



Talk to the City と広聴AIの歴史

 NISHIO Hirokazu
2025年3月20日 00:25

...

安野たかひろ氏が立ち上げた「デジタル民主主義2030」プロジェクトは、2025年3月16日に新しいブロードリスニングの道具「広聴AI」をオープンソース公開しました。この「広聴AI」と「Talk to the City」の関係について歴史を遡って解説します。

デジタル民主主義2030プロジェクトの一貫として『広聴AI』のリポジトリを公開しました。<https://t.co/WyQgyIvDxp>

Talk to the Cityを参考に、日本の自治体や政治家の実務に合わせた機能改善を進めています。

・機能例開発者以外でも扱いやすいような機能 (CSVアップロードなど)... pic.twitter.com/VJFc0l2yyG

— 安野貴博 @ 新刊『1%の革命』発売中！ (@takahiroanno)

March 16, 2025

Talk to the City (TTTC) は、アメリカのNPOであるAI Objectives Instituteによって開発されたAIツールです。

Talk to the City 開発の時系列

初期開発 (2023年10月以前)

- Talk to the Cityは当初、Python/Next.jsベースのアプリケーションとして開発されました
- このアプリケーションは、CSV形式のコメントデータから、インタラクティブな散佈図(Scatter Plot)での可視化レポートを生成するように設計されました
- これは後に「Scatter」と呼ばれるようになりました。この記事でもそう呼ぶことにします

オープンソースリリース (2023年10月)

- **2023年10月24日:** 後の「Scatter」がtalk-to-the-city-reportsリポジトリでオープンソースとして公開されました
 - [コミット 0debc1a](#): "first open-source commit" by Bruno Marnette

並行開発 (～12月)

- この期間中、新しい実装 (Turboと呼ぶことにします) が別途開発されていました
- 元の実装 (Scatter) も引き続き強化されていました

モノレボ統合 (2023年12月)

- **2023年12月23日:** リポジトリの大規模な再編成が行われ、talk-to-the-city-reportsリポジトリにScatterとTurboという名前でそれぞれのソースコードが置かれました

軽量JS版公開開始 (2024年3月)

- **2024年3月:** ttic-light-jsリポジトリが作成され、Turboの軽量JavaScriptの実装が始まりました

Scatter実装

Scatter実装は、Talk to the Cityの最初のバージョンであり、メインの分析処理はPythonで実装されています。この実装は、CSVファイルからコメントを取り込み、Pythonで分析し、Next.jsでインタラクティブな散布図での可視化レポートを静的HTMLとして生成します。

この散布図機能はScatter実装に特有のもので、Turbo実装にはこの機能が含まれていません。2023年12月23日のコミット（12308ba）で、既存のファイルがscatterフォルダに移動され、モノレポ内での整理が行われました。Scatter実装はこのタイミングまで、Talk to the Cityの唯一の公開実装でした。

技術スタック

- Python 3.10+: バックエンドの処理とAI処理
- Next.js: フロントエンドのインタラクティブなレポート
- OpenAI API: 議論の抽出とクラスタリング

ワークフロー

1. CSVファイルからコメントを読み込む
2. OpenAI APIを使用して主要な議論を抽出
3. 抽出された議論をクラスタリング
4. クラスターにラベルと要約を生成
5. 必要に応じて複数の言語に翻訳
6. インタラクティブなHTMLレポートを生成

Turbo実装

Turboは、Talk to the Cityの新しい実装であり、JavaScript/TypeScriptベースのアプリケーションです。ブラウザでノードを接続してグラフを構築し、それを処理する設計になっています。

アーキテクチャ

Turbo実装は、依存関係グラフベースのデータおよび計算モデルを使用しています。このモデルでは、ノードが方向性のあるエッジで接続されています。ノードとエッジはパイプラインを形成し、一部のノードはデータを提供し、他のノードは計算ステップを提供します。

計算はトポロジカルソートを行い、ノードの出力が下流のノードの入力に渡されます。各ステップでは、各ノードの「compute」関数が上流の入力データで呼び出され、すべてのノードが計算されるまで続きます。

Turboの計算には「run」と「load」の2つのモードがあります。「run」はパイプライン作成者がパイプラインを積極的に実行する場合、「load」はレポートページが視聴者によってロードされる場合に呼び出されます。

Turbo実装は、2023年12月23日のモノレポ統合の一環として「turbo」フォルダに追加されました。

技術スタック

- TypeScript/JavaScript: 主要な開発言語
- Svelte: UIフレームワーク
- Firebase: ホスティングとデータ永続化
- Svelteflow: グラフビューの実装
- OpenAI API: 自然言語処理

モノレポ統合

2023年12月23日、Talk to the Cityリポジトリは大幅な再編成を行い、ScatterとTurboの両方の実装を含むモノレポとなりました。

統合プロセス

1. 元のコードベースが「scatter」フォルダに移動されました
 - [コミット 12308ba](#): "Move existing files to scatter folder" by Orb
2. Turbo実装が別のリポジトリからマージされました
 - [コミット 408988d](#): "moved content to turbo folder" by Orb
 - [コミット fc4f2b9](#): "Merge remote-tracking branch 'ttc-turbo/master'" by Orb
3. 構造を説明する共通のREADMEが追加されました
 - [コミット e956e91](#): "added common readme" by Orb

4. 変更はプルリクエスト #28 を通じてマージされました

- [コミット 702e964](#): "Merge pull request #28 from AIObjectives/merge-repos" by Onurbon

この再編成により、両方の実装が「Talk to the City」と呼ばれるようになりましたが、それらは異なるコードベースと異なるアーキテクチャおよび機能を持つ別個のプロジェクトです。

統合後の構造

統合後、リポジトリは以下の構造になりました：

```
talk-to-the-city-reports/
├── README.md           # 共通のREADME
├── scatter/            # Python/Next.jsベースの元の実装
│   ├── pipeline/      # Python処理パイプライン
│   ├── next-app/      # Next.jsフロントエンド
│   └── ...
└── turbo/             # JS/TSベースの新しい実装
    ├── src/           # Svelteアプリケーション
    └── ...
```

tttc-light-js リポジトリ

2024年3月、[tttc-light-js](#)という新しいリポジトリが作成されました。これはTalk to the Cityの軽量JavaScriptの実装です。

アーキテクチャ

tttc-light-jsは、以下の3つの主要コンポーネントで構成されています：

1. **NextJSクライアント**: ユーザーインターフェースとレポート表示
2. **Expressサーバー**: バックエンドAPI
3. **Python FastAPIサーバー**: LLM呼び出し処理

ワークフロー

1. CSVファイルまたはGoogleスプレッドシートをNextJSクライアントまたはExpressAPIを通じて送信
2. バックエンドアプリがLLMを使用してデータを解析
3. 解析されたデータがGoogle Cloud Storage/バケットに保存
4. レポートはNextJSクライアントを通じて閲覧可能

tttc-light-jsは、Talk to the Cityの機能を維持しながら、より軽量な実装を提供することを目的として開発が行われています。

Turboとの関係

tttc-light-jsは、基本的にTurbo実装に近い思想を持っています。JavaScript/TypeScriptベースであり、インタラクティブなレポート生成を重視している点がTurboと共通しています。しかし、Turboの複雑な依存関係グラフベースのパイプライン処理を採用せず、より単純化されたNextJS/Express/FastAPIの3層構造を採用しています。

tttc-light-jsには「turboToSchema」という変換ユーティリティが含まれており、Turboのデータ形式をtttc-light-js形式に変換する機能があります。これはTurboのデータ構造を利用しつつも、グラフベースの複雑な計算モデルは採用しなかったことを示しています。Turboの複雑なグラフベースモデルを簡素化することで、より理解しやすく、メンテナンスしやすいアーキテクチャを実現しています。

(2025-10-20追記: 2015-08-31にこのtttc-light-jsを今後の開発の中心とする旨の説明とセットでtalk-to-the-city-reportsリポジトリはread-onlyのpublic archiveになりました)

広聴AI開発の時系列

2024年5月から7月にかけて、安野たかひろ氏とそのチームがTalk to the CityのScatter実装を東京都知事選挙に活用し、「ブロードリスニング」という言葉と散布図での可視化が日本国内で多くの人に知られました。筆者はこのときのチームの一員でした。このときはScatterをごくわずかな修正を入れて使っていました。このノウハウと修正差分は下記の記事でpatch形式で公開しました。

その後、2024年10月にかけて、安野チームは日本テレビと連携して衆議院選挙の可視化にTalk to the Cityを活用しました。(放送は2024-10-15~28)

このときのソースコードは[日本テレビさんのリポジトリ](#)で公開されています。筆者はこのときの開発には参加していません(JAPAN CHOICEの世論地図を作る方が先約だったので...)

この時に色々な機能追加を行いました。下記記事: "今回のプロジェクトで、ブロードリスニングが圧倒的進化"の節をご覧ください。また今後も継続的に開発していく方針だったことから、安野チーム自身として[リポジトリ](#)を公開しました。このリポジトリを見ると、整理の過程でTurboが削除されているのがわかりますね。

その後11月ごろにかけて、東京都とGovTech東京と連携して、2050年の東京を見据えた長期戦略策定のためにブロードリスニングを使うプロジェクトが行われました。(2024-11-22~2025-01-31)

筆者はこのプロジェクトに関わっていました。このプロジェクトの議論の中で、テレビ番組の限られた時間で視聴者向けに解説をする場合と、時間をかけて分析して戦略策定などを行う場合でのニーズの違いが明らかになってきました。具体的には階層的に掘り下げる機能や濃い意見クラスタを抽出する機能などです。このプロジェクトの時には独立したスクリプトを使ったり、Talk to the Cityを繰り返し使ったりしていましたが、本体機能としてあるべきだという議論になり、それらを機能追加した新しいシステムの開発が始まりました。

シン東京2050（仮称）策定に向けたご意見大募集 みんなでつくる「シン東京2050」プロジェクト|11月|...

都庁総合ホームページのシン東京2050（仮称）策定に向けたご意見大募集 みんなでつくる「シン東京2050」プロジェクト(www.metro.tokyo.lg.jp)



1月に安野たかひる氏は「デジタル民主主義2030」プロジェクトを立ち上げました。詳しい話は下記の記事で解説されています。筆者の独断で雑に言えば「安野チームがクローズドで開発して事後的にソースコードを公開する形」から「誰でも参加できるオープンな場で開発されて、リアルタイムで公開されるオープンな開発スタイル」へと変化していこうということです。

そして、2025年3月16日に、新しいブロードリスニングの道具「広聴AI」や大規模熟議支援システム「いどばた」などをオープンソース公開しました。わずか1日で開発コミュニティとしてのSlackワークスペースも参加者250人を超え、たくさんのPR(ソースコード改善提案)が行われています。

チームで作っていたデジタル民主主義2030関連のソフトウェアを昨日オープンソースにしたばかりなのですが、たくさんPRをいただいていて大変ありがたい（数えた感じ、もともとのメンバー以外のPRで既に10件ぐらいマージされている・・・！） pic.twitter.com/Ryn1aiNZAh

— 安野貴博 @ 新刊『1%の革命』発売中！ (@takahiroanno)

March 17, 2025

広聴AIではノンエンジニア向けにCSVアップロードだけで利用できる管理画面と閲覧画面が追加され、また分析機能には元々Scatterが持っていた意見の全体像を散布図で表示する機能に加え、密度の濃い意見集団を表示する機能と階層的に掘り下げる機能が追加されました。

最後に

広聴AIはTalk to the CityのScatter実装を起点に、日本国内で多様な活用を経て進化してきました。アメリカで生まれたTalk to the Cityの種が、日本という新たな土壌に根付き、芽を出しました。開発プロセスもクローズドな環境からオープンコミュニティへと移行し、より多くの人が参加できる形へと変化しました。広聴AIの芽は、今後さまざまな人々の手によって大きく育っていくことでしょう。今後もこうしたツールは発展し、さらに多様な声を可視化し、人々のコミュニケーションを改善し、世界をより良くしていくことの助けになることでしょう。

このお祭りにあなたも参加しませんか？

開発に貢献いただける方も募集しています。一緒にデジタル民主主義を作っていきませんか？

下記URLよりSlackのぞいてみてください
<https://t.co/C4pvbyGIDC>

— 安野貴博 @ 新刊『1%の革命』発売中！ (@takahiroanno)

March 16, 2025

● ● ●