UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GUSTAVO ACAUAN LORENTZ

Analysis of Multimodal Methods for Automatic Misogyny Identification

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Science

Advisor: Profa. Dra. Viviane P. Moreira

Porto Alegre
May 2022

ABSTRACT

Social media has significantly impacted our lives by changing how we work, study, relax, inform ourselves, and communicate. Social media and the Web create a false sense of security. With apparent anonymity, users participate in the sharing and making of fake news and hateful speech, which explains why misogyny (*i.e.,* hatred targeted at women), is prevalent and (increasingly) abundant on the internet. Taking that into consideration, academic researchers and social media platforms dedicate considerable efforts to developing automatic hate speech identification methods. Because hateful speech and its branches are complex and involve matters of cultural background and societal norms, the question remains: is it possible to automatically identify and classify hateful content, and, more specifically, multimodal misogynous content, that is, content based not only on textual inputs but also on visual inputs? Previous research on general hateful content has shown that yes, it is possible to identify multimodal hateful content. Deep learning models achieve better-than-random performance. However, the performances fall short of human accuracy. It is known, however, that the knowledge obtained by algorithms about specific hate targets does not extend to other hate targets. It is not clear whether the same performances can be obtained when analyzing a type of hate with specific targets, namely women. Therefore, the goal of this work is to determine whether these same models can also automatically identify misogyny. To discover that, we trained models using the dataset from SemEval2022 Task 5 Multimedia Automatic Misogyny Identification (MAMI), which has the goal of improving the quality of existing methods for misogyny identification, many of which require dedicated personnel. The training dataset contains 10,000 memes, with both visual and textual information. The Modular Multimodal Framework (MMF), developed by Facebook A.I. Research was used for the training process. The evaluation consisted of obtaining the Macro-F1 measure for all models on their predictions for the test set, which contained 1,000 memes. We experimented with seven models: ViLBERT and VisualBERT both uni and multimodally pretrained, MMBT, and two unimodal models, Image-Grid (ResNet152) and BERT. The results show that all multimodal models achieved Macro-F1 scores above 0.649. While Image-Grid performed the worst, with a score of 0.59. ViLBERT was the best performer with a score of 0.698 and ranked 32^{nd} on MAMI's leaderboard. These results show that yes, these models are capable of identifying multimodal misogynous content, although still falling short of human accuracy. In conclusion, our work helps establish that multimodal automatic identification of misog-

ynous content is plausible but still has a lot to improve. We confirm the findings from previous research on general hateful content and show that the performance obtained in that dataset is also achievable on MAMI's datasets, which focus on women as hate speech targets.

Keywords: Multimodal. Misogyny. Classification. Deep learning. ViLBERT. VisualBERT. MMBT. ResNet152. BERT.

Avaliação de Modelos na Identificação Multimodal e Automática de Misoginia

RESUMO

As redes sociais impactaram significamente as nossas vidas, ao mudar a forma como trabalhamos, estudamos, relaxamos, nos informamos e comunicamos. As redes sociais e a internet criam uma falsa sensação de segurança. Com o aparente anonimato, os usuários participam no compartilhamento e na criação de fake news e discurso de ódio, o que explica porquê a misoginia – ódio destinado a mulheres –, cuja identificação é o principal foco desse trabalho, é prevalente e (crescentemente) abundante na internet. Considerando esse cenário, pesquisadores da academia e plataformas de rede sociais dedicam esforços consideráveis para desenvolver métodos automáticos de identificação de discurso de ódio. Por discurso de ódio e suas ramificações serem conceitos complexos que envolvem questões culturais e de normas da sociedade, pode-se perguntar: é possível identificar conteúdo de ódio automaticamente, e, mais especificamente para esse trabalho, conteúdo multimodal misógino, isso é, conteúdo baseado não somente em textos como entrada, mas também em imagens como entrada? Pesquisas existentes em conteúdo de ódio geral mostram que sim, é possível identificar conteúdo de ódio multimodal. Modelos de Deep Learning alcançam performances melhores do que aleatórias, apesar de ficarem para trás comparados a acurácia de humanos. Sabe-se, entretanto, que o conhecimento obtido por algoritmos sobre alvos específicos de discurso de ódio não se extende para outros alvos de discurso de ódio. Não é claro se as mesmas performances podem ser obtidas ao se analizar um tipo de discurso de ódio com alvos específicos, mulheres, nesse caso. Portanto, o objetivo desse trabalho é determinar se os mesmos modelos também podem identificar misoginia automaticamente. Para descobrir isso, os modelos deste trabalho foram treinados usando o dataset da *Task* 5 do SemEval2022, *Multimedia Automatic Misogyny Identification* (MAMI), ou Identificação Multimídia e Automática de Misoginia, a qual tem como objetivo melhorar a qualidade de métodos existentes para identificação de misoginia, muitos dos quais requerem funcionários dedicados para essa tarefa. O dataset de treinamento contém 10.000 memes, com informação visual e textual. O *Modular Multimodal Framework* (MMF), desenvolvido pelo *Facebook A.I Research* foi utilizado para o processo de treinamento. A avaliação consistiu em obter os valores de Macro-F1 de todos modelos após eles classificarem o dataset de teste, que contém 1.000 memes. Foram feitos experimentos com sete modelos existentes: ViLBERT e VisualBERT, ambos pré-treinados uni e

multimodalmente, MMBT, e dois modelos unimodais, Image-Grid (ResNet152) e BERT. Os resultados mostram que todos modelos multimodais alcançam *scores* de Macro-F1 acima de 0,649. Enquanto Image-Grid teve a pior performance, com um *score* de 0,59. ViLBERT foi o modelo com melhor performance, com um *score* de 0.698 e alcançou a posição 32 no ranking da competição MAMI. Esses resultados mostram que sim, esses modelos são capazes de executar identificação multimodal de conteúdo misógino, apesar de ainda ficarem abaixo de performances humanas. Em conclusão, esse trabalho ajuda a estabelecer que a identificação multimodal automática de conteúdo misógino é plausível porém ainda possui muito o que melhorar. Nós confirmamos os achados de pesquisas existentes sobre conteúdo de ódio geral e mostramos que a performance obtida naquele dataset também é alcançável no dataset da competição MAMI, que foca em mulheres como alvos de discurso de ódio.

Palavras-chave: Multimodal, Misoginia, Classificação, Deep Learning, ViLBERT, VisualBERT, MMBT, ResNet152, BERT.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
AGI	Artificial General Intelligence
BERT	Bidirectional Encoder Representation from Transformers
CNN	Convolutional Neural Network
CV	Computer Vision
LM	Language Model
LSTM	Long Short-Term Memory
MAMI	Multimedia Automatic Misogyny Identification
MLM	Masked Language Modeling
NLP	Natural Language Processing
POS	Part of speech
RNN	Recurrent Neural Network

CONTENTS

1 INTRODUCTION

***Sensitive content warning**: this work contains images that might be disturbing or triggering to readers. Some memes depict sexual violence and misogyny. Therefore, this work should be read only by a mature audience and reader discretion is advised.*

Before the rise of the Web, content was created in a centralized way. TV channels, the movie industry, newspapers, and magazines produced most of what people consumed as entertainment. That has changed in the last decade, with the creation of the first social media platforms, which allowed people to access forums on the most varied topics, participate in synchronous group chats, read up-to-date news from around the world, and contact others through instantaneous messaging. Social media empowers users to create new content and share and consume content created by other users, not only by the mainstream media.

Social media has significantly impacted our lives by changing how we work, study, relax, inform ourselves, and communicate. Among the positive effects, one can cite economic growth and societal change through campaigns that raise people's awareness of racism and sexual harassment through the Black Lives Matter and #MeToo movements, for example.

Nevertheless, when social media empowers social movements, it also strengthens people who oppose them. Outside the Web, certain behaviors are condemned by society, and so, through peer pressure, people are incentivized to behave according to the norm. However, social media and the Web create a false sense of security. With apparent anonymity, users participate in the sharing and making of fake news and hateful speech, which explains why misogyny, the primary focus of this work, is abundant on the internet.

It has been shown by Shifman (2013) that memes can work as persuasive tools to transmit ideas hidden behind humor. This fact is worrying, given that misogyny is not only present on social media but also increasingly so, as confirmed by Farrell et al. (2019). Furthermore, the findings from Drakett et al. (2018), state that misogynous memes have a role at "reproducing damaging constructions of women and femininity, and deriving humor from issues such as sexual assault or domestic violence". These facts amount to a worrying context in which sexist ways of thinking might influence users.

The platforms that contribute to the sharing of hateful content dedicate a considerable amount of human effort to detecting, analyzing, and eventually removing these

contents. The task is demanding due to the nature of the posts, which are frequently not straightforward – they often contain irony and slang. Additionally, the textual information needed to automatically classify a post as misogynistic might be part of an image in the form of a meme. That prevents sexist posts from being immediately detected by algorithms that rely solely on textual input.

Keeping the platforms free from misogyny is of interest to social media companies, considering that this type of content leads to declines in user experience and satisfaction, which might encourage users to leave the platforms. Therefore, reliable and automatic misogynistic content identification methods are desirable.

Intending to compare solutions to automatic misogyny identification, we experimented with multimodal models - models that use more than one source of information, *e.g.* texts and images - applied to SemEval2022 Task 5 – Multimedia Automatic Misogyny Identification (MAMI) (FERSINI et al., 2022). The work presented here improves upon the paper submitted for SemEval2022 Task 5, which was composed of the analysis of five multimodal models and was primarily influenced by The Hateful Memes (KIELA et al., 2021). We extend our work with analyses of two new unimodal models, Image-Grid (ResNet-152) (HE et al., 2016) for visual modality and BERT (DEVLIN et al., 2018) for textual modality, which provide a good basis for comparison for all other previously analyzed models.

MAMI consists of a classification problem using misogyny and sub-types of violence against women, such as shaming, violence, objectification, and stereotype. The goal is to classify memes based on both textual and visual information. A good example of a meme present in the dataset is Figure 4.4a, which shows how both modalities are often necessary for the correct prediction. This paper describes the training and usage of uni and multimodal models applied to Subtask A, a binary classification problem using misogyny. We explain the differences between the models – their distinctive features in architecture and input processing – and compare their performances in light of variances in pretraining and model modality.

Our goal is to answer questions about the performances of the models in terms of classification quality. We analyze and compare the models in search of similarities and disparities that clarify decisive factors for achieving good scores for this task. The questions we want to answer are, specifically:

- What are the best and worst models?
- Do multimodally pre-trained models perform better?

- Do multimodal models perform better?
- Can combining classifiers improve classification performance?
- How correlated are the models?
- Is there any pattern in memes that were erroneously classified?

Analyzing results shows that some aspects have a significant impact on performance. One important characteristic is the modality since most multimodal models outperform unimodal models. This finding is expected because of the inputs' nature. Texts and images often contradict each other, as explained by what is known as the incongruity theory, which states that the humor is created – in the case of memes – as the image introduces a situation and the caption turns out to be something that violates our expectations (MORREALL, 2020). Therefore, both textual and visual information are needed to classify memes correctly.

Another important distinction is the use of image features for training, with our worst models being the ones that do not make use of such features. Further analysis showed that combining the models in a majority-voting style did not improve performance, and neither did averaging their outputs to form a single prediction.

The scores and rankings described here are from the competition leaderboards, except for the new unimodal models, which are trained and scored utilizing the same datasets and metrics but are not included in the official competition and leaderboards. Among the models, the one which achieved the highest score was ViLBERT, reaching the 32^{nd} position on the leaderboard (out of 83 participants), with a Macro-F1 score of 0.698. The worst performer was Image-Grid, with a score of 0.599.

In conclusion, our work helps establish that multimodal automatic identification of hateful content is possible but still has a lot to improve. Knowledge obtained by detection models on hateful speech does not transfer to other hateful speech targets, as shown by Nozza (2021). We confirm findings by The Hateful Memes, which included hateful content in general. We show that the performance obtained in that dataset is also achievable on MAMI's datasets, which focus on women as hate speech targets.

This work will be divided into six chapters: Introduction, Background, Related Work, Materials and Methods, Results, and Conclusion. Chapter 2 presents the proper fundamental knowledge required to understand this work. It ranges from broad concepts such as AI and NLP to more specific mechanisms and techniques like the Attention mechanism and Transformers. Chapter 3 presents relevant works in multimodal models and hateful content identification and the impacts they have in this work is also be described.

Chapter 4 details the models we used and their distinctive factors and describes the training configurations and metrics. Chapter 5 is where analyses of results and comparisons between models are presented. The questions asked in the Introduction are answered and the results are scrutinized in search of patterns in mistakes and commonalities between different models. Chapter 6 concludes this work and presents opportunities for future research. An Appendix describes the models configurations in full.

2 BACKGROUND

In this chapter, we explain how algorithms are able to understand, identify, and predict subtle concepts such as misogyny and hate.

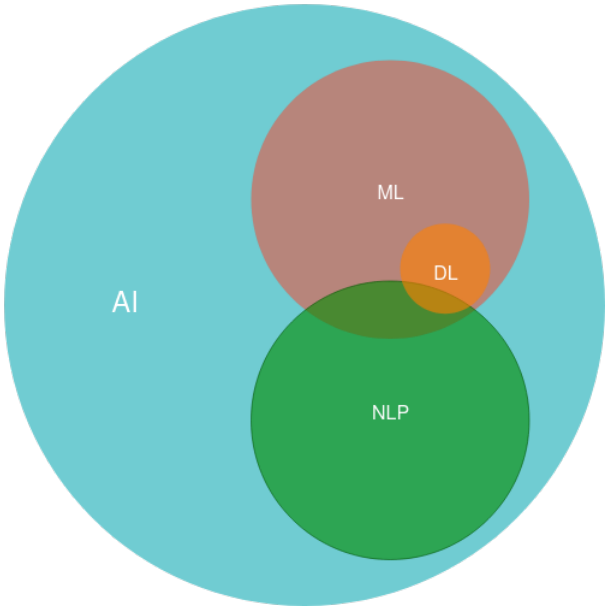
2.1 Artificial Intelligence

Artificial Intelligence (AI) is defined by Poole, Mackworth and Goebel (1997) as *the study of "intelligent agents": any system that perceives its environment and takes actions that maximize its chance of achieving its goals*. Initially, the ultimate goal of AI was to give machines the ability to think like humans. That means they must be able to learn from experience, understand written and spoken language, make intelligent decisions, and manipulate objects in the physical world. Specific skills are required to achieve this goal, such as reason and problem-solving, knowledge representation, planning, and learning. Because the currently popular methods – Machine Learning and Deep Learning – require specialization for each task, AI has branched into numerous areas of study. Due to this subdivision, researchers have now defined the term Artificial General Intelligence (AGI) as the area of AI that still considers its goal to give machines unbounded human-like intelligence. Machine Learning (ML) and Deep Learning (DL) are part of statistical AI, which uses information from correlations between data to draw a conclusion regarding the desired subject. Each area of study uses these methods for their purposes. Natural Language Processing (NLP) and Computer Vision (CV) are the main subjects of this work. Figure 2.1 helps to illustrate that AI is the general concept enveloping methods such as ML, DL, and areas of study like NLP and CV.

2.2 Machine Learning

The concept of ML stands for algorithms in which the performance of the machine improves with experience concerning some task, as defined by Jordan and Mitchell (2015). To achieve this, the algorithms must learn from training data to discover insights and patterns, with no explicit programming guiding them towards these insights and patterns (BISHOP, 2006). A complicating factor of ML algorithms is their reliance on input quality. Experts must determine the adequate features to feed to the algorithms. This

Figure 2.1 – Diagram showing the relationships among AI, ML, DL, and NLP.



Source: the author

process takes considerable time and effort.

ML techniques are usually divided into three categories, they are:

- Supervised learning: the algorithm receives example inputs and outputs and must learn a general mapping from one to the other. It does that by predicting the output for a given input and then adjusting itself based on how good the guess taken was.
- Unsupervised learning: only inputs are given to the algorithm, and its goal is to discover hidden patterns in the data. It does that by grouping or clustering data, for example. There is no feedback on how good the grouping is. Therefore the algorithms act based on commonalities (or the lack thereof) between data points.
- Reinforcement learning: the algorithm interacts with an environment, taking actions it believes to be optimized and receiving feedback based on their consequences. It does this continuously, learning from the feedback of each action.

2.3 Deep Learning

Deep learning is an ML technique that attempts to imitate the human brain using concepts such as neurons and neural networks. A neural network is created by organizing neurons (basic processing units) in multiple layers. These networks benefit significantly from increased volumes of data, contrasting with other ML algorithms, that usually

plateau at a certain performance level no matter how much additional data is used.

One crucial distinction from other ML techniques is that DL eliminates some of the data processing needed. For example, DL algorithms can process unstructured data, such as images and text. Furthermore, they can automate feature extraction, making it unnecessary for researchers to manually decide on essential features, a process that might be subject to human error and bias.

2.4 Natural Language Processing

The primary concern of NLP is to address how computers can effectively process and analyze natural language, that is, any kind of language that has evolved naturally in humans. The goal is to have a machine that can comprehend natural language at such a level that it will be able to formulate sentences, answer questions, hold conversations, interpret texts, metaphors, irony, humor, slang, and other complex linguistic concepts. In short, the goal of NLP is to create machines that can adequately communicate using written or spoken human languages. A machine must be able to understand natural language and generate it to achieve that. For that reason, NLP is often subdivided into two subareas, Natural Language Understanding and Natural Language Generation.

The history of NLP can be divided into three main phases: Symbolic, Statistical, and Neural NLP.

Alan Turing defined the now-famous Turing test (TURING, 1950) in the 1950s as a decisive criterion for intelligence. To succeed in this test, a machine must be able to completely fool a human into believing that the machine is, in fact, not a machine. It must do that through a conversation. We see then that NLP has been of interest to the scientific community since the beginnings of computer science.

The symbolic phase consists mainly of methods that can be described as different variations of the Chinese Room Experiment (SEARLE, 1980). In this famous experiment, John Searle describes a hypothetical situation where a machine has been programmed to behave as if it perfectly understands Chinese. It takes Chinese characters as input and produces Chinese characters as output. It can convincingly pass the Turing test when utilized by a native Chinese speaker. Searle's question is the following: *does this machine really* understand *Chinese? Or is it simply simulating?* His argument to answer this question is another scenario. Assuming now that, instead of a machine, a human had a book detailing what to answer based on the Chinese inputs, essentially functioning the same way as the

machine previously described. Searle says that we would *not* consider this human a fluent speaker of Mandarin and, therefore, should judge the machine accordingly.

The Georgetown Experiment (HUTCHINS; DOSTERT; GARVIN, 1955), the first notable algorithm of the symbolic phase back in the 1950s, consisted of a direct mapping between phrases in Russian to their versions translated into English. Initial hopes for the future of NLP were highly optimistic but turned out to be wrong. NLP crawled with slow progress to the 1980s, functioning with algorithms slowly increasing in complexity, using ontologies, semantics, and morphology, but still maintaining the idea of utilizing mappings, albeit not direct mappings like the ones in the Georgetown Experiment.

The Statistical phase began in the 1990s, with increased computational power and the introduction of ML methods for language processing. IBM was a leader in research in machine translation, using governmental corpora of considerable sizes. Some algorithms used during the statistical phase are hidden Markov models and decision trees. Other tasks faced the need to assemble specialized datasets. Nowadays, this problem is still a significant obstacle to NLP researchers, even though the Web and social media have greatly facilitated the data collecting process. This increase in raw unannotated data led researchers to focus on unsupervised methods, which begins the shift to the next phase in NLP history.

The Neural phase began in the late 2000s and is still the current trend in NLP. In 2009, the first neural language model was proposed by Bengio et al. (2003). It consists of a feed-forward neural network, *i.e.,* data can only move in one direction, from input to hidden layers to output layer. Its task was language modeling, in which the model has to predict the next word in a sentence based on the previous words. Language modeling is a crucial task in NLP, seeing that all state-of-the-art models use a form of LM during pretraining.

The use of word-embeddings is an essential development of the Neural phase. Word-embeddings represent words for analysis and processing in the form of vectors that encode word meaning. With word-embeddings, vectors close to each other are expected to share some semantic relationship. They might be synonyms or terms that often appear together, such as countries and capitals. In order to build these semantic relationships, the word-embeddings training process can occur with two different goals. One is Continuous Bag-Of-Words (CBOW) which consists of predicting a target word given the surrounding words. The other is Skip-Gram, the opposite of CBOW, which predicts the surrounding words given an input word. Two popular implementations are word2vec (MIKOLOV

et al., 2013a; MIKOLOV et al., 2013b) and GloVe (PENNINGTON; SOCHER; MANNING, 2014).

Recurrent Neural Networks (RNNs) were proposed in 1986 by Jordan (1997) and are a form of artificial neural network in which layers can have connections with previous layers, essentially creating a feedback loop that allows RNNs to identify spatial positions of words in phrases. Long short-term memory (LSTM) networks (HOCHREITER; SCHMIDHUBER, 1997) are an extension of RNNs that address the Vanishing Gradient Problem, as identified by Hochreiter (1991). The training process of RNNs involves backpropagation (BP), which computes the gradient of the loss function in relation to each weight using the chain rule, doing so for each layer starting from the last. The problem is that the gradient can sometimes explode towards infinity (exploding gradient problem) or tend to zero (vanishing gradient problem) due to the finite precision of computers. This vanishing may sometimes lead to total stagnation of the training process. To address that, LSTMs allow gradients to also flow *unchanged* if desired. That enables LSTMs to select which pieces of information to forget and which to remember.

2.5 Encoder/Decoder

Sequence-to-sequence models were proposed by Sutskever, Vinyals and Le (2014). These models worked by using the encoder and decoder concepts. An LSTM network receives a sentence in natural language as input and produces vector representations. This process is an *encoding*, hence the name Encoder. Another LSTM network, called the Decoder, receives as input the vector representations generated by the Encoder and then tries to predict the output sentence by *decoding* the vector representations into words.

2.6 Attention Mechanism

Another critical development in NLP has been the Attention Mechanism (BAHDANAU; CHO; BENGIO, 2014). It is responsible for addressing the problem of long sentences in RNNs and LSTMs, making it hard for the models to remember all the contextual information necessary to predict words correctly. The Attention Mechanism works by calculating *which terms are important to be considered by the model*, essentially providing the model the crucial context for the prediction.

2.7 Transformer

The Transformer architecture as seen in Figure 2.2 was proposed by Vaswani et al. (2017) in 2017 and quickly achieved state-of-the-art results, replacing LSTMs as the go-to solution to most NLP problems. Transformers build on top of the advances made by the encoder/decoder concepts, attention mechanism, and word-embeddings.

The model uses attention to obtain contextually accurate representations for words. It does that by improving on usual attention mechanisms as explained in Section 2.6, and relying on self-attention in three different parts of the model, namely in the encoder, decoder, and encoder-decoder.

2.7.1 Self-Attention

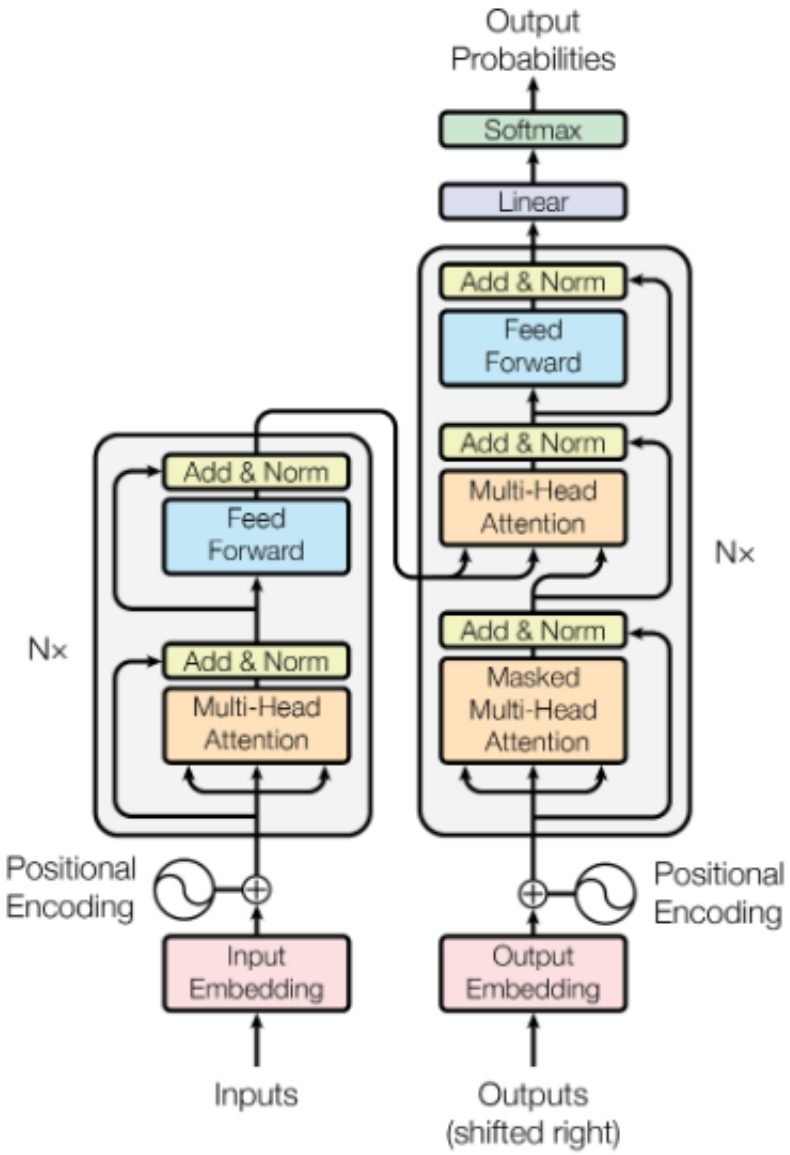
Figure 2.3 illustrates what self-attention is. Officially called *scaled dot-product attention units*, what they do is calculate weighted embeddings for all tokens in a sequence. These weighted embeddings contain contextual information for all tokens.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

To perform the attention calculation seen in Equation 2.1, the attention blocks receive three matrices as input, the query (Q), key (K), and value (V) matrices. Each token in the input sequence is represented as a vector, that is first summed with positional encoding information so that tokens really far from each other share less impact. The result is then projected using the three matrices, resulting in three different vectors per token. Then the dot product of the query and key matrices is taken. The dot product is helpful because, given two vectors, its result can inform us how similar the vectors are. Their similarity is larger the closer the result is to one. This calculation is done for each token against all others in the sentence, which gives us how strong the relationships between tokens are. To avoid extremely low values in gradients in the next step, these results are all divided by the square root of the length of the key vector. Softmax is then applied to the results to non-linearly normalize them, which in turn strengthens the previously high scores, and weakens previously low scores, emphasizing or neglecting the relationships between tokens. These normalized values are then used to multiply the V matrix, adequately reducing and increasing scores for tokens based on their contextual

relevancy. The output is then a weighted sum of all value matrices of the tokens.

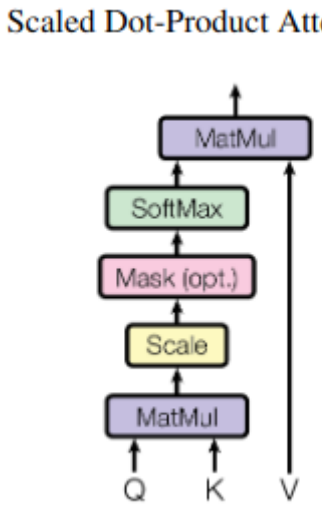
Figure 2.2 – The Transformer Architecture



Source: Vaswani et al. (2017)

This process happens in three distinct places. In the encoder self-attention, the input sequence pays attention to itself and is used as all parameters, V, K, and Q. In a similar fashion, in the decoder, namely at *Masked Multi-Head Attention* in Figure 2.2, the output pays attention to itself and is also passed as V, K and Q. At the encoder-decoder self-attention block, however, there are differences. At this stage, the target sequence pays attention to the input sequence, given that the output of the decoder is used as Q, shown in Figure 2.2 as the arrow coming from *Masked Multi-Head Attention* into *Multi-Head*

Figure 2.3 – Scaled Dot-Product Attention Unit



Source: Vaswani et al. (2017)

Attention, and the output of the encoder is passed as V and K, shown as the two other arrows entering *Multi-Head Attention*.

As the name *Multi-Head Attention* might suggest, the calculation of attention is not performed only once. In fact, there are multiple attention heads, each using different values of dimensionality, therefore impacting the calculations in Equation 2.1 and yielding different results. All results are concatenated and then once again projected thus obtaining the final output.

2.8 Bidirectional Encoder Representations from Transformers (BERT)

BERT (DEVLIN et al., 2018) was developed by Google and released in 2018. Since its publication, the model surpassed state-of-the-art performances in numerous tasks, including General Language Understanding and Question Answering. The model's performance reached such a high level that by late 2020 almost all Google searches in English were being processed by BERT. Part of its success comes from the pretraining the model is subjected to. BERTs pretraining involves the task of Language Modeling, in which the goal is to predict the word at a given position in a sentence, and Next Sentence Prediction, in which the model needs to indicate whether a sentence follows another. By doing these tasks, the model is able to learn contextualized embeddings for words. That is, words now have more than only one vector representation, unlike with word-embeddings

like word2vec (MIKOLOV et al., 2013a) and (MIKOLOV et al., 2013b) and GloVe (PENNINGTON; SOCHER; MANNING, 2014).

BERTs architecture uses the Transformer model (VASWANI et al., 2017) by stacking copies of them. Two versions were made available with publication, $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$. The difference is that $\text{BERT}_{\text{base}}$ stacks 12 Transformers with hidden layers of size 768 and 12 attention-heads, while $\text{BERT}_{\text{large}}$ stacks 24 Transformers with hidden layers of size 1024 and 16 attention-heads.

2.9 Computer Vision

Computer vision is an interdisciplinary field that seeks to give machines the same visual comprehension skills as humans have. It works with techniques to gather, process, and understand complex data from images and videos.

Instead of using eyes, brains, and intuition, CV uses cameras, data, and algorithms. Many of the concepts explained in previous sections are also used in this field. Although less precise than humans, machines can operate at a much larger speed, processing images in batches. Their speed and specification to tasks enabled their high use in industry today, with techniques such as object detection, recognition, and tracking in videos and still images. CV has applications in automation, surveillance, medicine, and many other areas.

Just like for AI, the initial hopes for CV were high. In the 1960s, MIT research assistant Seymour Papert assembled a team of students to attempt to solve a CV problem consisting of the description of regions likely containing background or objects. The established time for this project was of a single summer. The project became known as The Summer Vision Project (PAPERT, 1966). The short amount of time reserved for this task shows how optimistic researchers were for the future of CV.

In the 1970s and 1980s, the theoretical and practical foundations of today's techniques were laid down, such as edge identification and extraction, polyhedral modeling, optical flow (LUCAS; KANADE, 1981), and motion estimation. In 1974, Ray Kurzweil created a company intending to develop further omni-font OCR, which could identify text in any other font. The area deepened rigorous mathematical analysis, which allowed shape inference based on shading, texture, and focus. By the 1990s, areas like face recognition had their first developments, such as the use of eigenvectors in a technique called Eigenface (TURK; PENTLAND, 1991). Other advancements were made in various fields, including rendering and image morphing.

In recent times, CV has increased in popularity. Because of the surge in computer power and available data, ML and DL algorithms revolutionized the field and introduced it to a new era. Nowadays, feature-based methods are the current trend.

2.10 Features

In CV, features are pieces of information about images. They describe the contents of the image in a structured way, indicating the presence or absence of specific shapes in the image regions. For example, features might inform if and where there are edges, blobs, corners, or ridges in the image.

2.11 Convolutional Neural Networks

Convolutional Neural Networks are extensively used in CV, because they excel at image processing, particularly due to their ability to identify objects. CNNs receive input images and assign weights to objects in the image to make distinctions from one to another. They do that by using Convolutional Layers, which work by using filters that scan the whole image for features. CNNs use the hierarchical pattern in data and search for features with growing complexity using their filters.

2.12 Rectified Linear Unit (ReLU)

ReLu is an activation function defined as a function that returns the positive part of its input. That is, when the input values are below zero, the activation is zero. But when the inputs are positive values, the activation is the input. It can be easily described as in Equation 2.2. When used, ReLU introduces non-linearity into the network.

In 2011, it was found that this activation function is better for deeper networks (GLOROT; BORDES; BENGIO, 2010), which has made ReLU the usual choice for deep neural networks in CV and NLP.

$$f(x) = \max(0, x) \tag{2.2}$$

2.13 Residual Neural Network (ResNet)

Residual Neural Networks (HE et al., 2016) are Artificial Neural Networks that use *skip connections* in their architectures. As seen in Figure 3.1, some layers receive the summed output of previous layers as input. The skipped layers often contain ReLU between them – introducing non-linearity in learning – and batch normalization (IOFFE; SZEGEDY, 2015). This technique helps avoid the vanishing gradient problem and the degradation problem, which happens when adding more layers to a network *increases* training error.

3 RELATED WORK

Works relevant to the development of this monograph and the recent evolution in multimodal models will be briefly explained in this chapter, specifying their impact and influence on this work.

3.1 The Hateful Memes Challenge

The Hateful Memes Challenge (KIELA et al., 2021) is similar to MAMI (FERSINI et al., 2022) since both address hateful multimodal contents. Participants in the Hateful Memes Challenge received a dataset of memes with visual as well as textual inputs and had to predict whether the memes were hateful. MMF (SINGH et al., 2020) is a multimodal framework from Facebook AI Research and it implements state-of-the-art visual and language models, such as VisualBERT (LI et al., 2019), ViLBERT (LU et al., 2019), MMBT (KIELA et al., 2019) M4C (HU et al., 2020), and Pythia (JIANG et al., 2018), among others. MMF provides code and model implementations for The Hateful Memes Challenge. Their work served as the primary inspiration for our experiments, in which we apply many of the same models to the Multimedia Automatic Misogyny Identification (MAMI) dataset.

3.2 Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution

Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution (ZHU, 2020) was the winning submission to the Hateful Memes Challenge. It used concepts like object, web entity, and human face detection, as well as feature extraction. This information was fed to three pretrained multimodal models, which had their predictions averaged, VL-BERT (SU et al., 2019), UNITER-ITM (CHEN et al., 2019), VILLA-ITM (GAN et al., 2020), and ERNIE-Vil (YU et al., 2020). Key factors for the top performance were linking text and image regions – because a caption saying “sandwich maker” on top of an image of a woman has a totally different meaning from the same caption over objects –, and racism detection, given that approximately half the memes in the Hateful Memes Challenge could be labeled as racist.

This submission achieved an AUROC score of 0.8450.

3.3 Vilio: State-of-the-art Visio-Linguistic Models applied to Hateful Memes

Vilio: State-of-the-art Visio-Linguistic Models applied to Hateful Memes (MUENNIGHOFF, 2020) was the submission that achieved second place in the Hateful Memes Challenge. Like some of the models used in this work, it utilized image features. However, Vilio uses different models to extract features, including Detectron2 (WU et al., 2019) and models pretrained on VisualGenome (KRISHNA et al., 2016) with and without attributes. In addition to image features, various Image Regions are also used. The features, image regions, and the text are fed into five models in an ensemble. ERNIE-Vil, both large and small, UNITER, OSCAR (LI et al., 2020), and VisualBERT. Each model predicts multiple times, using different features each time. The predictions are averaged to create the final prediction value for each model. These values are then fed to an ensembling loop that performs simple, rank, and power averaging and finally simplex optimization to produce a final prediction. This submission achieved an AUROC score of 0.8310.

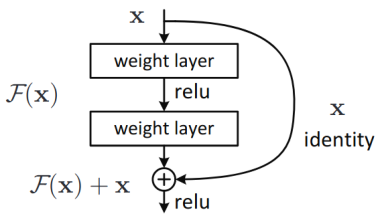
3.4 Detecting Hate Speech in Memes Using Multimodal Deep Learning Approaches: Prize-winning solution to Hateful Memes Challenge

Detecting Hate Speech in Memes Using Multimodal Deep Learning Approaches: Prize-winning solution to Hateful Memes Challenge (VELIOGLU; ROSE, 2020) is the third-placed submission and worked by first expanding the dataset by searching for similar datasets online. After some manual work, the original dataset was incremented by 328 new memes, found in the Memotion Dataset[1]. Image features were extracted using Detectron2 and were used for fine-tuning the VisualBERT model using MMF. Hyperparameter search was applied and from this, different versions of the model were created. Twenty-seven models were chosen based on their AUROC scores, and the resulting prediction consisted of a majority vote between these models. This submission achieved an AUROC score of 0.8108.

The works cited in this chapter illustrate other techniques that have been applied

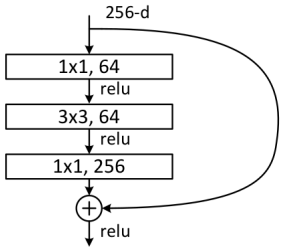
[1] <https://www.kaggle.com/datasets/williamscott701/memotion-dataset-7k>

Figure 3.1 – Residual learning: a building block



Source: He et al. (2016)

Figure 3.2 – A building block for ResNet-152



Source: He et al. (2016)

to the task of detecting hateful content in memes and that can also be applied to misogyny identification. They differ from this work mainly because each of them adds a specific nuance to the identification process. For instance, using multiple models to extract image features, expanding the dataset, ensembling, and object, human face or web entity detection. This work, however, aims at establishing a direct comparison to Kiela et al. (2021), by utilizing many of the same models and the same training process. By doing this we can assert if the models can perform in a similar fashion considering the two distinct contexts of general hate and misogyny. Nonetheless, the techniques used by the works in this chapter could be used in conjunction with the models in this work and potentially lead to an improvement in results.

4 MATERIALS AND METHODS

The models used for this work will be detailed in this chapter, describing their distinctive features, as well as the configurations used for training. Details about the dataset will also be presented, including label distribution, dataset sizes and word frequencies for the top 100 terms. A brief explanation of the important metrics used during training and evaluation will also be described.

4.1 General explanation of the models used

We used MMF (SINGH et al., 2020), described in Section 3.1, to train seven models on the MAMI dataset, which are briefly described as follows:

1. **MMBT-Grid** - a supervised multimodal bitransformer that jointly finetunes unimodally pretrained text and image encoders by projecting image embeddings to text token space.
2. **ViLBERT** - model for learning task-agnostic joint representations of image content and natural language.
3. **ViLBERT CC** - multimodally pretrained version of ViLBERT, trained on Conceptual Captions (SHARMA et al., 2018).
4. **VisualBERT** - consists of a stack of Transformer layers that implicitly align elements of an input text and regions in an associated input image with self-attention.
5. **VisualBERT COCO** - multimodally pretrained version of VisualBERT, trained on COCO (LIN et al., 2015).
6. **Image-Grid** - convolutional features extracted from ResNet-152 res-5c layer with average pooling.
7. **BERT** - a BERT model.

Two versions of ViLBERT and VisualBERT models were used. The distinction between these two versions lies not in the architecture, but rather in how they were pretrained. The multimodally pretrained versions, ViLBERT CC and VisualBERT COCO, are the official ones published by Lu et al. (2019) and Li et al. (2019), respectively. The unimodally pretrained versions are, as explained by Kiela et al. (2021), *multimodal models that were unimodally pretrained (where for example a pretrained BERT model and a*

pretrained ResNet model are combined in some way).

To further explain the difference, assume the models are to be trained using a single image. The training for the multimodal pretrained models would occur in this fashion: the model receives the image as input; both parts – textual and visual – receive that input and generate an output each; the outputs are combined into a *single* prediction; the prediction is judged and *both models are adjusted based on it*. However, for the unimodally pretrained models, it is different: the model receives the input, outputs are generated, but instead of being combined into a single prediction, each part of the model is adjusted based on *its own output, without influence from the other modality*.

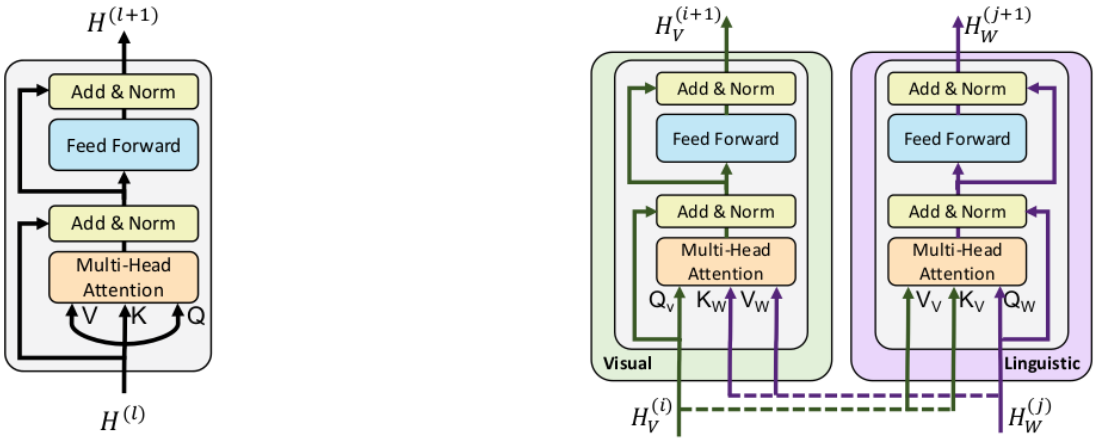
4.2 Distinctive characteristics of each model

In this section, the key characteristics that differentiate models from each other are explained. These models were chosen because they create a good basis for comparisons. There are unimodal/multimodal models, uni and multimodally pretrained models, models that use image features and models that do not. Additionally, their presence in Hateful Memes (KIELA et al., 2021) allows for direct comparisons to that paper.

4.2.1 ViLBERT and ViLBERT CC

ViLBERT (LU et al., 2019) consists of two parallel models, one that operates over visual inputs, and the other that operates over textual inputs. Both models operate similarly to BERT, *i.e.*, they are a series of transformer blocks, the difference lies in the *Co-attentional Transformer Layers* (Co-TRM) introduced by the researchers. Figure 4.1 illustrates how these Co-TRMs operate. They compute the usual Q, K, and V matrices explained in Section 2.6. However, the textual K and V are passed to the visual multi-headed attention block, and the visual K and V are passed to the textual multi-headed attention block. The rest of the transformer operations proceed normally, causing multi-modal features since each modality pays attention to the other.

Figure 4.1 – Usual Transformer Encoder block (left) and Co-TRM (right)

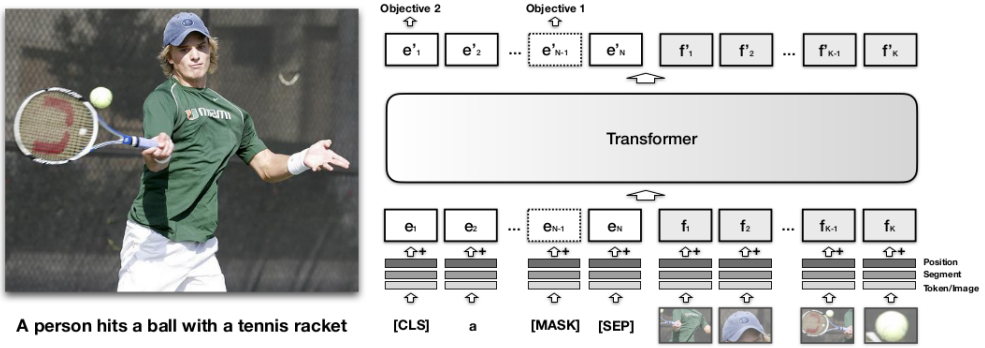


Source: Lu et al. (2019)

4.2.2 VisualBERT and VisualBERT COCO

VisualBERT extends BERT by modifying the input it processes. Making use of features extracted from Object Proposals – a set of image regions likely to contain objects – the model can capture the interaction between text and image. The model does that by treating these features as usual BERT input tokens, appending them to the textual tokens. That is, VisualBERT uses the self-attention mechanism to align textual and visual elements implicitly. The network used in this work to extract the image features is ResNet-152. A representation of input processing can be seen in Figure 4.2.

Figure 4.2 – VisualBERT has as input not only the usual text tokens BERT uses but also image features

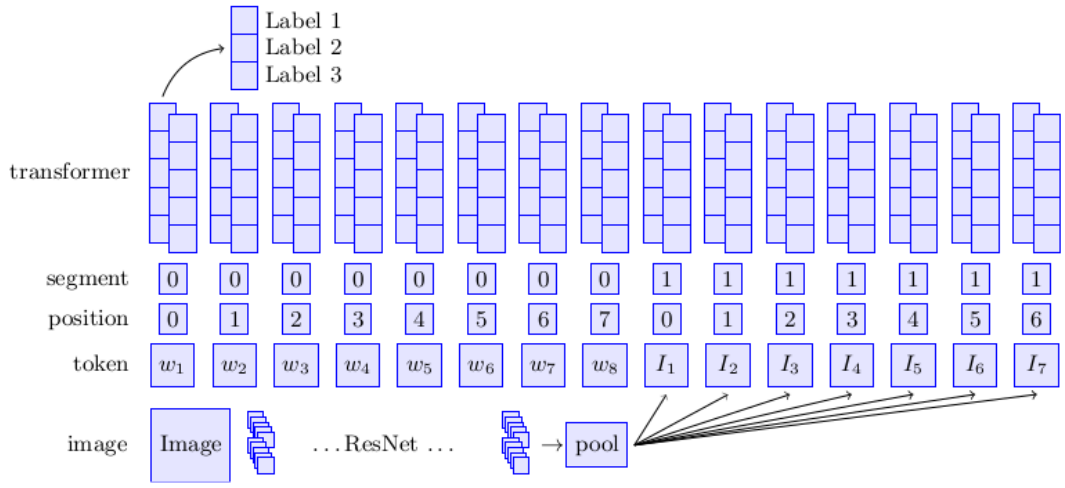


Source: Li et al. (2019)

4.2.3 MMBT-Grid

MMBT is a supervised multimodal bitransformer that jointly finetunes unimodally pretrained text and image encoders by projecting image embeddings to text token space. Its inputs are the concatenation of textual embeddings and the final activations of a ResNet after pooling – the downsampling of dimensions – and positional and segment encodings. The final activations are transformed so that they fit the dimensions of the transformers' hidden layers. ResNet-152 is, again, the network used in this work to extract features. Figure 4.3 illustrates this process.

Figure 4.3 – MMBT inputs are word embeddings, the final activations from a ResNet and positional/segment encodings.



Source: Kiela et al. (2019)

4.2.4 Image-Grid and BERT

Finally, the last two models, Image-Grid and BERT, are the models detailed in Section 2.13 and Section 2.8. That is, they are the unimodal models on top of which all other models are based.

4.3 Feature Extraction

The version utilized in this work for feature extraction and on which the visual models are based is ResNet-152. When released, this ResNet was the deepest neural network ever trained on ImageNet (DENG et al., 2009), a visual database with more than 14 million manually-annotated images for visual object identification. Despite having a depth of 152 layers – 8 times deeper than then-state-of-the-art models – the model still achieved a lower complexity while establishing new records for performance. This increase in depth was only possible due to the skip connections technique, with their actual configuration illustrated in Figure 3.2. A script by MMF that uses ResNet-152 to extract image features was used in this work to allow the training of models like VisualBERT (including VisualBERT COCO), and ViLBERT (including ViLBERT CC).

4.4 Experimental Setup

Section 4.4.1 details the configurations used for the models in this work. Section 4.4.2 consists of the dataset description, detailing input examples, label distribution, dataset size and also word-frequency. Section 4.4.3 explains the metrics used for training and evaluation.

4.4.1 Configurations

The MMF framework comes with implementations of state-of-the-art models, preconfigured with hyperparameters. In our experiments, the default configurations of the models were used. All models were trained with batch sizes of 16.

In Appendix A each model configuration is shown in full, for reproducibility and repeatability. It is also important for the configurations to be shown in full given that MMFs code, including configuration files, are available at Github[1] could change. The specific commit that was used for this work is available here[2].

Since the configuration files can point to other files, the ones that are included by others will only be shown once, and an explicit reference to them will be made every time

[1] <https://github.com/facebookresearch/mmf>

[2] <https://github.com/facebookresearch/mmf/tree/d31f8776f3bee53e7be722cb6d6c7ecf0827cc30/mmf/configs>

they are included. Each model configuration description begins by their main configuration file and expands the other included files. It is important to notice that the topmost file has the last valid configuration, *i.e.,* if file **A** includes file **B** and both define a configuration with value v, then the value used will be v_A. Furthermore, if **A** includes **B**, and **B** includes **C**, then **A** includes **C**.

4.4.2 Dataset

The dataset used to train all models was the one provided by MAMI's organization team. Table 4.1 shows each dataset's size and their respective label distributions. Training data consists of 10,000 memes, trial data has 100 memes, and the test dataset, the one used in the competition to evaluate participants' performance, has 1000 memes. All datasets are balanced in a 50/50 proportion. All memes in the datasets have their visual and textual parts, as well as their respective labels. The textual information has been extracted via OCR and has been provided by the organization team. The data has not been augmented or modified in any way.

Table 4.1 – Dataset sizes and label distribution

Dataset	**Size**	**Misogynous memes**	**Not misogynous memes**
Training	10000	5000	5000
Trial	100	50	50
Validation	1000	500	500

Source: the author

4.4.2.1 Examples of memes

One crucial aspect of this task is the multimodality of inputs. Most of the time, a meme requires both textual and visual information to be correctly understood. Not only because the punch line usually comes in written form, but also because texts and images often contradict each other for humouristic purposes. Take for example Figure 4.4a. The text alone indicates a positive feeling towards an object that makes sandwiches. The image, if one would remove the caption, would show a woman standing in front of a fridge. But when taken into consideration simultaneously, it is a sexist meme implying that women exist to make men sandwiches.

Figure 4.4a is an example in which both modalities are necessary for the correct prediction. It is not always the case that both textual and visual information are necessary,

take for example Figure 4.4b, Figure 4.5a, and Figure 4.5b. For Figure 4.4b it is clear that any of the two modalities would suffice for correct classification. Figure 4.5a, however, only needs the textual information to be classified, while Figure 4.5b is the opposite, needing only the visual information.

Figure 4.4 – Example of memes with contradicting or corroborating texts/images

(a) Example of a meme from the MAMI dataset with contradicting textual and visual information

Source: dataset provided by the MAMI organizers

(b) Example of a meme from the MAMI dataset in which both textual and visual information point towards a positive prediction (*i.e.*, misogynous)

Source: dataset provided by the MAMI organizers

4.4.2.2 Word Frequency

Table 4.2 is obtained by first normalizing all text inputs and then analyzing token frequency. The normalization process involves punctuation removal, changing text to lower case, stopword removal, and filtering by tokens with length greater than 3.

The memes present in the training data were visually analyzed in search of patterns in either visual information or text information. One pattern that was noticed is the presence of the words *women, woman, girl,* and *female* in misogynous memes. It is interesting to note that, of all the top-8 words, only one – *like* – was not linked to a tendency toward positive (misogynous) labels. The presence of words like *rape, fuck, prostitute, bitch*, and *hooker* also highlight sexually violent tendencies towards women.

Figure 4.5 – Example of memes in which one modality suffices to reach the correct prediction

(a) Example of a meme from the MAMI dataset in which textual information would suffice for the correct prediction

Source: dataset provided by the MAMI organizers

(b) Example of a meme from the MAMI dataset in which visual information would suffice for the correct prediction

Source: dataset provided by the MAMI organizers

Table 4.2 – Table showing the top 100 words and their respective frequencies.

Begin of Table 4.2		
Ranking	Word	Frequency
1	women	1208
2	like	863
3	woman	682
4	wife	502
5	girlfriend	485
6	girl	432
7	kitchen	414
8	house	376
9	call	356
10	girls	352
11	female	351
12	make	346
13	want	326
14	know	324
15	people	316

Continuation of Table 4.2		
Ranking	Word	Frequency
16	made	281
17	feminist	270
18	time	267
19	good	258
20	clean	247
21	first	240
22	cheat	231
23	back	225
24	look	225
25	hooker	221
26	never	218
27	think	218
28	meme	211
29	prostitute	205
30	cooking	202
31	need	196
32	would	195
33	feminism	193
34	work	193
35	love	191
36	said	189
37	still	188
38	right	187
39	home	182
40	feminists	179
41	male	175
42	fuck	173
43	life	162
44	going	162
45	white	161
46	years	156
47	tell	155

Continuation of Table 4.2		
Ranking	Word	Frequency
48	friend	154
49	take	149
50	stop	141
51	really	141
52	find	139
53	memes	137
54	rape	135
55	always	133
56	every	133
57	real	133
58	shit	133
59	gold	133
60	world	132
61	better	132
62	fucking	130
63	says	130
64	best	130
65	well	127
66	sandwich	126
67	mematic	125
68	something	123
69	milf	123
70	little	122
71	face	122
72	much	122
73	could	122
74	dick	121
75	game	121
76	without	119
77	give	119
78	black	116
79	someone	116

Continuation of Table 4.2		
Ranking	Word	Frequency
80	getting	116
81	man	115
82	dishwasher	114
83	money	114
84	bitch	111
85	today	109
86	ever	107
87	guys	107
88	last	106
89	even	106
90	show	106
91	food	105
92	great	105
93	come	104
94	rights	104
95	coronavirus	104
96	boys	102
97	friends	101
98	also	101
99	everyone	99
100	finally	99
End of Table 4.2		

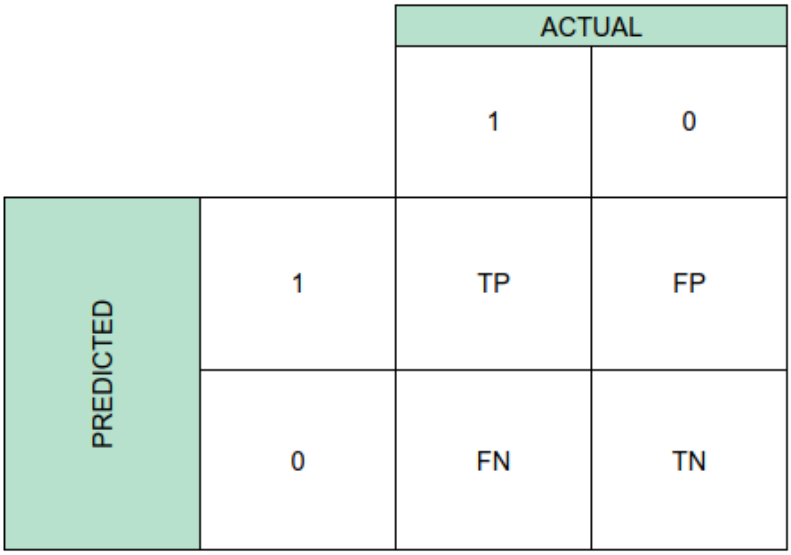
Table 4.2 – Source: the author

4.4.3 Metrics

The main evaluation metric used in the task and during training is Macro-F1. Here we also report True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) Rates, which concepts are shown in Figure 4.6. TP is the number of positive predictions that are actually positive. TN is the same applied to the negative class. FP is the number of positive predictions that are actually negative, and FN is the number of

negative predictions that are actually positive. Additionally, we also present the test used to determine the statistical significance of differences in performances between models, the Wilcoxon signed-rank test.

Figure 4.6 – A confusion matrix



Source: the author

4.4.3.1 Accuracy

Accuracy is the number of correct guesses divided by the total of guesses, as seen in Equation 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

4.4.3.2 Precision

Precision measures how many of the predicted positives are, in fact, positives. It is calculated as seen in Equation 4.2.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.4.3.3 Recall

Recall answers a slightly different question. It measures how many of the actual positives were predicted as such. It is calculated as seen in Equation 4.3.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.4.3.4 Macro-F1

F1 uses both Precision and Recall in a harmonic mean, and is calculated as seen in Equation 4.4. Using F1 allows us to penalize higher scores in FP and FN, which is important for tasks dealing with health issues and safety, for example. Macro-F1 is the mean of F1 scores for all classes. It is better than accuracy for datasets with imbalanced classes.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.4.3.5 Wilcoxon signed-rank

The Wilcoxon signed-rank test is a paired difference test. It is used when the differences in measurements might not follow a normal distribution, and it's goal is to determine whether the population mean ranks differ. This test compares all of two models' predictions and determines if the difference in performance can be considered not caused by randomness.

5 RESULTS

In this section, we report on our experimental results organized around four questions.

Table 5.1 – Macro-F1 scores, true positive and negative rates, and false positive and false negative rates for our models

Model	*Macro-F1*	*TP*	*TN*	*FP*	*FN*
Image-Grid	0.599	0.864	0.382	0.618	0.136
MMBT-Grid	0.649	0.866	0.462	0.538	0.134
VisualBERT	0.666	**0.874**	0.484	0.516	**0.126**
Text BERT	0.674	0.684	**0.664**	**0.336**	0.316
VisualBERT COCO	0.679	0.786	0.580	0.420	0.214
ViLBERT CC	0.697	0.836	0.570	0.430	0.164
ViLBERT	**0.698**	**0.874**	0.540	0.460	**0.126**

Source: the author

5.1 What are the best and worst models?

The results obtained by each model can be seen in Table 5.1. The models which are *not* in bold are the ones used for the competition. Considering the original models used for the competition, the best and worst-performing models were, respectively, ViLBERT and MMBT-Grid, with Macro-F1 scores of 0.698 and 0.649. With this score, ViLBERT ranked 32^{nd} on Subtask A [1]. It is worth pointing out that they had very similar values for TP-rate. MMBT-Grid achieved a value of 0.866, despite being the worst-ranked among all five original models. That means it had a good performance in identifying misogynistic memes. The problem is evidenced by the TN and FP rates. MMBT-Grid was the worst at classifying memes that are not misogynistic, with a TN-rate of 0.462, the lowest of all. It also has the highest FP rate of 0.538. Analyzing ViLBERT's metrics, we can see that what guaranteed its first place among the five models was the TP and FN rate, which were, respectively, the highest and the lowest. VisualBERT COCO was the best at correctly classifying the negative class (TN rate = 0.581), but it also had, by far, the highest FN rate (0.21).

However, when considering all models analysed, including Image-Grid and Text BERT, we can observe that the worst-performing model changes. Image-Grid, the visual unimodal model takes the last spot, with a Macro-F1 score of 0.599, being far out-

[1] <https://competitions.codalab.org/competitions/34175#results>

performed by the others, and having a difference of 0.099 compared to the best model, ViLBERT. Although this poor performance is disappointing, it is not entirely unexpected. Analyzing the Hateful Memes' results, we can see that in their work too, Image-Grid was the model to perform worst. Image-Grid's metrics of TP and FP are interesting, because the model, although performing the worst out of all models, had a TP rate close to ViLBERT's. However, its FP rate was by far the highest, suggesting that this model had a tendency to predict images as belonging to the positive class (*i.e.*, misogynous).

Text BERT, the other model added after the competition, had a surprisingly good performance. In the Hateful Memes, the model was short of surpassing MMBT-Grid by 0.03 points (in accuracy). Our results, however, show that Text BERT not only outperformed MMBT-Grid by a large margin, (0.679 compared to 0.649), but it also outperformed VisualBERT, which had a score of 0.666. The TN and FN metrics suggest that this model has a tendency of predicting a negative label (*i.e,* not misogynous). It is interesting to see that, while it had the greatest TN rate, it also had the greatest FN rate, and the lowest TP rate.

Taking into consideration both added unimodal models, we can see that while Image-Grid had a tendency of predicting the positive label, Text BERT had a tendency of doing the opposite. This is interesting because it further corroborates the hypothesis that both modalities are necessary to correctly interpret a meme. This is, again, explained by the incongruity theory (MORREALL, 2020).

Since the models share the same architecture (ViLBERT with ViLBERT CC and VisualBERT with VisualBERT COCO), the architecture can not be the explanation for the differences in performance. However, the differences in performance can be explained by the usage of uni or multimodal pretraining, given that the modality of pretraining is the only distinctive factor between the models. This is evidenced by the differences in scores obtained by unimodally pretrained models (VisualBERT and ViLBERT) and that by multimodally pretrained models (VisualBERT COCO and ViLBERT CC). Additionally, what seems to have impacted scores the most is the use of image features during training, since MMBT-Grid and Image-Grid, the models that do not use features in their inputs, perform the worst.

5.2 Do multimodally pretrained models perform better?

It is interesting to notice that there was no great difference in performance between unimodally and multimodally pretrained models, such as VisualBERT vs. VisualBERT COCO and ViLBERT vs. ViLBERT CC. This finding is in line with Kiela et al. (2021) on the Hateful Memes dataset. Nevertheless, while multimodally pretrained models were slightly better on Hateful Memes, here the unimodally pretrained version of ViLBERT yielded slightly better results, but the difference was not statistically significant (according to a Wilcoxon signed-rank test p-value of 0.71).

5.3 Do multimodal models perform better?

By analyzing Table 5.1 we can see that all multimodal models perform better than Image-Grid by statistically significant differences in Macro-F1. The differences were tested performing a Wilcoxon test with 95% significance level. However, when comparing with Text BERT, only three multimodal models performed better: two statistically significant at a 95% significance level, ViLBERT and ViLBERT CC, and one with a not statistically significant difference, VisualBERT COCO. The other two multimodal models, *i.e.* VisualBERT and MMBT-Grid, perform worse with score differences of 0.008 and 0.025. The difference between MMBT-Grid and Text BERT is considered not statistically significant. Furthermore, when comparing VisualBERT with Text BERT, the difference is also considered not statistically significant.

Table 5.2 – Percentage of memes correctly classified by at least *N* models.

% of instances Now - MAMI	**correctly predicted by N**
92.5% - 89.89%	At least 1
84.8% - 77.58%	At least 2
77.3% - 69.67%	At least 3
70.4% - 61.76%	At least 4
62.8% - 47.95%	At least 5
51.5%	At least 6
34.0% - 47.95%	All models

Source: the author

5.4 Can combining classifiers improve classification performance?

To answer this question for the MAMI competition (FERSINI et al., 2022), we analyzed the predictions of the seven models for each instance on the evaluation dataset. The analysis is extended to include the two new models.

For the competition, the analysis shows that, if we were to use a simple majority voting system to determine the predicted label for images, the obtained Macro-F1 score would be 69.62%, which does not surpass ViLBERT's Macro-F1 score, which is 69.85%. The same analysis shows that a majority voting system would also not be helpful after including the unimodal models, because the majority's Macro-F1 score is 66.45%. Additionally, we tried combining the predictions of the classifiers by averaging their output probabilities. Similar to what we found with majority voting, there were no performance improvements in relation to ViLBERT alone. Although the two new models helped increase the Macro-F1 score resulting from averaging the five original models, which was 69.29%, the resulting Macro-F1 score, 69.66%, is still lower than ViLBERT's, even if by a small amount.

Table 5.3 – Pearson correlation for each pair of models

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
ViLBERT CC (1)	1.00	0.66	0.58	0.61	0.78	0.31	0.57
VisualBERT (2)		1.00	0.57	0.56	0.69	0.34	0.47
VisualBERT COCO (3)			1.00	0.58	0.59	0.27	0.56
MMBT-Grid (4)				1.00	0.61	0.27	0.58
ViLBERT (5)					1.00	0.35	0.55
Image-Grid (6)						1.00	0.20
Text BERT (7)							1.00

Source: the author

5.5 How correlated are the models?

Table 5.3 shows the Pearson correlation coefficient calculated for all pairs of models to measure their level of agreement, *i.e.* how many images they classified with the same label. We can see that ViLBERT and ViLBERT CC have the highest score, 0.78. We initially assumed that the reason for their high similarity was that they share the same architecture, but further analysis showed that VisualBERT and VisualBERT COCO, the other models that also share architectures, have low similarity. Therefore, the initial hy-

pothesis was wrong and we can suppose that the reason for the difference in similarity resides in the pretraining modality, since that is the only distinction between the models. It might also be the case that the difference in how ViLBERT and VisualBERT deal with multimodality might have caused VisualBERT to be more sensitive to the pretraining modality, and that would explain why the correlation value between VisualBERT and VisualBERT COCO is lower compared to that between ViLBERT and ViLBERT CC. We see that MMBT-Grid and ViLBERT have a correlation score of 0.61, while the lowest score is between Image-Grid and Text BERT, 0.20. This low score between the two unimodal models is expected, since Image-Grid tended to predict inputs as misogynous, and Text BERT tended to predict the opposite and can be explained by the contradiction necessary to create the incongruity that induces humor. The fact that all correlation scores can be classified between strong and moderate explains why there were no gains in combining the models in an ensemble.

5.6 Is there any pattern in memes that were erroneously classified?

We analyzed images that were wrongly classified by all seven models. They were, in total, 75 images. Through visual inspection, we were able to identify a pattern in the captions. We noticed that most false positives contained words like "girl", "girls", "woman", and "women", while false negatives did not present these words. To confirm this, we examined the frequency of these words in training and test datasets. The term "girl" appeared in approximately 4.57% of not misogynous memes in the training dataset, and in 6.37% of misogynous memes, that is, 1.39 times more often. This proportion, however, is almost reversed in the test dataset, in which the term appears in 11.1% of *not* misogynous memes, and only in 7.1% of misogynous memes, that is, 1.56 times *less* frequently. This might explain the high number of wrong classifications for memes that contain this word. For the term "women", training dataset analysis shows that 8.27% of misogynous memes had this word, while appearing in only 2% of not misogynous memes, about 4.13 times less often, while in the test dataset, 5.8% of misogynous memes had it, and 2.4% of not misogynous memes, that is, 2 times less. The change in word frequency for this term might also have contributed to misclassification.

The inclusion of the two unimodal models allowed analyses regarding memes that were correctly classified only by one of them. These cases might indicate that multimodality might have impeded reaching the correct predictions.

Analyzing Image-Grid, the unimodal image model, it is possible to see that there are cases like Figure 5.1a in which texts induce models to believe the meme is misogynous. Still, the image itself is innocent, depicting baby ducklings (*chicks*), for example. Figure 5.1b is a more interesting case in which text is close to ineligible, with slangs like *OwO* (used because it resembles a face) and *blrussed*, which is a typo of *blursed*, the mixing of *blessed* and *cursed*, and the only words that really stand out are *image* and *dishwasher*. The latter is a term closely related to the idea of the *sandwich-maker*, and both are frequently used in the dataset to describe women as objects for men. Another pattern is that many images correctly classified only by Image-Grid depict women in sexual or exposing situations, like Figure 5.2a and Figure 5.3, but have texts that are not unquestionably misogynous. This uncertainty might have led the other six models to judge the image as not misogynous.

An even more evident pattern is observed in BERT, the unimodal text model. All images correctly classified only by BERT are not misogynous. That fact by itself does not mean much. However, when considering that 89% of these memes depict women, it becomes clear that models with visual inputs tend to classify memes with women as misogynous. Examples of this can be seen in Figure 5.2b and Figure 5.3.

Figure 5.1 – Example of Memes with contradicting or unclear texts/images

(a) Example of a meme with contradicting text and image

(b) Example of a meme with a term that leads models with textual input to making the wrong prediction (positive label, *i.e.* misogynous)

Source: dataset provided by the MAMI organizers

Figure 5.2 – Example of memes with women in sexual or exposing situations

(a) Example of a meme depicting a drunk woman with text that is unclear if it should be considered misogynous or not.

(b) Example 1 of a meme depicting a woman that is wrongly classified as misogynous by models with visual input

Source: dataset provided by the MAMI organizers

Figure 5.3 – Example 2 of a meme depicting a woman that is wrongly classified by models with visual input

Source: dataset provided by the MAMI organizers

6 CONCLUSION

In this work, we described the use of multimodal models in the task of misogyny identification and analyzed the results obtained. Using Hateful Memes and MMF as inspiration, we wanted to replicate their methods in a similar context and validate if the same models would be able to learn how to identify a specific type of hatred, aimed at women. Although hateful and misogynistic memes share some overlap, there are important distinctions between them, regarding different vocabulary, context, and targets (*i.e.* hate can be directed towards anyone, while misogyny cannot).

We trained seven models and confirmed that they reach similar performances in this dataset as they do in Hateful Memes. Our best model, ViLBERT, reached a Macro-F1 score of 0.698 and ranked 32^{nd} out of 83 on the leaderboard in the MAMI competition[1]. We analyzed the predictions of all models and identified patterns in mistakes. We showed that multimodality can be a double-edged sword and help establish the correct context, but also introduce confusion in specific cases. We showed that using a majority voting system with all models would not be beneficial.

The models could be further improved by hyper-parameter tuning. We could also have experimented with late/early fusion, which, as suggested by Hateful Memes (KIELA et al., 2021), has an impact on performance, and we leave this as future work. Furthermore, it would be interesting to assess the performance of the same models used in this work but with the additional use of techniques shown in Chapter 3, like object, face and web entity detection.

In conclusion, we confirm that automatic misogyny identification with the utilized models is possible but it still has a lot to improve. We show that the models achieve similar results in the MAMI dataset as they do in the Hateful Memes (KIELA et al., 2021).

Our participation on the MAMI challenge is reported in a paper that is under review (LORENTZ; MOREIRA, 2022).

[1] <https://competitions.codalab.org/competitions/34175>

REFERENCES

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **ArXiv**, v. 1409, 09 2014.

BENGIO, Y. et al. A neural probabilistic language model. **J. Mach. Learn. Res.**, JMLR.org, v. 3, n. null, p. 1137–1155, mar 2003. ISSN 1532-4435.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

CHEN, Y.-C. et al. **UNITER: UNiversal Image-TExt Representation Learning**. arXiv, 2019. Available at: <https://arxiv.org/abs/1909.11740>.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, 2018. Available at: <https://arxiv.org/abs/1810.04805>.

DRAKETT, J. et al. Old jokes, new media – online sexism and constructions of gender in internet memes. **Feminism & Psychology**, v. 28, n. 1, p. 109–127, 2018. Available at: <https://doi.org/10.1177/0959353517727560>.

FARRELL, T. et al. Exploring misogyny across the manosphere in reddit. In: **Proceedings of the 10th ACM Conference on Web Science**. New York, NY, USA: Association for Computing Machinery, 2019. (WebSci '19), p. 87–96. ISBN 9781450362023. Available at: <https://doi.org/10.1145/3292522.3326045>.

FERSINI, E. et al. SemEval-2022 Task 5: Multimedia automatic misogyny identification. In: **Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)**. [S.l.]: Association for Computational Linguistics, 2022.

GAN, Z. et al. **Large-Scale Adversarial Training for Vision-and-Language Representation Learning**. arXiv, 2020. Available at: <https://arxiv.org/abs/2006.06195>.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: . [S.l.: s.n.], 2010. v. 15.

HE, K. et al. Deep residual learning for image recognition. In: . [S.l.: s.n.], 2016. p. 770–778.

HOCHREITER, S. **Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München**. 1991.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov 1997. ISSN 0899-7667. Available at: <https://doi.org/10.1162/neco.1997.9.8.1735>.

HU, R. et al. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020.

HUTCHINS, W. J.; DOSTERT, L.; GARVIN, P. The georgetown-i.b.m. experiment. In: **In**. [S.l.]: John Wiley & Sons, 1955. p. 124–135.

IOFFE, S.; SZEGEDY, C. **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**. arXiv, 2015. Available at: <https://arxiv.org/abs/1502.03167>.

JIANG, Y. et al. **Pythia v0.1: the Winning Entry to the VQA Challenge 2018**. 2018.

JORDAN, M. I. Chapter 25 - serial order: A parallel distributed processing approach. In: DONAHOE, J. W.; Packard Dorsel, V. (Ed.). **Neural-Network Models of Cognition**. North-Holland, 1997, (Advances in Psychology, v. 121). p. 471–495. Available at: <https://www.sciencedirect.com/science/article/pii/S0166411597801112>.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, v. 349, n. 6245, p. 255–260, 2015. Available at: <https://www.science.org/doi/abs/10.1126/science.aaa8415>.

KIELA, D. et al. **Supervised Multimodal Bitransformers for Classifying Images and Text**. arXiv, 2019. Available at: <https://arxiv.org/abs/1909.02950>.

KIELA, D. et al. **The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes**. 2021.

KRISHNA, R. et al. **Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations**. arXiv, 2016. Available at: <https://arxiv.org/abs/1602.07332>.

LI, L. H. et al. **VisualBERT: A Simple and Performant Baseline for Vision and Language**. 2019.

LI, X. et al. **Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks**. arXiv, 2020. Available at: <https://arxiv.org/abs/2004.06165>.

LIN, T.-Y. et al. **Microsoft COCO: Common Objects in Context**. 2015.

LORENTZ, G. A.; MOREIRA, V. P. INF-UFRGS at SemEval-2022 Task 5: analyzing the performance of multimodal models. In: **Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)**. [S.l.]: Association for Computational Linguistics, 2022.

LU, J. et al. **ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks**. 2019.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In: **Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981. (IJCAI'81), p. 674–679.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **CoRR**, abs/1301.3781, 2013. Available at: <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119.

MORREALL, J. Philosophy of Humor. In: ZALTA, E. N. (Ed.). **The Stanford Encyclopedia of Philosophy**. Fall 2020. [S.l.]: Metaphysics Research Lab, Stanford University, 2020.

MUENNIGHOFF, N. **Vilio: State-of-the-art Visio-Linguistic Models applied to Hateful Memes**. arXiv, 2020. Available at: <https://arxiv.org/abs/2012.07788>.

NOZZA, D. Exposing the limits of zero-shot cross-lingual hate speech detection. In: **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**. Online: Association for Computational Linguistics, 2021. p. 907–914. Available at: <https://aclanthology.org/2021.acl-short.114>.

PAPERT, S. The summer vision project. In: . [S.l.: s.n.], 1966.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **EMNLP**. [S.l.: s.n.], 2014. v. 14, p. 1532–1543.

POOLE, D.; MACKWORTH, A.; GOEBEL, R. **Computational Intelligence: A Logical Approach**. USA: Oxford University Press, Inc., 1997. ISBN 0195102703.

SEARLE, J. R. Minds, brains, and programs. **Behavioral and Brain Sciences**, Cambridge University Press, v. 3, n. 3, p. 417–424, 1980.

SHARMA, P. et al. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 2556–2565. Available at: <https://aclanthology.org/P18-1238>.

SHIFMAN, L. **Memes in digital culture**. [S.l.]: MIT press, 2013.

SINGH, A. et al. **MMF: A multimodal framework for vision and language research**. 2020. <https://github.com/facebookresearch/mmf>.

SU, W. et al. **VL-BERT: Pre-training of Generic Visual-Linguistic Representations**. arXiv, 2019. Available at: <https://arxiv.org/abs/1908.08530>.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2**. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 3104–3112.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. **Mind**, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Available at: <https://doi.org/10.1093/mind/LIX.236.433>.

TURK, M.; PENTLAND, A. Face recognition using eigenfaces. In: **Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 1991. p. 586–591.

VASWANI, A. et al. Attention is all you need. In: **Proceedings of the 31st International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 6000–6010. ISBN 9781510860964.

VELIOGLU, R.; ROSE, J. **Detecting Hate Speech in Memes Using Multimodal Deep Learning Approaches: Prize-winning solution to Hateful Memes Challenge**. arXiv, 2020. Available at: <https://arxiv.org/abs/2012.12975>.

WU, Y. et al. **Detectron2**. 2019. <https://github.com/facebookresearch/detectron2>.

YU, F. et al. **ERNIE-ViL: Knowledge Enhanced Vision-Language Representations Through Scene Graph**. arXiv, 2020. Available at: <https://arxiv.org/abs/2006.16934>.

ZHU, R. **Enhance Multimodal Transformer With External Label And In-Domain Pretrain: Hateful Meme Challenge Winning Solution**. arXiv, 2020. Available at: <https://arxiv.org/abs/2012.08290>.

APPENDIX A — MODEL CONFIGURATIONS

A.1 MMBT-Grid

Listing A.1 – MMBT Defaults, includes Listing A.2 and Listing A.3 respectively.

```
includes:
- configs/models/mmbt/classification.yaml

scheduler:
  type: warmup_linear
  params:
    num_warmup_steps: 2000
    num_training_steps: ${training.max_updates}

optimizer:
  type: adam_w
  params:
    lr: 1e-5
    eps: 1e-8

evaluation:
    metrics:
    - accuracy
    - binary_f1
    - roc_auc

training:
  batch_size: 32
  lr_scheduler: true
  max_updates: 22000
  early_stop:
    criteria: hateful_memes/roc_auc
    minimize: false

checkpoint:
  pretrained_state_mapping:
```

```
    bert: bert
```

Listing A.2 – configs/models/mmbt/classification.yaml

```
model_config:
  mmbt:
    training_head_type: classification
    num_labels: 2
    losses:
    - type: cross_entropy
```

Listing A.3 – configs/datasets/hateful_memes/bert.yaml

```
dataset_config:
  hateful_memes:
    processors:
      text_processor:
        type: bert_tokenizer
        params:
          tokenizer_config:
            type: bert-base-uncased
            params:
              do_lower_case: true
          mask_probability: 0
          max_seq_length: 128
```

A.2 ViLBERT

Listing A.4 – ViLBERT Defaults, includes Listing A.5

```
includes:
- configs/datasets/hateful_memes/with_features.yaml

model_config:
  vilbert:
    training_head_type: classification
    num_labels: 2
```

```
    losses:
    - cross_entropy

dataset_config:
  hateful_memes:
    return_features_info: true
    processors:
      text_processor:
        type: bert_tokenizer
        params:
          tokenizer_config:
            type: bert-base-uncased
            params:
              do_lower_case: true
          mask_probability: 0
          max_seq_length: 128
      transformer_bbox_processor:
        type: transformer_bbox
        params:
          bbox_key: bbox
          image_width_key: image_width
          image_height_key: image_height

optimizer:
  type: adam_w
  params:
    lr: 1e-5
    eps: 1e-8

scheduler:
  type: warmup_linear
  params:
    num_warmup_steps: 2000
    num_training_steps: ${training.max_updates}

evaluation:
```

```
  metrics:
  - accuracy
  - binary_f1
  - roc_auc

training:
  batch_size: 32
  lr_scheduler: true
  max_updates: 22000
  find_unused_parameters: true
  early_stop:
    criteria: hateful_memes/roc_auc
    minimize: false

checkpoint:
  pretrained_state_mapping:
    model.bert: model.bert
```

Listing A.5 – configs/datasets/hateful_memes/with_features.yaml

```
dataset_config:
  hateful_memes:
    use_images: false
    use_features: true
    # Disable this in your config if you
    # do not need features info
    # and are running out of memory
    return_features_info: true
```

A.3 ViLBERT CC

Listing A.6 – ViLBERT CC Configuration, includes Listing A.4

```
includes:
- ./defaults.yaml

checkpoint:
```

```
  resume_pretrained: true
  resume_zoo: vilbert.pretrained.cc.original
```

A.4 VisualBERT

Listing A.7 – VisualBERT Direct, includes Listing A.8

```
includes:
- ./defaults.yaml

training:
  batch_size: 128
```

Listing A.8 – ./defaults.yaml, includes Listing A.5

```
includes:
- configs/datasets/hateful_memes/with_features.yaml

model_config:
  visual_bert:
    training_head_type: classification
    num_labels: 2
    losses:
    - cross_entropy

dataset_config:
  hateful_memes:
    return_features_info: true
    processors:
      text_processor:
        type: bert_tokenizer
        params:
          tokenizer_config:
            type: bert-base-uncased
            params:
              do_lower_case: true
          mask_probability: 0
```

```
            max_seq_length: 128

optimizer:
  type: adam_w
  params:
    lr: 5e-5
    eps: 1e-8

scheduler:
  type: warmup_linear
  params:
    num_warmup_steps: 2000
    num_training_steps: ${training.max_updates}

evaluation:
  metrics:
  - accuracy
  - binary_f1
  - roc_auc

training:
  batch_size: 64
  lr_scheduler: true
  max_updates: 22000
  find_unused_parameters: true
  early_stop:
    criteria: hateful_memes/roc_auc
    minimize: false

checkpoint:
  pretrained_state_mapping:
    model.bert: model.bert
```

A.5 VisualBERT COCO

Listing A.9 – VisualBERT COCO Defaults, includes Listing A.8

```
includes:
- ./defaults.yaml

checkpoint:
  resume_pretrained: true
  resume_zoo: visual_bert.pretrained.coco
```

A.6 Image-Grid

Listing A.10 – Image-Grid Configuration

```
model_config:
  unimodal_image:
    classifier:
      type: mlp
      params:
        num_layers: 2
    losses:
    - type: cross_entropy

scheduler:
  type: warmup_linear
  params:
    num_warmup_steps: 2000
    num_training_steps: ${training.max_updates}

optimizer:
  type: adam_w
  params:
    lr: 1e-5
    eps: 1e-8

evaluation:
  metrics:
  - accuracy
```

```
  - binary_f1
  - roc_auc

training:
  batch_size: 32
  lr_scheduler: true
  max_updates: 22000
  early_stop:
    criteria: hateful_memes/roc_auc
    minimize: false

checkpoint:
  pretrained_state_mapping:
    base: base
```

A.7 BERT

Listing A.11 – BERT Configuration, includes Listing A.12, Listing A.13 and Listing A.14 respectively

```
includes:
- ./text.yaml
- configs/datasets/hateful_memes/bert.yaml
- configs/models/unimodal/bert.yaml

model_config:
  unimodal_text:
    classifier:
      type: mlp
      params:
        in_dim: 768
        num_layers: 2

training:
  batch_size: 128
```

Listing A.12 – ./text.yaml

```
model_config:
  unimodal_text:
    classifier:
      type: mlp
      params:
        num_layers: 2
    losses:
    - type: cross_entropy

scheduler:
  type: warmup_linear
  params:
    num_warmup_steps: 2000
    num_training_steps: ${training.max_updates}

optimizer:
  type: adam_w
  params:
    lr: 5e-5
    eps: 1e-8

evaluation:
  metrics:
  - accuracy
  - binary_f1
  - roc_auc

training:
  batch_size: 32
  lr_scheduler: true
  max_updates: 22000
  early_stop:
    criteria: hateful_memes/roc_auc
    minimize: false
```

```
checkpoint:
  pretrained_state_mapping:
    base: base
```

Listing A.13 – configs/datasets/hateful_memes/bert.yaml

```
dataset_config:
  hateful_memes:
    processors:
      text_processor:
        type: bert_tokenizer
        params:
          tokenizer_config:
            type: bert-base-uncased
            params:
              do_lower_case: true
          mask_probability: 0
          max_seq_length: 128
```

Listing A.14 – configs/models/unimodal/bert.yaml

```
model_config:
  unimodal_text:
    bert_model_name: bert-base-uncased
    text_hidden_size: 768
    num_labels: 2
    text_encoder:
      type: transformer
      params:
        bert_model_name:
            ${model_config.unimodal_text.bert_model_name}
        hidden_size: 768
        num_hidden_layers: 12
        num_attention_heads: 12
        output_attentions: false
        output_hidden_states: false

    classifier:
```

```
    params:
      in_dim: 768
```