August 26th 2021

Modergator: Automated Content Moderation for Hateful Memes, Speech and Text

Korbinian Koch, Skadi Dinter, Katrin Caragiuli

Modergator

bot

Info

Hi, I am Modergator 🐊 I help you moderating hateful content in groups. Just add me!

Bio

@modergator_bot

Username

Notifications

On

Photo by Björn Antonissen on Unsplash

17% of German youth aged 18-24 have been **personally affected** by hate speech online*

*YouGov/Institut für Demokratie und Zivilgesellschaft (2019)

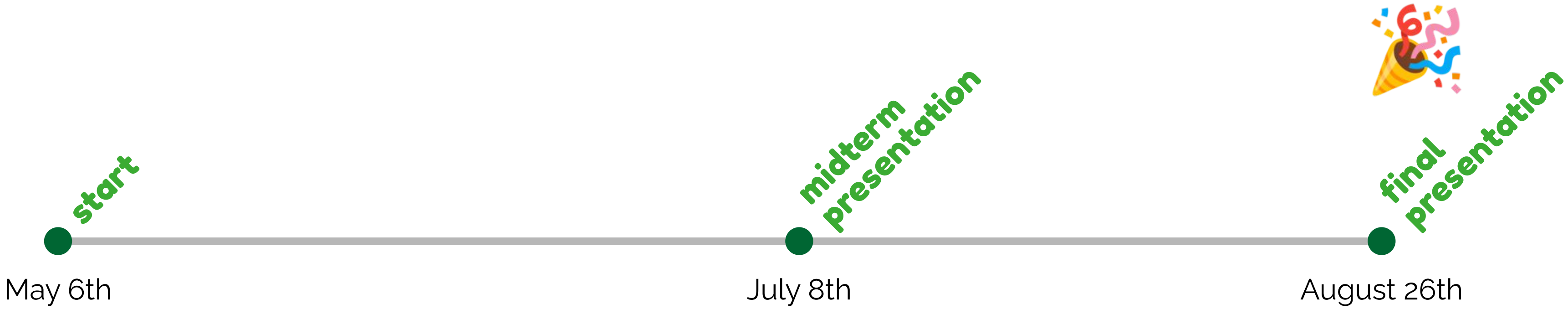
Photo by Andrew Guan on Unsplash

The Mission

Providing safer social spaces for everyone by making group moderation for hateful content easier and faster.

The Scope

- Master's project *Web Interfaces for Language Processing Systems*
- Team of 3(+) students
- duration of ~4 months



The Team

Korbinian

M.Sc. Intelligent Adaptive Systems

bot functionality &

voice processing

design

(and more)

Skadi

M.Sc. Informatics

meme model + api

interactivity and feedback

target group detection

(and more)

Katrin

M.Sc. Informatics

meme model + api

target group detection

meme detection api

(and more)

Modergator is ...

a collection of 6 APIs revolving around hate speech detection

1	2	3
4	5	6

curated components

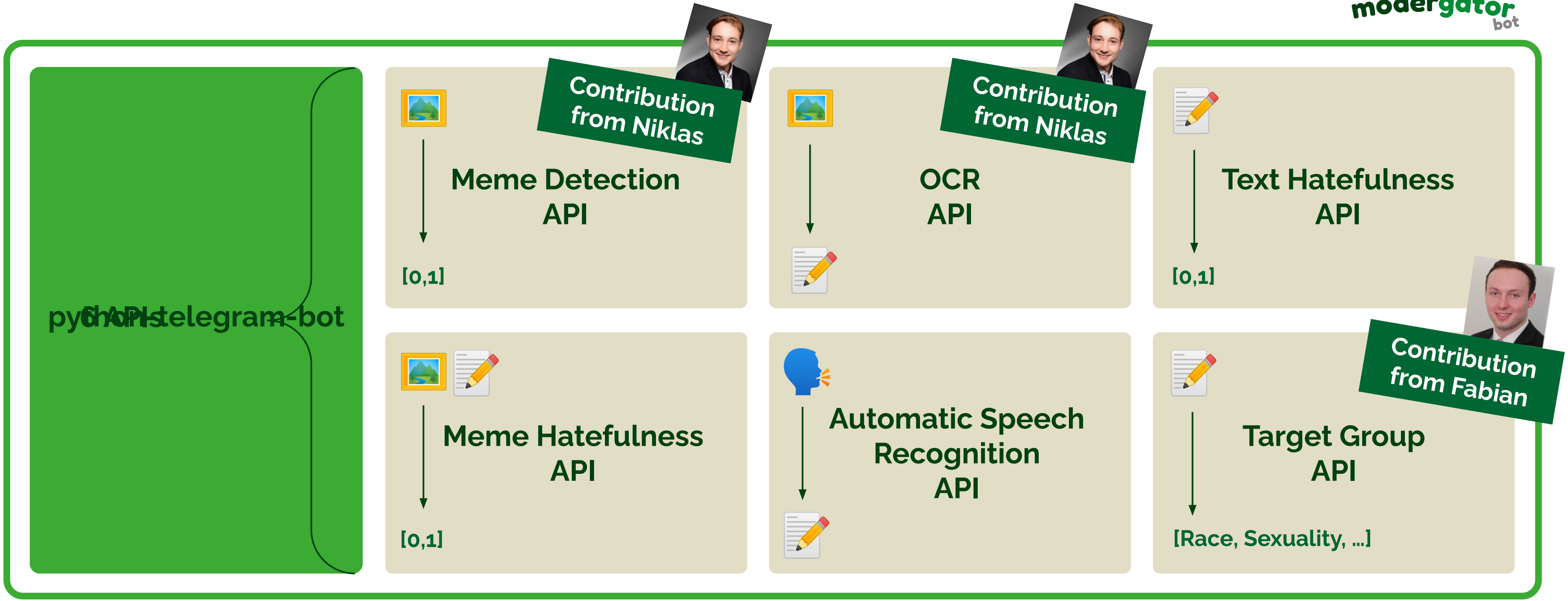
and

a ready-to-use Telegram bot moderating hateful content in groups

modergator bot

useable product

The Components



python-telegram-bot

6 APIs

Meme Detection API

[0,1]

Contribution from Niklas

OCR API

Contribution from Niklas

Text Hatefulness API

[0,1]

Meme Hatefulness API

[0,1]

Automatic Speech Recognition API

Target Group API

[Race, Sexuality, ...]

Contribution from Fabian

The Features

Automatic text classification: hateful, offensive or normal

Analyze voice messages

GDPR-friendly opt-out

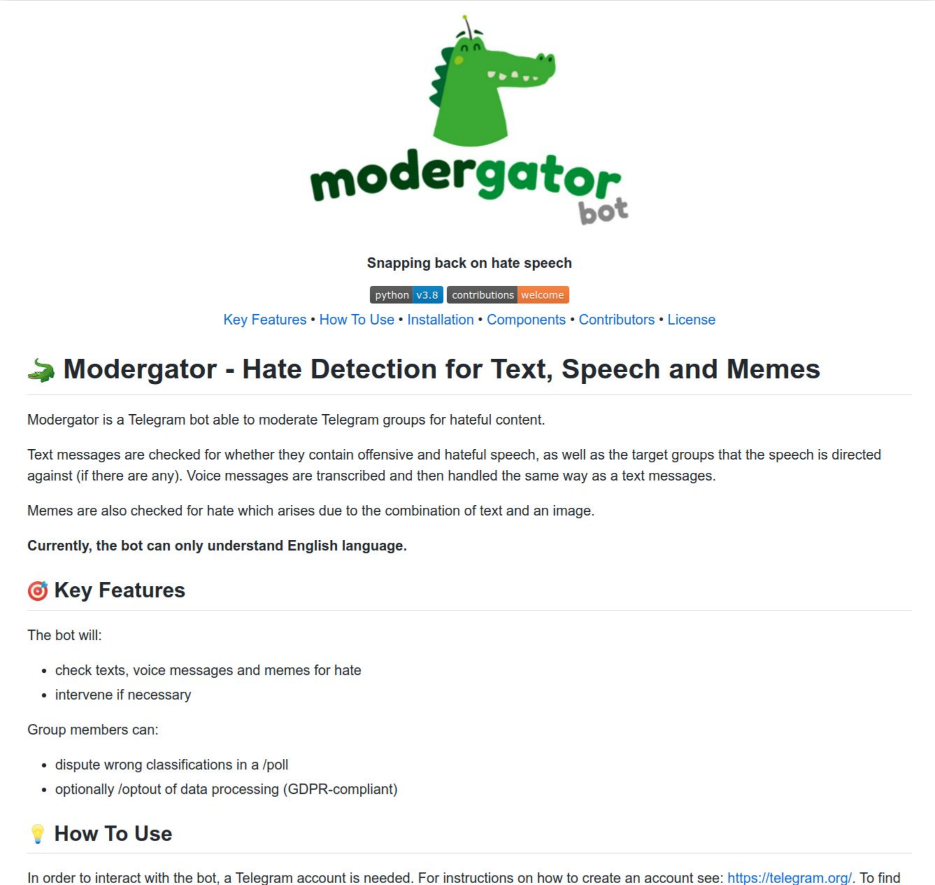
Group members can discuss the classification(s)

modergator bot

Detect affected target group(s)

Analyze memes

Published Project



Snapping back on hate speech

python v3.8 contributions welcome

Key Features • How To Use • Installation • Components • Contributors • License

🐊 Modergator - Hate Detection for Text, Speech and Memes

Modergator is a Telegram bot able to moderate Telegram groups for hateful content.

Text messages are checked for whether they contain offensive and hateful speech, as well as the target groups that the speech is directed against (if there are any). Voice messages are transcribed and then handled the same way as a text messages.

Memes are also checked for hate which arises due to the combination of text and an image.

Currently, the bot can only understand English language.

🎯 Key Features

The bot will:

- check texts, voice messages and memes for hate
- intervene if necessary

Group members can:

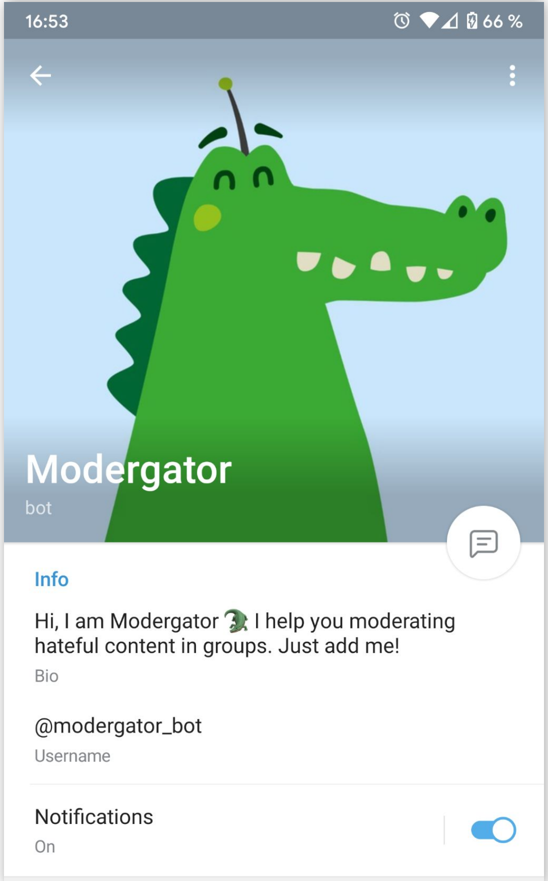
- dispute wrong classifications in a /poll
- optionally /optout of data processing (GDPR-compliant)

💡 How To Use

In order to interact with the bot, a Telegram account is needed. For instructions on how to create an account see: https://telegram.org/. To find

GitHub

Published Bot



16:53 66 %

Modergator

bot

Info

Hi, I am Modergator 🐊 I help you moderating hateful content in groups. Just add me!

Bio

@modergator_bot

Username

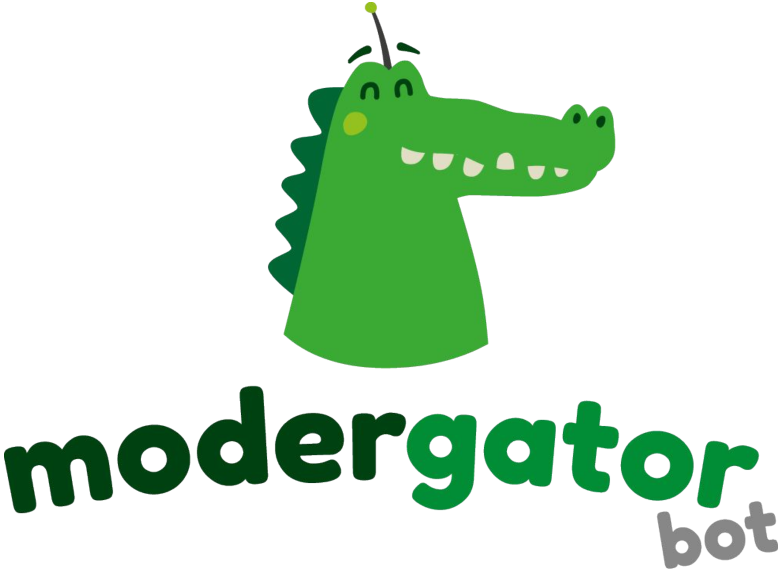
Notifications

On

August 26th, 2021

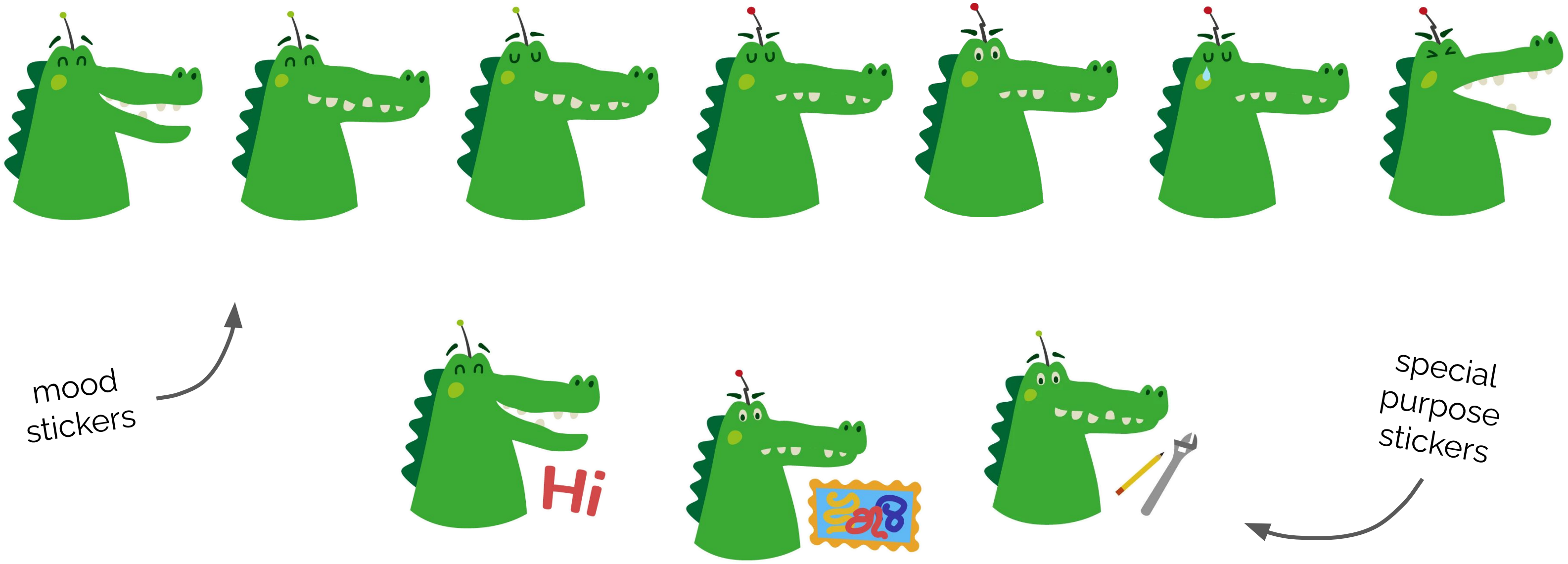
Design and UX: Name And Logo

modergator = portmanteau of moderator + alligator



Design and UX: Stickers

Custom Telegram Stickers to provide engaging user experience and create emotional connection



Technologies used

python-telegram-bot

Flask

web development, one drop at a time

python

PyTorch

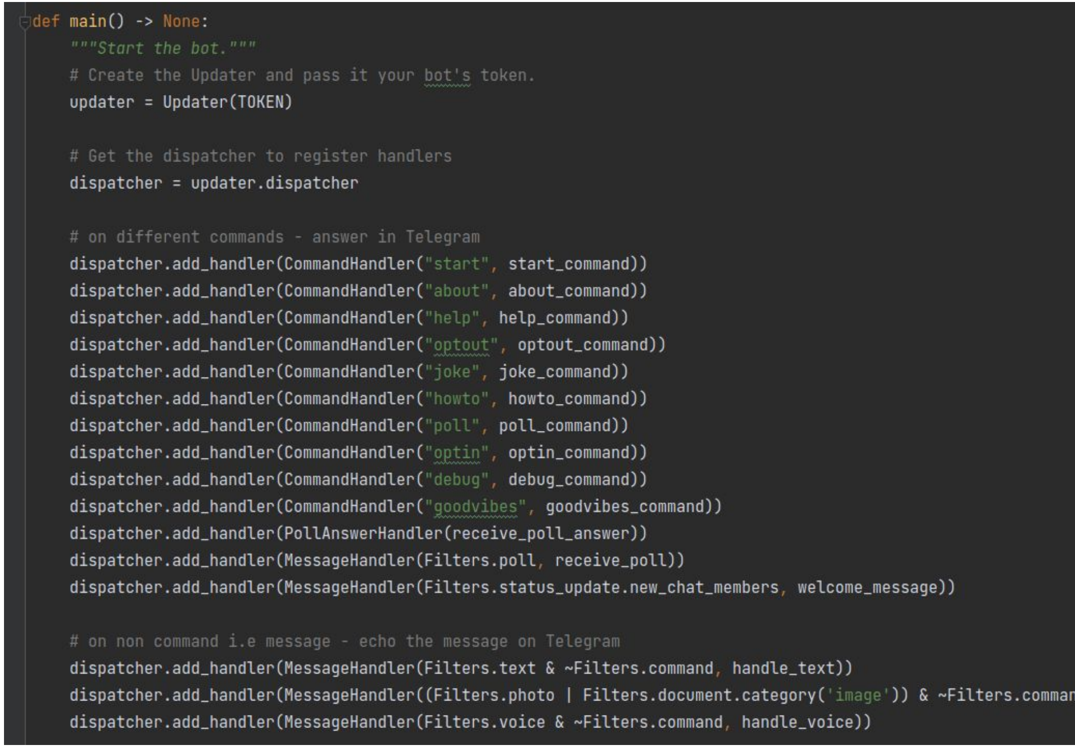
VILIO

docker

Image sources: https://commons.wikimedia.org/wiki/File:Pytorch_logo.png https://www.python.org/community/logos/ https://en.wikipedia.org/wiki/Flask_(web_framework) https://www.startpage.com/av/proxy-image?piurl=https%3A%2F%2Fencrypted-tbn0.gstatic.com%2Fimages%3Fq%3Dtbn%3AANd9GcT_gVQL4e-dJj-rujJZk2ilAHhPfDZO-MUE4W7wo4lT14NMhlyo%26s&sp=1629920522T4bf13d0774a565978e0be1ffe5ab33a2f2b8a64a395e6c373a1d23a5e582cd0b https://swagger.io/ https://github.com/python-telegram-bot/python-telegram-bot https://github.com/ https://www.docker.com/company/newsroom/media-resources https://github.com/Muennighoff/vilio

Telegram Bot

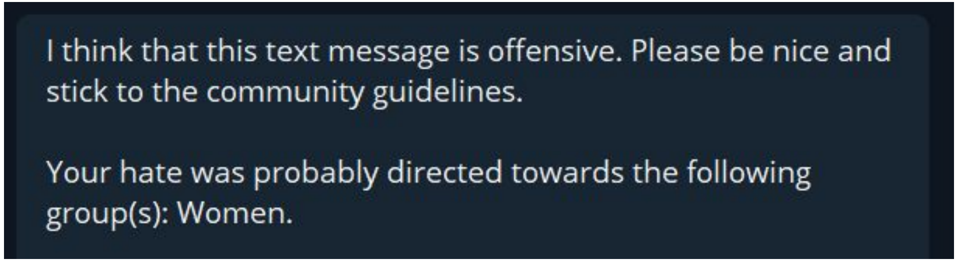
- programmed in Python
- defines all commands such as
  - /help
  - /optout
  - ...
- and accesses the other APIs
- responsible for receiving and filtering messages
- as well as sending answers



```
def main() -> None:
    """Start the bot."""
    # Create the Updater and pass it your bot's token.
    updater = Updater(TOKEN)

    # Get the dispatcher to register handlers
    dispatcher = updater.dispatcher

    # on different commands - answer in Telegram
    dispatcher.add_handler(CommandHandler("start", start_command))
    dispatcher.add_handler(CommandHandler("about", about_command))
    dispatcher.add_handler(CommandHandler("help", help_command))
    dispatcher.add_handler(CommandHandler("optout", optout_command))
    dispatcher.add_handler(CommandHandler("joke", joke_command))
    dispatcher.add_handler(CommandHandler("howto", howto_command))
    dispatcher.add_handler(CommandHandler("poll", poll_command))
    dispatcher.add_handler(CommandHandler("optin", optin_command))
    dispatcher.add_handler(CommandHandler("debug", debug_command))
    dispatcher.add_handler(CommandHandler("goodvibes", goodvibes_command))
    dispatcher.add_handler(PollAnswerHandler(receive_poll_answer))
    dispatcher.add_handler(MessageHandler(Filters.poll, receive_poll))
    dispatcher.add_handler(MessageHandler(Filters.status_update.new_chat_members, welcome_message))

    # on non command i.e message - echo the message on Telegram
    dispatcher.add_handler(MessageHandler(Filters.text & ~Filters.command, handle_text))
    dispatcher.add_handler(MessageHandler((Filters.photo | Filters.document.category('image')) & ~Filters.comman
    dispatcher.add_handler(MessageHandler(Filters.voice & ~Filters.command, handle_voice))
```

Target Detection

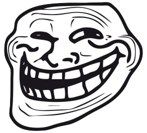
- Based on HateXplain data set (https://github.com/hate-alert/HateXplain)
- Built a model and training pipeline
- Classify input into one (or more) target groups out of 24
- Evaluation results:
  - F1: , 0.058, Precision: 0.3, Recall: 0.032
- Immense help from Fabian



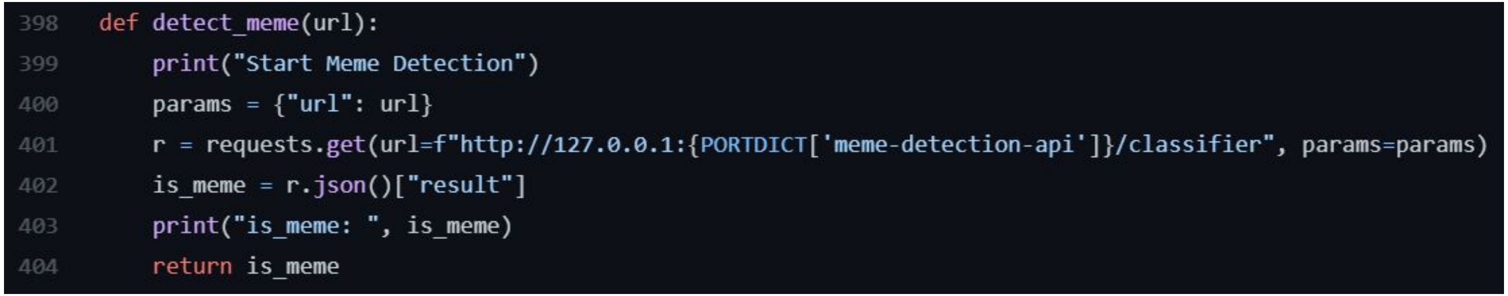
I think that this text message is offensive. Please be nice and stick to the community guidelines.

Your hate was probably directed towards the following group(s): Women.

Meme Detection



- Contribution from Niklas
- We built the meme-detection-api
- In the bot the input image is classified: is it a meme?
    - If False: do nothing with the image
    - If True: hand the image over to the ocr-api and meme-model-api to classify



```
def detect_meme(url):
    print("Start Meme Detection")
    params = {"url": url}
    r = requests.get(url=f"http://127.0.0.1:{PORTDICT['meme-detection-api']}/classifier", params=params)
    is_meme = r.json()["result"]
    print("is_meme: ", is_meme)
    return is_meme
```

Documentation

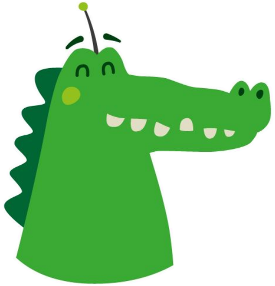
- **Swagger**
  - ○ Build with flask-apispec
  - ○ Automatically builds Swagger documentation
  - ○ Documentation can be used to test functionality
  - ○ Later shown in demo

- **Readme**
  - ○ Detailed instructions both for the end-user and anyone hosting an instance of the bot
  - ○ Includes links to files that need to be downloaded

https://commons.wikimedia.org/wiki/File:Swagger-logo.png

Demo

The End.



Thank you for your attention.