Python を用いた Zoom チャットログ分析アプリケーション

2023/06/09
Kaien 代々木 訓練生
楠元 宏幸

目次

１．背景

■目的

現在私は、Kaien という発達障害者向けの支援事業所に通所している。そこでは、私の通う就労移行支援や学生向けの支援プログラムであるガクプロなどいくつかのコースが存在し、それぞれで Zoom を用いたオンライン講座が行われている。そこで講座を担当する講師から「オンライン講座に参加した聴講者の Zoom チャットログから発言を抽出し分析を行いたい」という依頼を受けた。

本アプリケーションは、txt 形式で保存された Zoom チャットログを発言者と発言内容などに分解し、自然言語処理や発言数のプロットなどを行うことで、チャットから参加者の反応を分析できるようにすることを目的とした。

■開発環境

○OS

・Windows 11

○プログラミング言語

・Python 3.10(注 1

○IDE

・PyCharm 2022.3 (Community Edition)

○標準ライブラリ(抜粋)

・tkinter
・pathlib
・re
・json　　等

○サードパーティ・ライブラリ

・tkinterDnD
・GiNZA
・nlplot(注 1
・pandas
・plotly.offline
・matplotlib

注 1)nlplot は python3.11 に対応しておらず、pip install ができなかったため 3.10 で仮想環境を構築した。

■開発期間

・調査	2023/03/15(水)～2023/03/20(月)	8 時間
・プロトタイプ開発	2023/03/22(水)～2023/04/03(月)	20 時間
・追加調査	2023/04/04(火)～2023/04/07(金)	9 時間
・本開発	2023/04/10(月)～2023/06/02(金)	86 時間
		計 123 時間程度

2. 成果物紹介

■チャットログ処理

保存したチャットログのひとつをメモ帳で開いたものの一部を図 1 に示す。Zoom のチャットログは示した通りマークアップされていないプレーンテキストで保存されるため、発言内容とそれ以外を判別する必要があった。

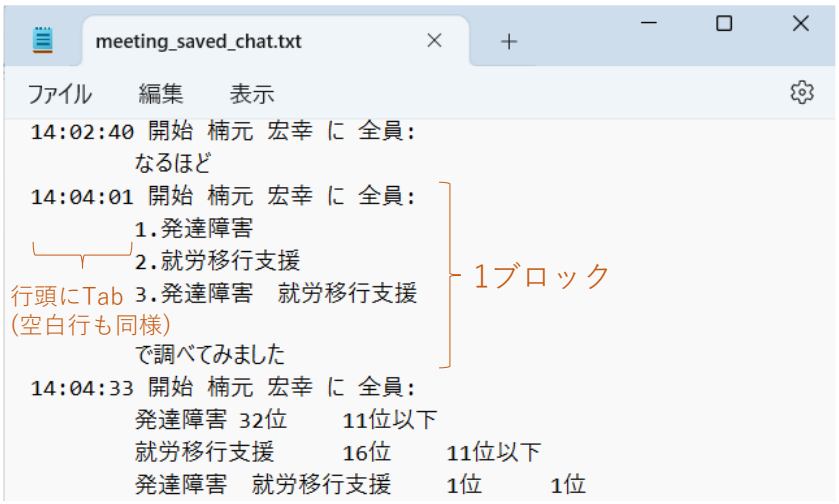

図 1. チャットログ例

これは 1 行ずつ抜き出した際に本文は行頭が Tab 文字(“¥t”)となることに着目し判別した。行頭を確認し Tab 文字でなければそれをヘッダーとみなし、Tab 文字であれば本文とみなして直前のヘッダーと関連付ける処理を行い、次のヘッダーまでを 1 ブロックとした。

ヘッダー部分は時刻が必ず 8 文字になること、「_開始_」「_に_全員:」(_は半角スペース)の部分は定型文であることから、それらは文字数によって前後から除去することで発言時刻と発言者名を抽出することとした。なおダイレクトメッセージは「_に_全員:」とはならないが、それらは分析対象外とし、ヘッダー末尾が一致しない場合はブロックそのものを排除するフラグ処理を行った。

ログファイルを 1 行ずつ読み取ってブロックごとに分割したのち、時刻・発言・本文などを pandas の DataFrame 型に格納し、以後この DataFrame を用いて処理を進めていくこととした。

■GUI 作成

GUI は tkinter を用いて作成した。ファイル読み込み画面と解析後の操作画面をそれぞれ図 2, 3 に示す。ここでは同一の講座を複数回に渡って一度に読み込む機能が求められた。

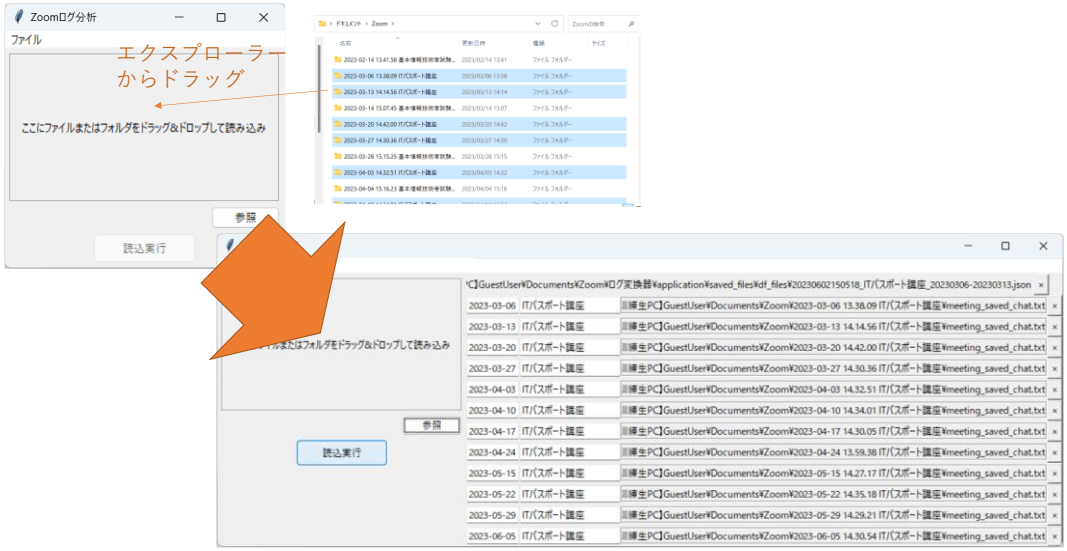

図 2. ファイル読み込み画面

アプリケーションを起動すると図 2 の左上のような画面が開く。Zoom から保存したログは回ごとにフォルダを分けて保存されるため tkinter の filedialog では問題があった(ディレクトリを選択するダイアログのメソッドには複数選択オプションが存在しないため)。そこで tkinterDnD というサードパーティのライブラリを導入し、エクスプローラーで複数選択したフォルダをドラッグ&ドロップすることで読み込みファイルを選択できるようにした。また、ここでは生チャットログだけでなく、一度分析処理を行った DataFrame を json で保存したものも読み込めるようにしている。

読み込み処理を行ったあとは図 3 のような画面になり、分析処理の設定を指定して出力を行う。

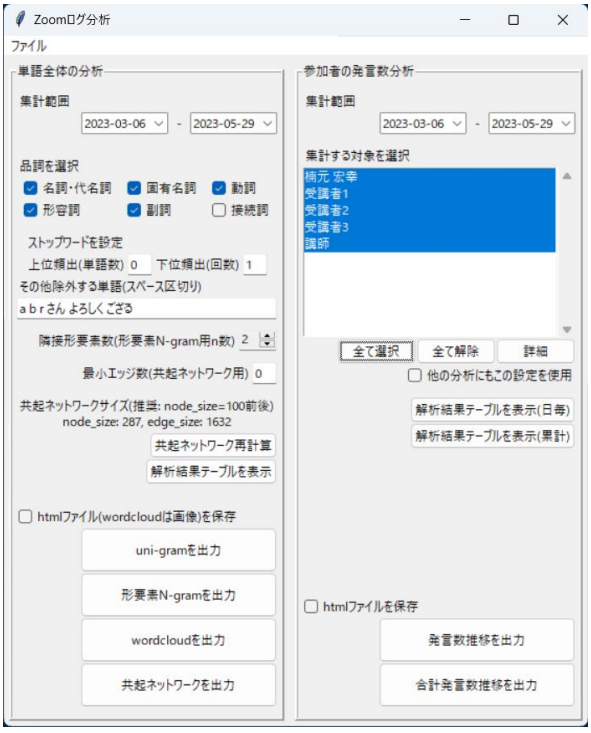

図 3. 分析操作画面

■自然言語処理・可視化

前項の図 3 で示した画面の左側で行えることを紹介する。ここではチャットでの発言に自然言語処理を行い、単語ごとに分解してその回数や繋がりについて分析できる。

自然言語処理には GiNZA(参考 1 を用いた。GiNZA は日本語用の自然言語処理ライブラリであり、入力した文を単語に分解し、それぞれの品詞・レンマ(語彙素の基本形)などを返す。チャットの本文それぞれについて、GiNZA で処理し選択した品詞のものだけを抽出したリストを作成する。

処理したデータを用いた可視化については nlplot(参考 2 を用いた。nlplot では上記の処理を施した DataFrame からいくつかの種類のグラフを作成することができるが、ここでは N-gram バーチャート(n=1 だと単純に単語ごとの出現数をグラフ化できる)・ワードクラウド・共起ネットワークを出力できるようにした。

例として、共起ネットワーク・ワードクラウドをそれぞれ図 4, 5 に、その際の出力設定を図 6 に示す。

参考 1) https://megagonlabs.github.io/ginza/

参考 2) https://www.takapy.work/entry/2020/05/17/192947

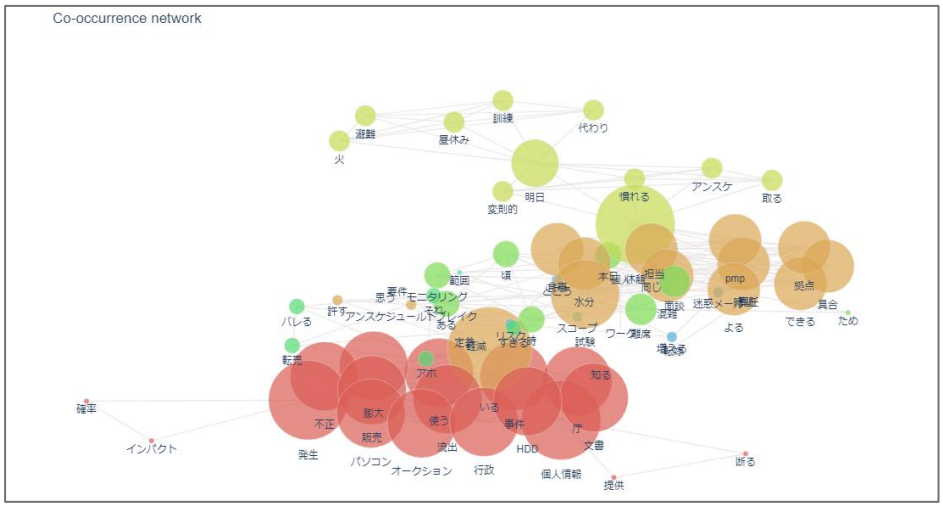

図 4. 共起ネットワーク

図 5. ワードクラウド

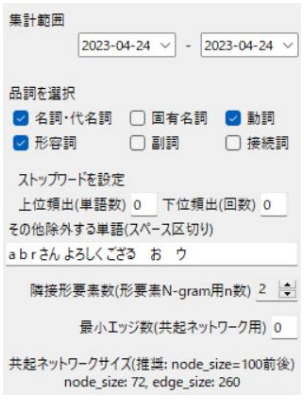

図 6. 出力設定の例

■集計対象選択・発言数分析

図 3 の画面右側では参加者それぞれについて回ごとの発言回数やその累計を折れ線グラフで出力できる。発言者の名前の中から集計する対象をリストボックスで選択し、抜き出しての比較や明らかに回数が違うなどでノイズになる参加者の除外などを行えるようにした。累計グラフの例を図 7 に示す。出力前に対象のチェックを外した「受講者 2, 受講者 3」以外の参加者について出力されていることがわかる。

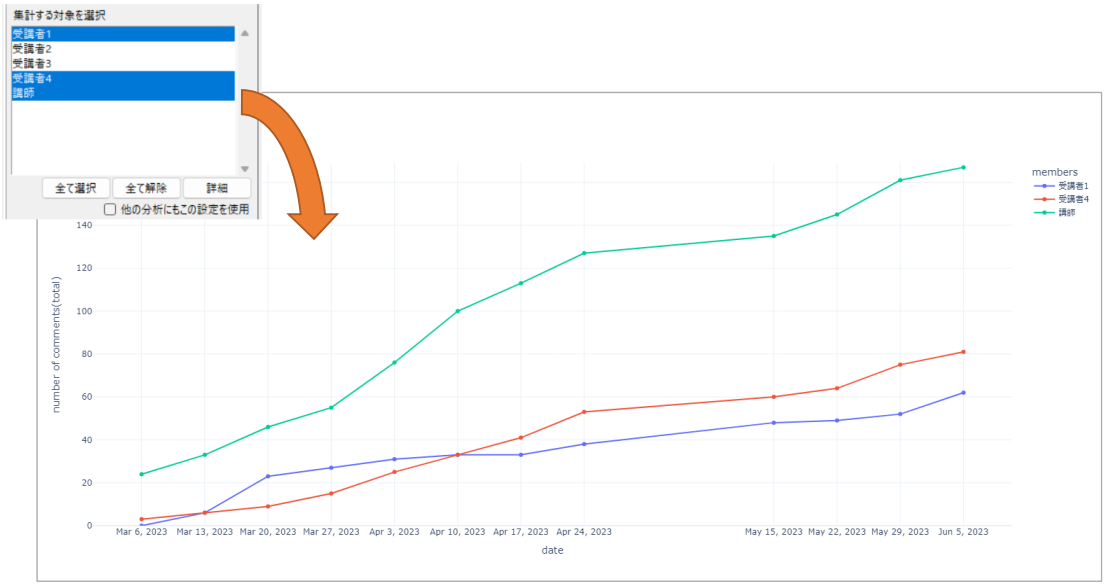

図 7. 発言回数の累計

また、集計対象に関して詳細設定を行えるようにした。これによって表示名を変更することや、また回によってログイン名が異なる参加者(例:『楠元 宏幸』と『楠元宏幸』の表記ゆれ)などを手動で 1 つにまとめることができる。その例を図 8 に示す。

ここでの対象選択は前項の集計に反映させることもできる。例えば、発言回数が著しく多い、特定の発言を繰り返すなど分析の際にノイズとなる参加者がいた場合、一時的に除外した結果を見ることができる。

図 8 集計対象の詳細設定

3. おわりに

以上で本アプリケーションの紹介を終了し、この項では感想と今後の展望について記す。

感想は、プログラミング(特にオブジェクト指向という考え方)にそこそこ慣れてきたかというものだった。見様見真似の VBA からプログラムというものに触れ始めて 1 年、Python に触ってから半年程度、本作で 4 つめの自作アプリケーションとなるが、オブジェクトや関数定義といったことすら無縁だった頃から考えるとだいぶ手が慣れてきたように思えた。ただ、作るものが複雑になっていくに従いその構想が膨れ上がり纏めるのに時間がかかったり、コーディングそのものが楽しいためか概要程度の設計で手をつけ始めたりしたため、要件定義や設計といった作業にもやり方を見つけ慣れていかなければならないと感じた。

今後について、設定機能など追加で入れられるように設計しているが未実装の機能があるため、それらを加えることで完成版としたいと考えている。現時点ではオフラインで使用する前提のアプリケーションしか作っておらず Web アプリなどは全くの素人なので、機会があれば勉強してみたい。