Training a Deep Learning Model for Multimodal Data Fusion to Detect Misogyny

Dimitar Georgiev

dimitar.georgiev@students.finki.ukim.mk

ABSTRACT

Misogyny (hatred or prejudice against women) is unfortunately present all over the internet, in all online platforms. Since the volume of user-generated content is growing rapidly, manual monitoring and removing such harmful content is becoming nearly impossible. This issue calls for an automated solution, that can effectively, analyze, monitor, detect and filter misogynistic content. An automated solution makes it possible to address this issue at scale, dealing with enormous amounts of data in sufficient time. Existing approaches to misogyny detection have primarily focused on analyzing either textual or visual data and such deep learning solutions are already utilized in many online platforms. But user-generated content is not usually strictly textual or visual. For example, most content on Instagram, Facebook and Twitter contains both text and images. A combination of text and image data can provide a more nuanced understanding of the data, as misogyny can manifest in a not so obvious manner that can be easily detected by a deep learning model. By цоррецтлs fusing both these modalities in a single representation, deep learning classifiers can capture the overall intention of the content and detect misogyny more robustly. In this paper I present my attempt to train such a fusion model, as a part of my Natural Language Processing course.

Project Repository: https://github.com/Dimitar-G/AutomaticMisogynyDetection

1. INTRODUCTION

As misogyny continues to be a major pervasive issue in today's digital landscape, more and more online platforms are recognizing the importance of detecting and addressing misogynistic content that could potentially harm many people on the internet. It is important to detect such content as quickly as possible after (or before) publishing, in order to stop the spreading of misogynistic sentiments over the internet. The most sophisticated way of detecting such content is by utilizing properly trained deep learning models. The problem with today's approaches is that they use deep learning classifiers specifically trained on textual or visual data. These models excel in detecting misogyny in user-written text, or images posted on the internet, but they fail when the published content includes data of multiple modalities. For example, an Instagram post contains an image and an optional textual description, while a Twitter post contains text and an optional

image. Often, the misogynistic sentiment is subtle and cannot be detected by analyzing the textual and visual data separately and can only be detected when the full content is analyzed as a whole. In order to develop a classifier that takes the whole content of multiple modalities as input and classifies it in its entirety, it is important correctly fuse the different types of data in a single representation.

Although there are many fusion methods, they can be generally categorized as early fusion or late fusion methods. Early fusion refers to fusing the multiple modalities of data early in the process, either in their raw form, a preprocessed version of the data or even feature extraction representations. Usually, models trained on data which is formed by joining data of multiple modalities in their raw format, can better extract features and perform better with the price of enormous increase in training time and computational resources needed. Because of this, it is more preferred to preprocess the data of each modality separately and even perform feature extraction (using suitable methods/models for each modality), and then joining the results (concatenating or other methods). The joint representation can then be fed to a classification module which makes the decision based on the features. Late fusion, on the other hand, refers to separately preprocessing, performing feature extraction and even classifying the multiple modalities of data, so each of these results (or output nodes) can be used for decision making at the end of the process. Although late fusion methods provide flexibility in independently analyzing and processing each modality, it may not be efficient in effectively capturing the interaction between them.

In this project, I present my approach for creating and training a deep learning fusion model for detection of misogyny. I explore different architectures and methods, aiming to develop a classifier that can accurately detect misogyny in the used dataset. In the next sections, I describe in detail the dataset I used, experiments I conducted and the results that I achieved in terms of accuracy, precision, recall and f1 score.

2. RELATED WORK

The issue of detecting and combating misogyny in online platforms has garnered significant attention in recent years. Numerous studies have explored different approaches to tackle this problem, employing various techniques and methodologies. In this section, I review relevant works that focus on misogyny detection, with an emphasis on the fusion of text and image data.

Fabio Del Vigna et al. in their paper [1] presented their solution for detecting hateful speech in Facebook posts. For this purpose, they crawled their own dataset from Facebook, gathering 17567 comments and manually annotating them. They experimented with both classical machine learning algorithms such as Support Vector Machines (SVM) and with more sophisticated classifiers such as recurrent neural networks (LSTM). Surprisingly, they achieved similar results with both SVM and LSTM. On their three class hate detection task (no hate, weak hate, string hate), they achieved not so good accuracy of 64.61 using SVM and 60.5 using LSTM. However, they managed to achieve better results in a two class hate detection problem (hate, no hate), achieving 80.60 accuracy using SVM and 79.81 accuracy using LSTM.

In their paper [2], Ziqi Zhang and Lei Luo, discuss their approach to train deep learning architectures to detect and classify hate speech. They used a very large datasets of tweets (Twitter posts), gathered throughout their previous researches. Using different kind of deep architectures, they demonstrated some very good results in terms of precision, recall and f1 score. Although this paper along with the first one mentioned above does not directly refer to misogyny detection, the topic and methods used are closely related to it and are of great use.

A very interesting and useful paper I read was written by Abhinav Kumar Thakur et al. [3]. The paper delves into multimodal classification of internet memes (funny content containing images and text). This paper is very related to this project since it discusses fusion of the same two modalities and also includes misogyny as part of their classification process. The dataset used in this project is a combination of the MAMI dataset [6] (used in this project) and the Hateful Memes dataset (Kiela et al. 2020). In order to perform data fusion, they processed the textual data using BERT and BERTweet (BERT trained on Twitter data), the visual data using the CLIP (Contrastive Language-Image Pre-training) model and they concatenate the feature vectors in a single representation model that is finally used for classification. The best results they achieved are 0.701 accuracy in misogyny detection (MAMI Dataset), 0.688 accuracy in misogyny classification (MAMI Dataset) and 0.583 accuracy in hate detection (Hateful Memes Dataset).

During my research for this project, I found a paper [4] written as part of the SemEval-2022, Multimedia Automatic Misogyny Identification competition, written by the creators of the competition (Elisabeta Fersini et al.) after analyzing the overall results of the 65 teams involved. The performance of the proposed solutions by all participants is presented in terms of summary statistics of the F1-scores achieved. The overall results are presented in the table below:

	Min	**Q1**	**Mean**	**Median**	**StDev**	**Q3**	**Max**
Sub-task A	0.481	0.649	0.680	0.679	0.064	0.722	0.834
Sub-task B	0.467	0.634	0.663	0.680	0.059	0.706	0.731

Table 1: F1-score summary of the SemEval 2022 MAMI competition

For clarification, sub-task A refers to general misogyny detection and sub-task B refers to misogyny classification (identifying the type of misogyny).

Aside from the official paper which discusses the competition, the used dataset and the results, I found several papers from participants that explain their solutions in detail. One of these papers [5] is written by Shubham Kumar Barnwal et al. in which the authors discuss their methods in detail. Although they did not use the images for this task, they experimented with a few methods based on the textual data. For text preprocessing purposes they used lemmatization (NLTK wordnet), tokenization (keras) and TfidfVectorizer (scikit-learn). They achieved 0.656 F1-score using BERT, 0.631 using logistic regression, 0.584 using SVM and 0.651 using LSTM.

3. DATA

The dataset I used in this project is the SemEval 2022 Multimedia Automatic Misogyny Identification (MAMI) [6]. This dataset was published as a part of a CodaLab competition with the same name, with the purpose of developing modern solutions for multimodal misogyny detection and identification in online content. The dataset itself was gathered and put together by researchers from the University of Milan Bicocca, Google Jigsaw and the Polytechnical University of Valencia. The competition includes two sub-tasks A and B which are based on the same dataset. Sub-task A refers to detecting whether misogyny is present or not in a data point. This task is basically a binary classification problem. Sub-task B is a multiclass classification problem, in which multiple overlapping categories of misogyny (such as stereotype, shaming, objectification and violence) should be identified. This project primarily focuses on sub-task A.

The dataset is divided into three subsets: training, trial and test. The training subset consists of 10000 images, the trial subset consists of 100 images and the testing subset contains 1000 images. These subsets are intended to be used for training, validation and testing purposes, respectively. The images in the MAMI dataset are memes, which can be defined as visual content containing text, usually intended to be funny. Every image is of different dimensions, usually uneven (the width does not match the height), which has to be handled before using them for deep learning purposes. One CSV file is present for each of the image subsets, containing textual transcriptions and labels for each image. The first label in each record is about misogyny, and the rest of the labels are about the categories of misogyny.

4. METHOD

The method I used in this project to conduct binary classification with the purpose of detecting whether misogyny is present in internet memes, utilizes shallow (early) fusion in order to properly join the meaning of both image and textual data in a single representation. The first step in the classification pipeline is to separately preprocess the image and text data. For preprocessing of images, I used custom transforms that rescale the image based on its bigger dimension while maintaining the aspect ratio. In order for the result images to have a square shape, the images are padded with 0 pixels to fill up the necessary space (black pixel image paddings do not usually interfere with the learning process). The preprocessed images are of size 256x256 pixels. Also, the colors are normalized in order to match the requirements for the feature extraction model. This pipeline step also includes image augmentation such as random rotation, color jittering etc. Regarding the text preprocessing, I used a BERT tokenizer, which is a tokenizer that transforms the text exactly as needed for BERT. After the image and text are preprocessed, they are fed forward to the feature extraction process. The image is fed to a pretrained version of EfficientNet (pretrained on the ImageNet dataset) with excluded classification layers, so it outputs a feature vector. The image feature vector is of size 1280. The preprocessed text is fed to a pretrained version of BERT and a representation vector of size 768 is outputted. After both vectors are generated, they are concatenated in order to form a single joint representation vector of size 1280, of which the point is to capture the meaning of both image and text. This joint vector is then used as an input

for the classification module. The classification module is a fully connected neural network with one output node and is used to make a binary prediction. For implementation of all these steps, I used the PyTorch framework.

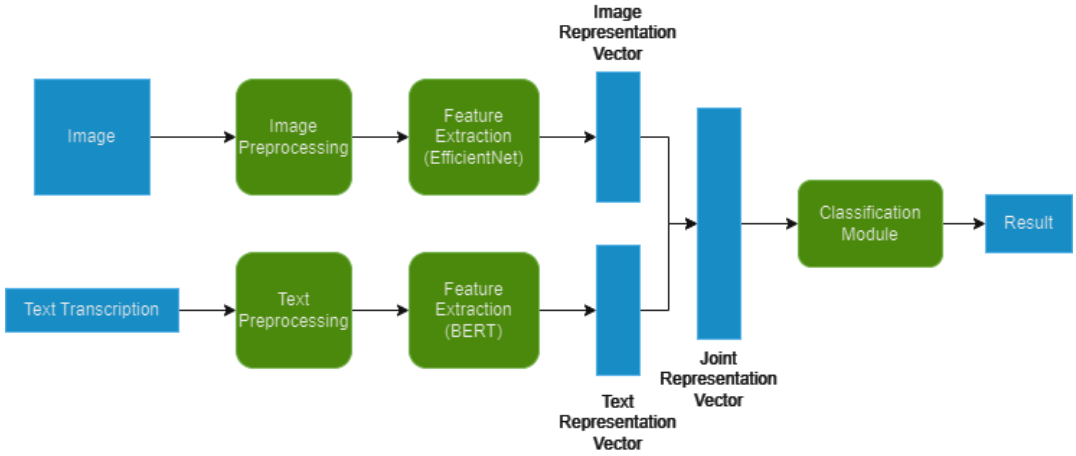


Image 1: Graphical representation of the classification pipeline

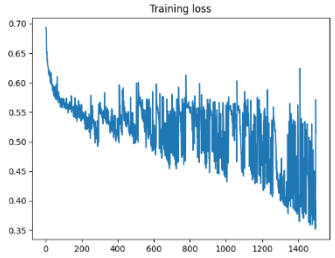
5. EXPERIMENTS AND RESULTS

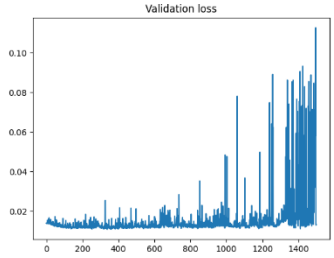
First, I tried training the fusion network as a whole, with the pretrained feature extraction models fully frozen (not trainable) in order to decrease training time. Despite this, I still faced some issues with the enormous training time needed per epoch. After analyzing the data flow, I realized that most of the time per epoch is spent by the preprocessing steps and slightly less but still notable, by the inference through the feature extraction models. To speed up the training process (and be able to conduct experiments), I preprocessed and saved all the data, so it can be used as such in every epoch. This greatly decreased the training time at the cost of not being able to make data augmentation. Although not as useful as real augmentation, I repeated this process with augmented data, so for each data point, there would be an augmented version saved too. At this point, the data was already preprocessed and the feature extraction models were frozen, so I performed feature extraction on all the saved preprocessed and augmented data, joined the feature vectors and saved the fused representation vectors for each data point. This way, I created a new dataset of feature vectors, that can be directly used for training a classifier. The training time was enormously decreased and I was able to make a few different experiments.

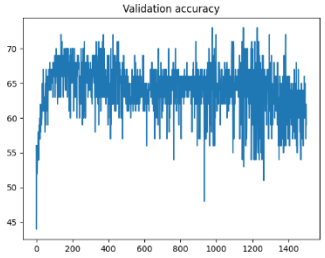
I documented nine different training attempts, all using slightly different training parameters and two different architectures for the classification module. One of the models consists of 4 fully connected layers of size 1024, 516, 258, 1, while the other model is a bit deeper, containing 5 fully connected layers of size 1024, 1024, 512, 512, 1 respectively. I trained these models using both the augmented and non-augmented vectors. Although both architectures are different, I managed to achieve similar results with both of them.

The best score I achieved regarding the validation data, was in a training attempt using the first (shallower) architecture, first trained on the non-augmented embeddings (1500 epochs) and then

trained again on the augmented embeddings (1000 more epochs on the best model), using a learning rate of 0.0001, batch size of 128, and shuffling of batches. The best model, regarding the validation set, achieved accuracy of 0.79 and F1-score of 0.769. Unfortunately, when tested on the testing dataset, it produced much lower results, with accuracy of 0.537 and F1-score of 0.574. After this, I continued searching for other models that performed good on the validation dataset, in order to test them again on the testing dataset, coming to a conclusion that all the models that achieved good scores on the validation dataset, underperformed on the testing dataset. After extracting other models that seemed promising (all of them had an F1-score over 0.7 on the validation set) and testing them, I found the highest testing F1-score to be 0.63. This was achieved by the second model with deeper architecture, trained in the same manner as the previously discussed attempt. The performance of this training attempt can be analyzed on images 2, 3 and 4.







Images 2, 3, 4: Training loss, validation loss and validation accuracy of one of the training attempts.

The previously discussed model which performed the best on the testing dataset was extracted at the end of this training process, despite of the growing of the validation loss. This could mean that the validation set is not representative enough of the whole problem and relying on it could be misleading in some circumstances.

6. CONCLUSION

In this project, I experimented with fusion of data of two modalities as an attempt to detect the presence of misogyny in internet memes. Although the trained models performed good on the validation dataset (F1-score over 75), they achieved less satisfactory results on the testing dataset, with the best model achieving 0.63 F1-score. The result is comparable to the average F1-score achieved by all participant teams on the MAMI competition as stated in the official paper (0.68). As further steps in this research, in order to improve the generalization of the model, I recommend doing active augmentation (in every epoch) which is more time-consuming than the method I used, trying different models for feature extraction and fine-tuning them to better fit the problem and experiment with deeper architectures for the classification module at the end of the pipeline. I believe that it is important to continue the research for automatic misogyny detection on multimodal content since it can help in making online platforms a safer space for everyone.

REFERENCES

[1] Fabio Del Vigna, Andrea Cimino, Felice Dell'Orletta, Marinella Petrocchi, Maurizio Tesconi, 2017, *Hate Me, Hate Me Not: Hate Speech Detection on Facebook*

[2] Ziqi Zhang, Lei Luo, October 2018, *Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter*

[3] Abhinav Kumar Thakur, Filip Ilievski, Hong-An Sandlin, Alain Mermoud, Zhivar Sourati, Luca Luceri, Riccardo Tommasini, 2023, *Multimodal and Explainable Internet Meme Classification*

[4] Elisabeta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, Jeffrey Sorensen, 2022, *SemEval-2022 Task 5: Multimedia Automatic Misogyny Identification*

[5] Shubham Kumar Barnwal, Ritesh Kumar, Rajendra Pamula, 2022, *IIT DHANBAD CODECHAMPS at SemEval-2022 Task 5: MAMI – Multimedia Automatic Misogyny Identification*

[6] Elisabeta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, Jeffrey Sorensen, SemEval-2022, *Multimedia Automatic Misogyny Identification (MAMI) Dataset*