Project Report: Whatsapp Lens

Yahya Jabary, 11912007

Code: `https://github.com/sueszli/whatsapp-lens/`

Contents

Proposal **2**
Motivation 2
Contribution 2

Execution **3**
Synthetic Data Generation 3
Robustness Dataset 4
Parsing 4

Results **4**

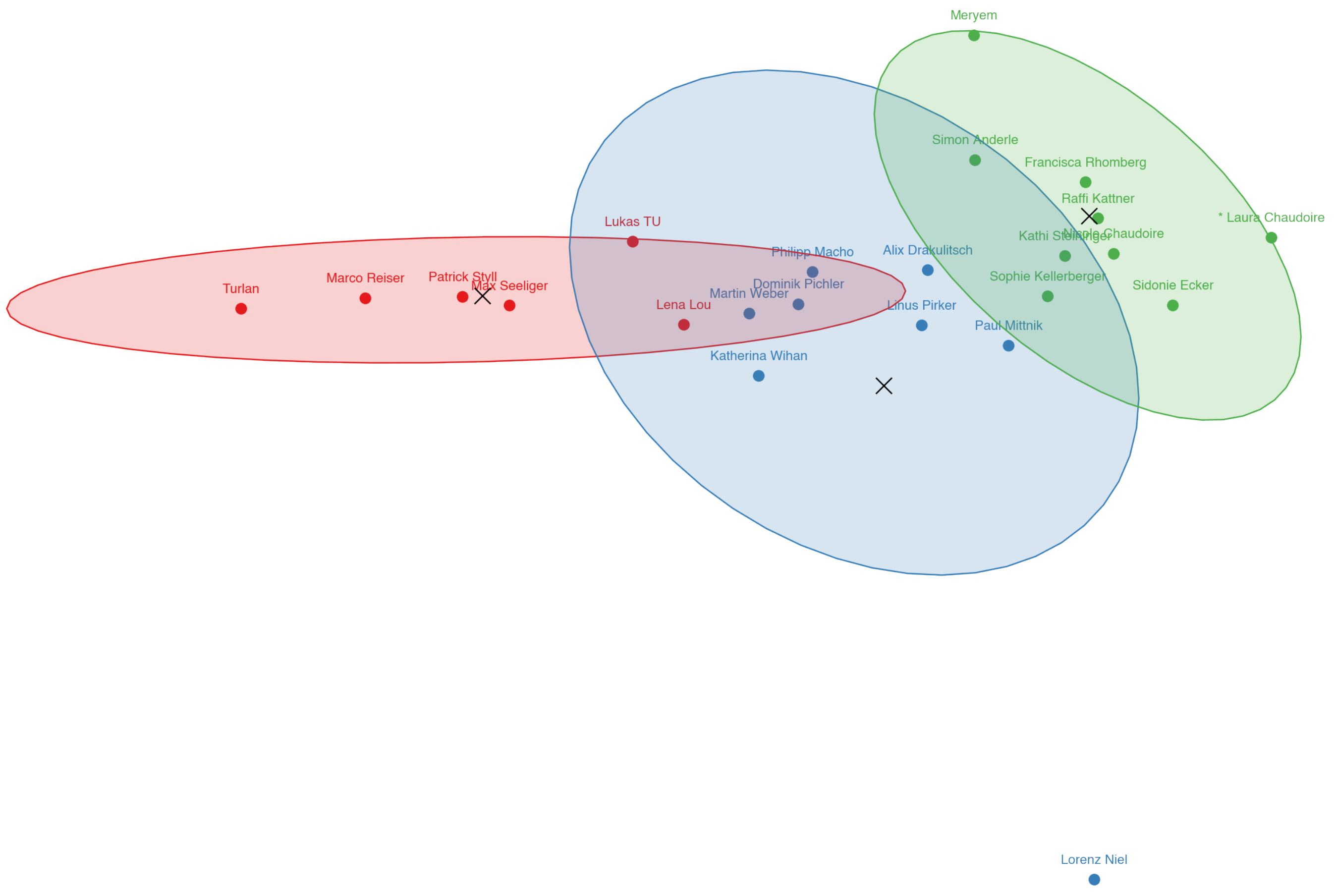

Figure 1: Latent Representation Clustering of WhatsApp Conversations via PCA

Proposal

Assignment 1 - Initiate

In this chapter we discuss the state-of-the-art in WhatsApp analytics, the motivation behind the project, the contribution of the project and a detailed plan of execution with a time estimate for each task.

Motivation

With over 2 billion users WhatsApp has reached the highest user penetration in the European Union[1] [2] [3] and has become the primary means of Internet communication in regions including the Americas, the Indian subcontinent and large parts of Europe and Africa[4].

This ubuiquity has made WhatsApp a rich source of data for social and behavioral analysis. For instance, Kaushal and Chadha[5] have surveyed various sentiment analysis techniques of WhatsApp, while Ngaleka, Uys[6] have studied m-learning and recently, as of 2023 Jannah [7] has explored the utilization of "WhatsApp Business", a Facebook-Business driven platform for customer service and marketing purposes.

However, these studies are mostly of academic interest and lack both practicality and accessibility for the average user in order to gain data driven insights from their own WhatsApp conversations. This is particularly relevant in the context of WhatsApp Business which is increasingly used for customer service and marketing purposes and where customer interactions are a valuable source of feedback and could be quantified and analyzed to improve business processes.

Analyzing WhatsApp conversations can provide valuable insights into:

- Sentiment patterns: The emotional tone of conversations by using sentiment classifiers.
- Temporal patterns: Message frequency, response times, interaction patterns and engagement metrics between participants as an indicator of relationship strength.
- Topic modeling: Extracting underlying themes in conversations to understand how discussion topics evolve over time using techniques such as Latent Dirichlet Allocation (LDA) or more modern models capable of capturing context for Q&A tasks.

Especially given recent advancements in natural language processing with the advent of transformer models, downstream tasks could also use latent representations of the chat data to train models to both classify and cluster relationships based on conversational patterns, as well as predict future interactions with high accuracy.

Contribution

One of the main challenges in applying the mentioned state-of-the-art NLP techniques to this domain however is the lack of both open datasets and accessible parsing tools for WhatsApp chat exports, given that the data comes in an unstructured and proprietary format, increasing the barrier to entry for researchers and practitioners alike.

The project category "Bring your own data" in this course lend themselves itself nicely to bridging this gap by providing a ground for the development of an end-to-end prototype for WhatsApp chat data analysis:

- (1) Data Generator: implementing synthetic data generation tools, a robustness testing dataset and a personal dataset for demonstrative purposes.
  - (1.1) Synthetic dataset: a synthetic conversational data generation tool, based on role-playing large language model agents.
  - (1.2) Robustness dataset: a custom dataset, demonstrating all edge cases and possible media to be handled by the tool.
  - (1.3) Personal dataset: a private dataset consisting of the author's own WhatsApp chat data for demonstrative purposes, not to be shared in the open-source repository.
- (2) Parser: parsing the proprietary format into a standardized CSV format using regular expressions.
- (3) Feature Engineering: validating, preprocessing and extracting key features from the data.
  - (3.1) Validation: ensuring the data is correctly parsed and standardized, including message timestamp normalization, emoji encoding, media message handling, language detection and processing.

[1] Shan, S. The battle between social giants: WeChat and WhatsApp's influence in digital marketing.

[2] WhatsApp-Website. http://blog.whatsapp.com/. Accessed 30 Oct 2024.

[3] Montag, C., Błaszkiewicz, K., Sariyska, R., Lachmann, B. Andone, I., Trendafilov, B. & Markowetz, A. (2015). Smartphone usage in the 21st century: who is active on WhatsApp?. BMC research notes, 8, 1-6.

[4] Metz, Cade (April 5, 2016). "Forget Apple vs. the FBI: WhatsApp Just Switched on Encryption for a Billion People". Wired. ISSN 1059-1028. Archived from the original on April 9, 2017. Retrieved May 13, 2016.

[5] Kaushal, R., & Chadha, R. (2023, March). A Survey of Various Sentiment Analysis Techniques of Whatsapp. In 2023 2nd International Conference for Innovation in Technology (INOCON) (pp. 1-6). IEEE.

[6] Ngaleka, A., & Uys, W. (2013, June). M-learning with whatsapp: A conversation analysis. In International Conference on e-Learning (p. 282). Academic Conferences International Limited.

[7] Jannah, R. (2023). Utilization of whatsapp business in marketing strategy to increase the number of sales through direct interaction with customers. Syntax Idea, 5(4), 488-495.

  - (3.2) Feature Extraction: extracting key features such as message frequency metrics, response time patterns, sentiment scores, topic distributions and user interaction patterns.
- (4) Analytics: performing exploratory data analysis and applying state-of-the-art NLP techniques to the data.
  - (4.1) Sentiment Analysis: training a sentiment classifier on the data to predict the emotional tone of conversations.
  - (4.2) Topic Modeling: applying LDA or more modern models to extract underlying themes in the data.
  - (4.3) Relationship Clustering: clustering relationships based on conversational patterns using latent representations of the chat data.
  - (4.4) Frequency Analysis: analyzing message frequency, response times, interaction patterns and engagement metrics between participants.
  - (4.5) Demographic Analysis: predicting user demographics based on conversational patterns.
- (5) Deployment: deploying the analytics results in a user-friendly and interactive web application.
- (6) Presentation: preparing a final report and a video presentation on YouTube.

Given our time budget of just 3 ECTS credits, equivalent to 75-81 hours of work (25-27 hours per ECTS credit), excluding the time spent on optional lectures and exercises, we constrain each task to a maximum of 15 working hours, ensuring that the project is completed within a reasonable time frame.

We plan to dedicate the largest portion of our time to the Feature Engineering and Analytics tasks, as these are the most critical components of the project, which will provide the most value to the end user in terms of insights gained from the data for our prototype.

Execution

Assignment 2 - Hacking

Synthetic Data Generation

The synthetic data generation component utilizes two instances of the TinyLlama-1.1B-Chat model to simulate a natural conversation between two personas with distinct characteristics. This approach allows for the creation of realistic WhatsApp chat data while maintaining privacy and providing a controlled environment for testing and development. The quantized 1.1B model was chosen for its capacity to run on a consumer-grade GPU such M2 Pro Metal Performance Shaders with 16GB of memory.

```
2024-10-31 03:08:00,Jane Doe,Hey Jane! How's your day going?
2024-10-31 03:10:00,Jane Doe,I had such an amazing weekend at the gym with my friends - it was great having some friendly competition while enjoying all
2024-10-31 03:13:00,John Smith,So excited for Sunday morning workout this AM (and hopefully a good night's sleep too).
2024-10-31 03:10:00,Jane Doe,How about you? Workout? Healthier lifestyle tips/strategies? Food preferences/diets suggestions? Can I tag your friends in t
```

The implementation leverages the Hugging Face Transformers library to load and utilize the language models. Each model instance is configured with specific generation parameters to ensure diverse yet coherent responses. The generation process employs a temperature of 0.9 and top-p sampling of 0.9, striking a balance between creativity and coherence in the generated text.

To maintain conversation authenticity, the system implements realistic temporal patterns through the `get_next_timestamp` function. This function introduces variable time delays between messages, with most intervals falling between 1 second and 10 minutes. Additionally, it incorporates a 1% chance of longer gaps ranging from 1 hour to 1 day, simulating natural conversation breaks.

The personas are carefully crafted to represent distinct personality types, using 2 prompts for this specific example:

- Jane Doe represents a book-loving introvert, programmed to reference literature and use book-related expressions
- John Smith embodies a fitness enthusiast, incorporating workout-related terminology and health-focused language

The conversation generation process is implemented as an iterative loop, where each model takes turns responding to the previous message. The responses are processed to ensure proper formatting and stored in WhatsApp's characteristic timestamp format (`MM/DD/YY, HH:MM`).

The implementation also includes safeguards against common issues in language model outputs, such as response truncation using regular expressions to ensure complete sentences and the removal of special tokens. A repetition penalty of 1.3 helps prevent the models from falling into repetitive patterns or loops, contributing to more natural-sounding conversations.

This approach however is not without limitations, as it occasionally produces nonsensical or off-topic responses. These issues are mitigated through manual filtering and post-processing, ensuring the generated data remains coherent and relevant. A more robust solution could involve fine-tuning the models on WhatsApp chat data to improve the quality of the generated conversations and using larger models to capture more nuanced conversational patterns.

Robustness Dataset

In addition to the synthetic data generation tool and the personal dataset, a robustness dataset was created to test the parser's ability to handle edge cases and various media types. The dataset includes a wide range of scenarios such as: phone numbers, URLs, emojis, media messages, special characters, calls, location sharing, disappearing messages and deleted messages. All of these scenarios are designed to test the parser's ability to correctly extract and encode the data into a standardized format and were handled appropriately in a post-processing step in which they were replaced with placeholders.

Parsing

The parser has been implemented as a Python function that processes WhatsApp chat export plaintext files. The script utilizes regular expressions to extract relevant information from the chat logs and converts them into a structured CSV format.

The function reads the input file line by line, using a regular expression pattern to match the timestamp, author and message content. This pattern is designed to handle the standard WhatsApp chat export format, which typically includes a timestamp, followed by the author's name and then the message content.

The expression used to match the chat log lines is: `r"(\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2})\s-\s(?:([^:]+):\s)?(.+)"`

One of the challenges addressed in the parser is handling multi-line messages. The script maintains a `current_message` dictionary to accumulate message content across multiple lines. This approach ensures that messages spanning multiple lines are correctly captured and preserved in the output.

The parser also handles server messages, which are messages generated by the WhatsApp system rather than by users. These messages are identified by the absence of an author name and are assigned the author "server" in the output CSV.

To ensure data integrity and consistency, the script includes a `validate_csv` function. This function performs several checks on the output CSV file, including verifying the correct number of columns and ensuring that timestamps are in ascending order. These validation steps are crucial for maintaining data quality and preventing errors in subsequent analysis stages.

The main execution block of the script processes all text files in a specified directory, converting each one to a CSV file. This batch processing capability allows for efficient handling of multiple chat export files, which is particularly useful for analyzing conversations across different groups or time periods.

Results

Assignment 3 - Deliver