2 | Measurement

This section describes the acquisition and measuring of the data for this project. To acquire the data necessary twitch replay videos where downloaded from twitch using YouTube-DL. A script was built to do this automatically for the most popular streamers for various games. It can download multiple replays in parallel which speeds up the process because twitch throttles the downloads. Only the chat replay is kept as audio and video analysis is out of scope of this project. YouTube-DL saves the files in json format, The chat replays are converted to pickle files to use with python. This results in about 1.5GB of pickle files, so the data was compressed to save about 750MB of space. The first step is to filter all the stored chats with a preprocessor to filter the twitch emotes and remove unusual characters to prepare the data for training and classification. When analyzing the data the classifier first needs to be trained to properly cast a verdict if a certain sentence is toxic or not. This was done manually by creating a web page that shows a message from a user and allowing us to decide if it is toxic. This verdict is stored in a large SQL database to use later for classification. The flowchart of all these prosesses together is seen in Figure 2.1. All the steps in this process are described in more detail below.

2.1 Video Downloader

First of all, all the video's that are uploaded by the 100 most popular streamers are collected. This is stored in a file which holds the streamer, together with the id's of each video. This is done using the TwitchVideoIDCollector script. The most popular streamers are hardcoded, but can be extended easily in case the current dataset is not large enough for our goals.

After all the video id's are collected, the script TwitchVideoDownloader uses this list to fetch all the video's. Some filtering is already at place here. The rechat functionality was implemented by Twitch at 2016-02-23, so all that are published before this date is useless since we are mainly interested in the chat. The downloader also downloaded the video which only contains audio. Later we decided to leave the video/audio, since speech recognition is still too hard.

The TwitchVideoDownloader return the following files:

- A thumbnail
- A json file with metadata of the video
- A json file with apiinfo
- A json file with the chat
- A mp4 with the video, which only contains audio.