko, S., and Nejadgholi, I. 2020. Towards ethics by design in online abusive content detection. *CoRR* abs/2010.14952.
- Kirk, H.; Jun, Y.; Rauba, P.; Wachtel, G.; Li, R.; Bai, X.; Broestl, N.; Doff-Sotta, M.; Shtedritski, A.; and Asano, Y. M. 2021. Memes in the wild: Assessing the generalizability of the hateful memes challenge dataset. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, 26–35. Online: Association for Computational Linguistics.
- Kora, R., and Mohammed, A. 2023. A comprehensive review on transformers models for text classification. In *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 1–7.
- Kovács, G.; Alonso, P.; and Saini, R. 2021. Challenges of hate speech detection in social media: Data scarcity, and leveraging external resources. *SN Computer Science* 2(2).
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- Kumar, G. K., and Nandakumar, K. 2022. Hate-CLIPper: Multimodal hateful meme classification based on cross-modal interaction of CLIP features. In Biester, L.; Demszky, D.; Jin, Z.; Sachan, M.; Tetreault, J.; Wilson, S.; Xiao, L.; and Zhao, J., eds., *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*,

- 171–183. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics.
- Kumar, R.; Reganti, A. N.; Bhatia, A.; and Maheshwari, T. 2018. Aggression-annotated corpus of Hindi-English code-mixed data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA).
- la Peña Sarracén, G. L. D.; Pons, R. G.; Muñiz-Cuza, C. E.; and Rosso, P. 2018. Hate speech detection using attention-based lstm. In *EVALITA@CLiC-it*.
- Lan, Y.; Hao, Y.; Xia, K.; Qian, B.; and Li, C. 2020. Stacked residual recurrent neural networks with cross-layer attention for text classification. *IEEE Access* 8:70401–70410.
- LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436–444.
- Lee, R. K.-W.; Cao, R.; Fan, Z.; Jiang, J.; and Chong, W.-H. 2021. Disentangling hate in online memes. In *Proceedings of the 29th ACM International Conference on Multimedia, MM '21*, 5138–5147. New York, NY, USA: Association for Computing Machinery.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.
- Li, D., and Dong, Y. 2014. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing* 7(3–4):197–387.
- Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.; and Chang, K. 2019a. Visualbert: A simple and performant baseline for vision and language. *CoRR* abs/1908.03557.
- Li, R.; Lin, C.; Collinson, M.; Li, X.; and Chen, G. 2019b. A dual-attention hierarchical recurrent neural network for dialogue act classification. In Bansal, M.,

- and Villavicencio, A., eds., *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 383–392. Hong Kong, China: Association for Computational Linguistics.
- Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P. S.; and He, L. 2022. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol.* 13(2).
- Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org.
- Li, Y.; Chen, Y.; Shao, H.; and Zhang, H. 2023b. A novel dual attention mechanism combined with knowledge for remaining useful life prediction based on gated recurrent units. *Reliability Engineering & System Safety* 239:109514.
- Lin, T.; Wang, Y.; Liu, X.; and Qiu, X. 2022. A survey of transformers. *AI Open* 3:111–132.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR* abs/1907.11692.
- Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; Wei, F.; and Guo, B. 2022. Swin transformer v2: Scaling up capacity and resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11999–12009.
- Liu, P.; Li, W.; and Zou, L. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 87–91. Association for Computational Linguistics.
- Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. In *Proceedings of NIPS*, 289–297.
- Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

- MacAvaney, S.; Yao, H.-R.; Yang, E.; Russell, K.; Goharian, N.; and Frieder, O. 2019. Hate speech detection: Challenges and solutions. *PLOS ONE* 14(8):1–16.
- Maharjan, S.; Montes, M.; González, F. A.; and Solorio, T. 2018. A genre-aware attention model to improve the likability prediction of books. In *Proceedings of EMNLP*, 3381–3391.
- Malviya, K.; Roy, B.; and Saritha, S. 2021. A transformers approach to detect depression in social media. In *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 718–723.
- Mandl, T.; Modha, S.; Majumder, P.; Patel, D.; Dave, M.; Mandlia, C.; and Patel, A. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation, FIRE '19*, 14–17. New York, NY, USA: Association for Computing Machinery.
- Maqbool, F., and Fersini, E. 2024. Multimodal Hate Speech Detection in Memes from Mexico using BLIP. In Jiménez-Zafra, S. M.; Chiruzzo, L.; Rangel, F.; Corrêa, U. B.; Jover, A. B.; Gómez-Adorno, H.; Barba, J. Á. G.; Farías, D. I. H.; Ráez, A. M.; Moral, P.; Abellán, C. R.; Rodríguez, M. E. V.; Taulé, M.; and Valencia-García, R., eds., *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), Valladolid, Spain, September, 2024*, CEUR Workshop Proceedings. Valladolid, Spain: CEUR-WS.org.
- Marcos, Z.; Shervin, M.; Preslav, N.; Sara, R.; Farra, N.; and Kumar, R. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 75–86. Association for Computational Linguistics.
- Mazari, A. C.; Boudoukhani, N.; and Djeflal, A. 2023. Bert-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing* 27(1):325–339.
- Mikolov, T.; Grave, E.; Bojanowski, P.; Puhersch, C.; and Joulin, A. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

- Mnassri, K.; Rajapaksha, P.; Farahbakhsh, R.; and Crespi, N. 2022. Bert-based ensemble approaches for hate speech detection. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 4649–4654.
- Mogadala, A.; Kalimuthu, M.; and Klakow, D. 2021. Trends in integration of vision and language research: A survey of tasks, datasets, and methods. *Journal of Artificial Intelligence Research* 71:1183–1317.
- Mohammed, A. H., and Ali, A. H. 2021. Survey of BERT (bidirectional encoder representation transformer) types. In *2nd International Conference on Physics and Applied Sciences (ICPAS 2021)*, volume 1963, 012173. IOP Publishing.
- Mozafari, M.; Farahbakhsh, R.; and Crespi, N. 2019a. A BERT-based transfer learning approach for hate speech detection in online social media. In *Complex Networks 2019: 8th International Conference on Complex Networks and their Applications*, volume Studies in Computational Intelligence book series (SCI, volume 881) of *Complex Networks and Their Applications VIII : Volume 1, Proceedings of the Eighth International Conference on Complex Networks and Their Applications*, 928–940. Lisbonne, Portugal: Springer.
- Mozafari, M.; Farahbakhsh, R.; and Crespi, N. 2019b. A bert-based transfer learning approach for hate speech detection in online social media. In *Complex Networks and Their Applications VIII - Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10-12, 2019*, volume 881 of *Studies in Computational Intelligence*, 928–940. Springer.
- Mutanga, R. T.; Naicker, N.; and Olugbara, O. O. 2020. Hate speech detection in twitter using transformer methods. *International Journal of Advanced Computer Science and Applications* 11(9).
- Naseem, U.; Khan, S. K.; Farasat, M.; and ali, f. 2019. Abusive language detection: A comprehensive review. *Indian Journal of Science and Technology* 12:1–13.
- Nikolov, A., and Radivchev, V. 2019. Nikolov-radivchev at SemEval-2019 task 6: Offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 691–695. Minneapolis, Minnesota, USA: Association for Computational Linguistics.

- Niu, Z.; Zhong, G.; and Yu, H. 2021. A review on the attention mechanism of deep learning. *Neurocomputing* 452:48–62.
- Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, 145–153. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.
- Nowak, S., and Rüger, S. 2010. How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the International Conference on Multimedia Information Retrieval, MIR '10*, 557–566. New York, NY, USA: Association for Computing Machinery.
- Nozza, D.; Cignarella, A.; Damo, G.; Caselli, T.; and Patti, V. 2023. Hodi at evalita 2023: Overview of the first shared task on homotransphobia detection in italian. In Lai, M.; Menini, S.; Polignano, M.; Russo, V.; Sprugnoli, R.; and Venturi, G., eds., *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, CEUR Workshop Proceedings. CEUR Workshop Proceedings (CEUR-WS.org). Publisher Copyright: © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).; 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2023 ; Conference date: 07-09-2023 Through 08-09-2023.
- OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; and Babuschkin, I. 2024. Gpt-4 technical report.
- Oriol, B.; Canton-Ferrer, C.; and i Nieto, X. G. 2019. Hate speech in pixels: Detection of offensive memes towards automatic moderation. In *NeurIPS 2019 Workshop on AI for Social Good*.
- Pavlopoulos, J.; Malakasiotis, P.; and Androutsopoulos, I. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1125–1135. Copenhagen, Denmark: Association for Computational Linguistics.

- Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. New Orleans, Louisiana: Association for Computational Linguistics.
- Pitsilis, G. K.; Ramampiaro, H.; and Das, Langseth, H. 2018. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence* 48:4730–4742.
- Plaza, L.; Carrillo-de Albornoz, J.; Amigó, E.; Gonzalo, J.; Morante, R.; Rosso, P.; Spina, D.; Chulvi, B.; Maeso, A.; and Ruiz, V. 2024. Exist 2024: sexism identification in social networks and memes. In *Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part V*, 498–504. Berlin, Heidelberg: Springer-Verlag.
- Poletto, F.; Basile, V.; Sanguinetti, M.; Bosco, C.; and Patti, V. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*.
- Pu, Y.; Min, M.; Gan, Z.; and Carin, L. 2018. Adaptive feature abstraction for translating video to text. *Proceedings of the AAAI Conference on Artificial Intelligence* 32(1).
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *Technical report, OpenAi* 24.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.
- Ramprasath, M.; Dhanasekaran, K.; Karthick, T.; Velumani, R.; and Sudhakaran, P. 2022. An extensive study on pretrained models for natural language processing based on transformers. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, 382–389.

- Rani, P.; Suryawanshi, S.; Goswami, K.; Chakravarthi, B. R.; Fransen, T.; and McCrae, J. P. 2020. A comparative study of different state-of-the-art hate speech detection methods in Hindi-English code-mixed data. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, 42–48. Marseille, France: European Language Resources Association (ELRA).
- Rey, D., and Neuhäuser, M. 2011. Wilcoxon-signed-rank test. *International Encyclopedia of Statistical Science* 1658–1659.
- Roy, P. K.; Tripathy, A. K.; Das, T. K.; and Gao, X.-Z. 2020. A framework for hate speech detection using deep convolutional neural network. *IEEE Access* 8:204951–204962.
- Saha, P.; Mathew, B.; Goyal, P.; and Mukherjee, A. 2018. Hateminers : Detecting hate speech against women. *CoRR* abs/1812.06700.
- Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; Mondal, S.; and Chadha, A. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications.
- Saksesi, A. S.; Nasrun, M.; and Setianingsih, C. 2018. Analysis text of hate speech detection using recurrent neural network. In *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 242–248.
- Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108.
- Satapara, S.; Masud, S.; Madhu, H.; Khan, M. A.; Akhtar, M. S.; Chakraborty, T.; Modha, S.; and Mandl, T. 2024. Overview of the hasoc subtracks at fire 2023: Detection of hate spans and conversational hate-speech. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE '23*, 10–12. New York, NY, USA: Association for Computing Machinery.
- Schmidt, A., and Wiegand, M. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1–10. Valencia, Spain: Association for Computational Linguistics.

- Sharma, C.; Bhageria, D.; Scott, W.; PYKL, S.; Das, A.; Chakraborty, T.; Pulabai-gari, V.; and Gambäck, B. 2020. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 759–773. Barcelona (online): International Committee for Computational Linguistics.
- Shen, Y.; Tan, S.; Sordoni, A.; and Courville, A. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019*. New Orleans, LA, USA: OpenReview.net.
- Shrivastava, A.; Pupale, R.; and Singh, P. 2021. Enhancing aggression detection using gpt-2 based data balancing technique. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1345–1350.
- Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In Bengio, Y., and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; and Dai, J. 2020. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*. Addis Ababa, Ethiopia: OpenReview.net.
- Suryawanshi, S.; Chakravarthi, B. R.; Arcan, M.; and Buitelaar, P. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, 32–41. European Language Resources Association (ELRA).
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.
- Tabinda Kokab, S.; Asghar, S.; and Naz, S. 2022. Transformer-based deep learning models for the sentiment analysis of social media data. *Array* 14:100157.
- Tang, E. K.; Suganthan, P. N.; and Yao, X. 2006. An analysis of diversity measures. *Machine Learning* 65(1):247–271.

- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. Llama: Open and efficient foundation language models. *ArXiv* abs/2302.13971.
- van Aken, B.; Winter, B.; Löser, A.; and Gers, F. A. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of CIKM*, 1823–1832.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is all you need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Wang, Y., and Markov, I. 2024. CLTL at DIMEMEX Shared Task: Fine-Grained Detection of Hate Speech in Memes. In Jiménez-Zafra, S. M.; Chiruzzo, L.; Rangel, F.; Corrêa, U. B.; Jover, A. B.; Gómez-Adorno, H.; Barba, J. Á. G.; Farías, D. I. H.; Ráez, A. M.; Moral, P.; Abellán, C. R.; Rodríguez, M. E. V.; Taulé, M.; and Valencia-García, R., eds., *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), Valladolid, Spain, September, 2024*, CEUR Workshop Proceedings. Valladolid, Spain: CEUR-WS.org.
- Wang, B.; Ding, Y.; Liu, S.; and Zhou, X. 2019. Ynu_wb at HASOC 2019: Ordered neurons LSTM with attention for identifying hate speech and offensive language. In Mehta, P.; Rosso, P.; Majumder, P.; and Mitra, M., eds., *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, 191–198. CEUR-WS.org.
- Wawer, A., and Sarzyńska-Wawer, J. 2022. Detecting deceptive utterances using deep pre-trained neural networks. *Applied Sciences* 12(12).
- Wenjie, Y., and Arkaitz, Z. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*.
- Xiao, T.; Fan, Q.; Gutfreund, D.; Monfort, M.; Oliva, A.; and Zhou, B. 2019.

- Reasoning about human-object interactions through dual attention networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 3918–3927.
- Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*, volume 37, 2048–2057. PMLR.
- Yan, C.; Hao, Y.; Li, L.; Yin, J.; Liu, A.; Mao, Z.; Chen, Z.; and Gao, X. 2022. Task-adaptive attention for image captioning. *IEEE Transactions on Circuits and Systems for Video Technology* 32(1):43–51.
- Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489. Association for Computational Linguistics.
- Yang, C.; Zhu, F.; Liu, Y.; Han, J.; and Hu, S. 2024. Uncertainty-aware cross-modal alignment for hate speech detection. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 16973–16983. Torino, Italia: ELRA and ICCL.
- Ye, L.; Rochan, M.; Liu, Z.; and Wang, Y. 2019. Cross-modal self-attention network for referring image segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10494–10503.
- Yigezu, M. G.; Kolesnikova, O.; Sidorov, G.; and Gelbukh, A. F. 2023. Transformer-based hate speech detection for multi-class and multi-label classification. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF), co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN)*, 9.
- Yu, Y.; Si, X.; Hu, C.; and Zhang, J. 2019a. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation* 31(7):1235–1270.
- Yu, Z.; Yu, J.; Cui, Y.; Tao, D.; and Tian, Q. 2019b. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Zampieri, M.; Nakov, P.; Rosenthal, S.; Atanasova, P.; Karadzhov, G.; Mubarak, H.; Derczynski, L.; Pitenis, Z.; and Coltekin, C. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.
- Zeerak, W., and Dirk, H. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, 88–93. Association for Computational Linguistics.
- Zhang, J., and Wang, Y. 2022. SRCB at SemEval-2022 task 5: Pretraining based image to text late sequential fusion system for multimodal misogynous meme identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 585–596. Seattle, United States: Association for Computational Linguistics.
- Zhang, H.; Goodfellow, I. J.; Metaxas, D. N.; and Odena, A. 2019a. Self-attention generative adversarial networks. In *Proc. of ICML*, volume 97, 7354–7363. PMLR.
- Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019b. ERNIE: Enhanced language representation with informative entities. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1441–1451. Florence, Italy: Association for Computational Linguistics.
- Zhang, E. Y.; Cheok, A. D.; Pan, Z.; Cai, J.; and Yan, Y. 2023. From turing to transformers: A comprehensive review and tutorial on the evolution and applications of generative transformer models. *Sci* 5(4).
- Zhang, Z.; Robinson, D.; and Tepper, J. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings*, 745–760. Berlin, Heidelberg: Springer-Verlag.
- Zhao, S., and Zhang, Z. 2018. Attention-via-attention neural machine translation. *Thirty-Second AAAI Conference on Artificial Intelligence* 32(1).

- Zhou, Y.; Chen, Z.; and Yang, H. 2021. Multimodal learning for hateful memes detection. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. Los Alamitos, CA, USA: IEEE Computer Society.
- Zhou, Y. 2020. A review of text classification based on deep learning. In *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis, ICGDA 2020*, 132–136. New York, NY, USA: Association for Computing Machinery.
- Zhu, R. 2020. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *ArXiv* abs/2012.08290:10.