Can You Hear the Player Experience? A Pipeline for Automated Sentiment Analysis of Player Speech

Philipp Sykownik
Entertainment Computing Group
University of Duisburg-Essen
Email: philipp.sykownik@uni-due.de

Felix Born
Entertainment Computing Group
University of Duisburg-Essen
Email: felix.born@uni-due.de

Maic Masuch
Entertainment Computing Group
University of Duisburg-Essen
Email: maic.masuch@uni-due.de

***Abstract*—This paper presents the ongoing work on a technological pipeline to automate the recording and analysis of player speech data in the evaluation of player experience. The pipeline generates a heatmap that visualizes where the players in the virtual environment have made positive, negative, and neutral statements. Moreover, the heatmap provides spatiotemporal information of in-game events, as well as interactive features like filters and modifiers. We describe how the pipeline is set up for the Unity game engine, how we implemented the sentiment classification module as well as the visualization of the tracked data. Concluding, we define goals for further development.**

***Index Terms*—Player experience evaluation, Speech analysis, Multimodal data analysis, Automated testing, Sentiment analysis, Heat maps, Data visualization**

I. Introduction

To understand how game interaction shapes player experience, researchers and developers should consider quantitative and qualitative data from within and outside the game [9], [11], [16]. Game Analytics provides techniques to automate the evaluation of player behavior and relies on vast amounts of quantitative data from within the game. However, one limitation of such data is that it mostly allows for indirect inferences about the player opinion and emotion that shape the player experience. In contrast, think-aloud testing and interviews are methods that can directly assess such information. They heavily rely on verbal communication and generate qualitative data. Although they suffer from certain methodological limitations from a researcher's perspective, practitioners value them as indispensable tools of player experience evaluation, in particular when combined with other methods [2], [10].

Overcoming methodological limitations by evolving methods and technologies that are used to assess, analyze, and communicate multimodal data is one of the current challenges in games user research [5]. As one approach for overcoming this challenge, the combination of artificial intelligence technologies and games user research methods is recently advocated in both domains [5], [18]. Our work contributes to this field of research by sharing the current state of a technological architecture, that automates the assessment, analysis, and presentation of player opinion. It utilizes sentiment analysis of player speech and combines it with spatiotemporal in-game data. Thereby, the current implementation is based on freely available software components to lower barriers for applying the pipeline in small-scale scenarios.

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

II. Related Work

Our approach lines up to earlier and more recent work in game user research that deals with the automation and integration of multimodal data from inside and outside the game.

TRUE [9] is a pioneering example that outlines a transferable and adaptable architecture for the orchestration of different information channels, that focuses on the automated collection, merging and presentation of data from within as well as outside the game. Later work further elaborated on ways to visualize various types of data in an integrated way. For instance, *Biometric Storyboards* combine biometric data with game events and player comments that are obtained in retrospective interviews [12]. Iterations of this approach focused on visualizing player movement paths, biometric information, and player comments on top of game world screenshots [13]. Moreover, various levels of detail for the visualizations were recently evaluated [16].

The tools G-player [3] and Vixen [6] highlight the importance of interactivity of visualizations. G-player generates heatmaps from data that it loads from a file or database. By modifying data queries, researchers can adapt the visualization and, for example, produce heat maps that visualize intersections of several event types that represent compound events. Vixen is a plug-in for the game engine Unity that generates interactive and customizable 3D visualizations within the game engine and tracks telemetry and observational data.

Although TRUE already demonstrates the value of automated assessment of opinion data during gameplay, later work describes how researchers capture, transcribe, and integrate opinion data with other data manually and retrospectively [12], [13], [16]. By automating these steps, our work aims to complement existing and future efforts in this field.

III. Architecture Components

Based on reviewed related work, we define a set of system requirements, that guides our work:

1) The architecture should be easily extendable.

2) The system components must enable automation of assessment, analysis, and visualization of data from inside and outside the game.
3) Components for data presentation should link spatiotemporal and speech data.
4) Visualization components should incorporate interactivity features.

We translated these requirements into an architecture, that covers the processes of data tracking, sentiment analysis, and data presentation (see Figure 1). In the following, we describe the proof of concept implementations for the core modules of these processes. They include the architecture's setup within the source code of the application that is tested, the sentiment analysis, and an application for data visualization. Currently, we realized these modules as a package for the Unity game engine, two Python classifiers, and a NodeJS web server that hosts a NoSQL MongoDB database, and a Typescript web application based on the React framework.

A. Data Tracking

We implemented the tracking component as an asset package for the Unity game engine. Thus, the package is usable in any Unity project. It provides functionality for the recording of player speech and the tracking of player positions and includes interfaces for realizing the logging of custom in-game events. The basic configuration for speech and position tracking does not require any programming. However, the integration of events and areas tags from the specific game at hand requires source code editing of the own project. Once implemented, Unity constantly buffers the specified microphones' audio streams at run time. Coupled with Microsoft Windows' built-in dictation recognizer, the tracking system can automatically detect passages of spoken words and converts them into text. Each passage is then extracted from the audio buffer and stored as an audio file into a speech data unit together with the corresponding text file. Position data units contain a position vector, an area tag, and a timestamp. Event data units specify the event name, actor, target, cause, and timestamp. The application uploads these data units to a server in a previously configured interval. This server runs the other core modules of the architecture, the sentiment classifiers, and the web application.

B. Sentiment Analysis

The core of automating speech analysis in the proposed architecture is sentiment classification of the text-based speech data units that are generated during gameplay. Our goal was to implement a pipeline, that automatically analyzes whether verbal statements of German players during gameplay are positive, neutral, or negative. Therefore, we realized the classification core module in two implementations, a lexicon-based approach and a machine learning approach based on an artificial neural network. Both classifiers are implemented as Python scripts that run along with the database and web application on the server that receives data units from the Unity application. They regularly check for updates in the database and classify newly uploaded speech data by attaching sentiment labels to them. Subsequently, they save the classified player speech units in the database.

The lexicon-based classifier calculates the overall sentiment of a player statement based on its syntactic structure and its positive, negative, and neutral constituents. This information is extracted from the text files of each speech data unit. For syntactic analysis of German statements, our classifier operates on the v2.2 TIGER corpus [1], that provides part-of-speech values and lemmas of German sentences. Sentiment tags are assigned based on the GermanPolarityClues resource [17].

To identify valuable approaches for future development, we also implemented a long short-term memory neural network utilizing Keras and Tensorflow. Additionally to the long short-term memory structure [8] the network consists of an embedding layer, a random dropout layer, and a softmax classifier, that calculates the probabilities for the input vector as being classified as either positive, negative, or neutral. Thereby, the input vector is the numerical representation of an input text file, that is constructed based on a bag-of-words model. The training and validation data sets were built from resources containing prelabeled German Twitter posts [4] [14], because we considered their syntax and length to be similar to expressions during gameplay. In sum, the network was trained and validated on nearly 80.000 posts and reached categorical accuracy of 64 percent.

C. Data Presentation

A web application acts as an access point to the in-game and speech data. The application offers basic navigation functionality to find and select data of interest from any project that has uploaded data to the architecture's database. The core of this application is a sentiment-based heatmap that shows where players have made positive, negative, and neutral statements in the virtual environment. Therefore, the application retrieves a 2D map from the level data in the database. Figure 2 depicts how the tool generates a grid overlay above the map. The grid indicates the sentiment within areas of the game world. Thereby, a grid cell's color corresponds to the sentiment of the statements that occurred within the area of the grid (i.e., red = negative, green = positive, grey hatched = neutral, red and green hatched = diverse). Additionally, icons represent individual speech and in-game events. Further, the map indicates individual player paths.

As depicted in Figure 2, the application provides several filters and modifiers to select which data the map displays. The user can i.a., show and hide speech data of individual players, or filter out certain sentiment types and in-game events. Beneath the speech map, the user can determine that the map only shows data up to a certain point in time of the gameplay. Further, he can increase or decrease the level

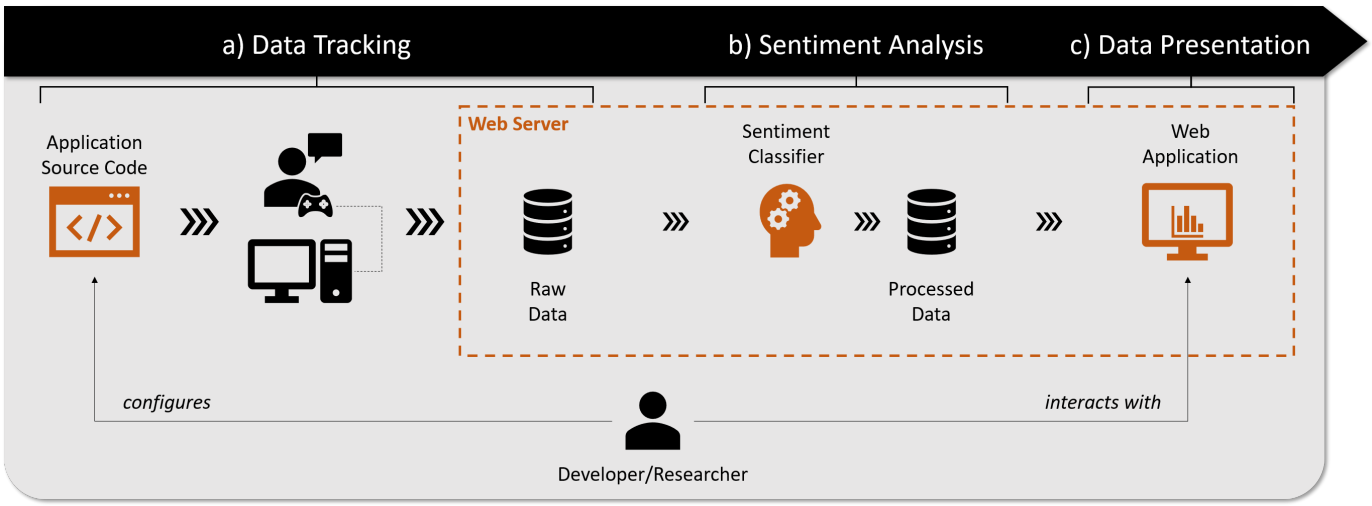

Fig. 1. Overview of the architecture. Core modules that are described in this paper are colored orange.

of detail of the sentiment grid. By selecting a grid cell, the user can access details of the speech and in-game events that occurred in the selected area (see the right panel in Figure 2). For example, player statements are displayed as text and can be played back based on the stored audio file. Moreover, the sentiment label can be adjusted manually.

IV. Next Steps

Though we did not use the current pipeline in a user study yet, we assume that it will be highly valuable in the future. However, the current core modules already automatically provide valuable information on player communication activity. Therefore, we plan to use the pipeline in upcoming user studies in the context of multiplayer virtual reality.

Currently, both classifiers are limited in their accuracy, but they suffice in demonstrating how to integrate speech data into established approaches of automated evaluation seamlessly. The next significant step in the development is the creation of an adequate training data set to achieve an acceptable classification accuracy [14]. Another recent work related to games and sentiment classification of game-related Twitter posts notes the need for domain-specific training data [7]. We consider transcribed audio tracks from Let's Play videos as a potential approach for this.

Thanks to the modularity of the architecture, the pipeline can easily be supplemented by other components for data analysis and visualization. For instance, the recorded audio files can also be used for voice analysis, to extract emotional dispositions or identify speakers in multiplayer settings, where only one microphone is available.

Although we think a sentiment-based heatmap suffices as an initial approach that integrates player speech and in-game data, we are aware that it needs further improvement and should be supplemented by other types of visualization and data. Thereby, suitable visualizations should be developed individually for games with various concepts of player representation or less spatiotemporal variability. However, guidelines like *overview first, details on demand* [15], as well as specific criteria of quality [16] should guide future work.
Currently, the heatmap is based solely on the position vectors of players and does not utilize the possibility to define custom area tags in the Unity package. These, however, could be used for other types of visualizations, that benefit from position data of lower granularity (e.g., bar charts presenting averaged sentiments for different areas). Other interesting approaches that should be considered in this context are individual and aggregated color-coded player trajectories [16] that represent the sentiment along player paths, and the integration of gameplay footage or screenshots at the time of a comment to better comprehend player comments [9], [12]. Moreover, approaches for visualizing the dynamics of verbal communication between multiple players caused by in-game events in multiplayer scenarios are another interesting direction for future work.

V. Conclusion

Research and development of games benefit not only from advances in game technology but also from advances in technologies like artificial intelligence. Also, access to such technologies is becoming easier, ultimately enabling everyone to use them effectively for innovative domain-specific projects. We present such a project and share insight into the current proof of concept implementation of a pipeline for automated player speech analysis that features machine learning [1]. The value of automated extraction of player opinion from communication during the game is measured by the effort required to manually extract it and the information content of this data about the gaming experience. Thus, our work contributes to overcoming current challenges in games user research. We

[1]Access to project: https://www.ecg.uni-due.de/showcase/overview.html