GitHub Copilot は何でもできる。

プログラミング用 AI で、動画制作

石崎 充良

自己紹介

石崎 充良 (@mishi_cs)

C# Tokyo コミュニティ管理メンバー

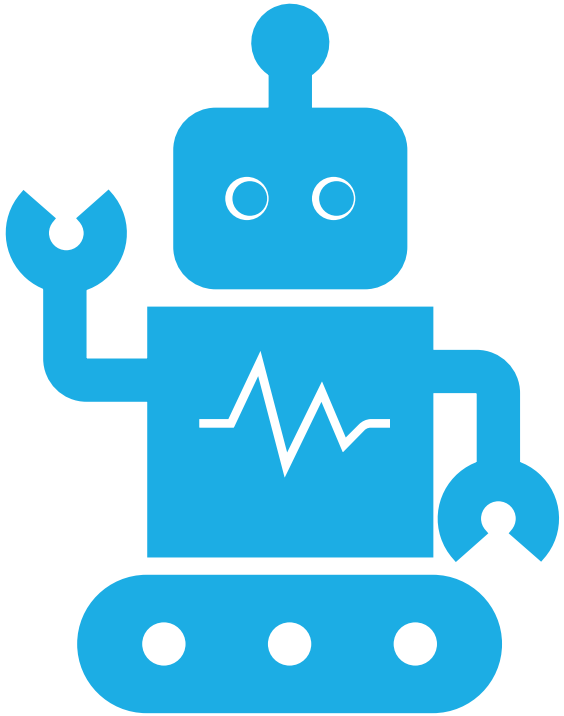
GitHub：

https://github.com/m-ishizaki

blog：

https://rksoftware.hatenablog.com/

「何でも」とは



- GitHub Copilot は AI
- プログラミング向けのプロダクト
- 以前は「プログラミング専用」だった
- 今は一般的な作業にもかなり素直に応えてくれる
- 規約も「プログラムだけ」と書かれていない様子（次ページ）

規約上「プログラム限定」はあるか

結論：プログラム専用に使うべしという規約類は見当たらなかった

	公式ドキュメント	説明
①	GitHub Terms of Service	全体の基本規約
②	Terms for Additional Products and Features	追加製品向け
③	#J AI Features	AI機能のデータ取り扱い

- ①の「C. Acceptable Use」はおおむね「違法行為に使うの禁止」。プログラム限定の記述はなし
- Copilot は②にセクションがあるが内容は薄く、プラン別のリンクが載るだけ
- そのリンク先は③へ。オプトアウトすれば学習には使われず、出力は自己責任で使う旨のみ

https://docs.github.com/en/site-policy/github-terms/github-terms-of-service#c-acceptable-use
https://docs.github.com/en/site-policy/github-terms/github-terms-for-additional-products-and-features#github-copilot
https://docs.github.com/en/site-policy/github-terms/github-terms-of-service#j-ai-features-training-and-your-data

実際にこれから行う

プログラムそのものではない作業を行う

具体的には、プログラム系 YouTube 動画を作る作業

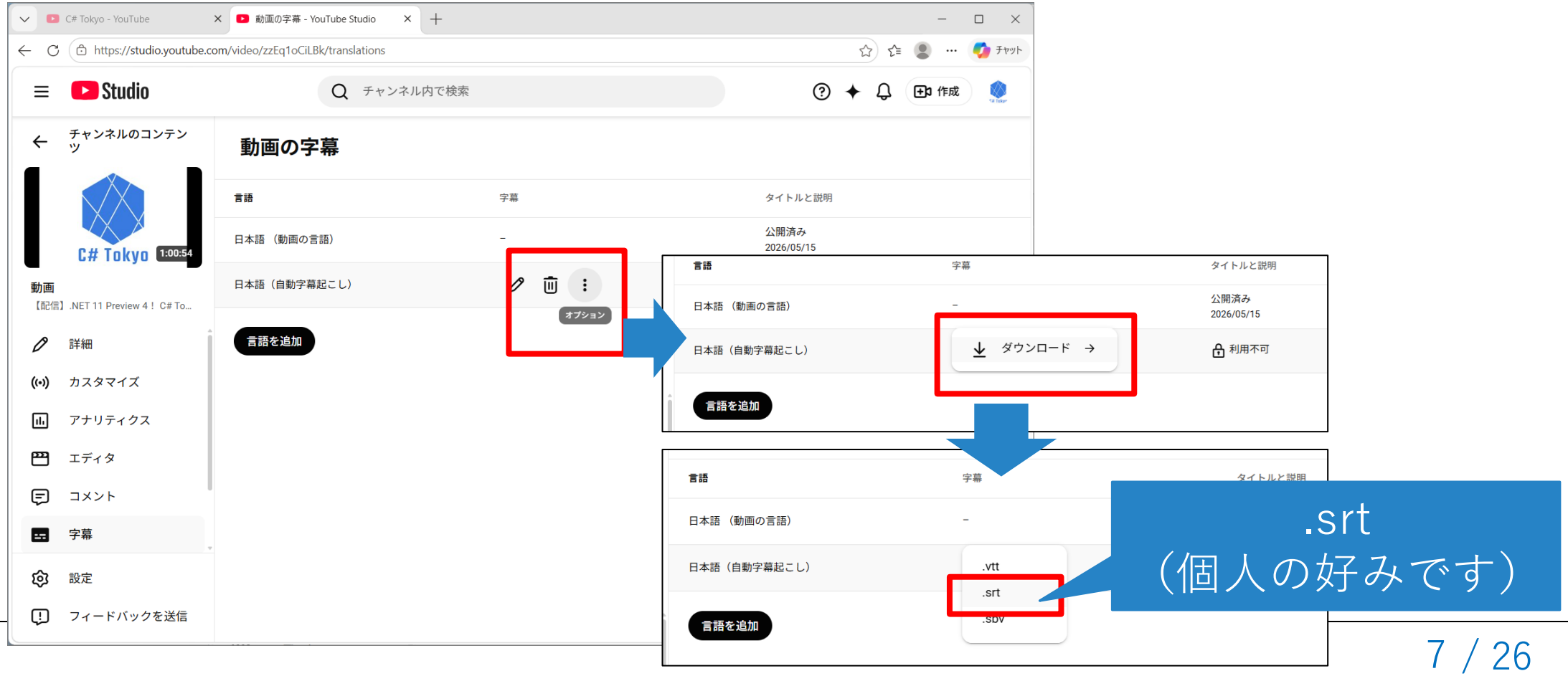
C# コミュニティ「C# Tokyo」の動画制作を行います

その素晴らしい動画制作を GitHub Copilot で！

まずは文字起こしデータを入手

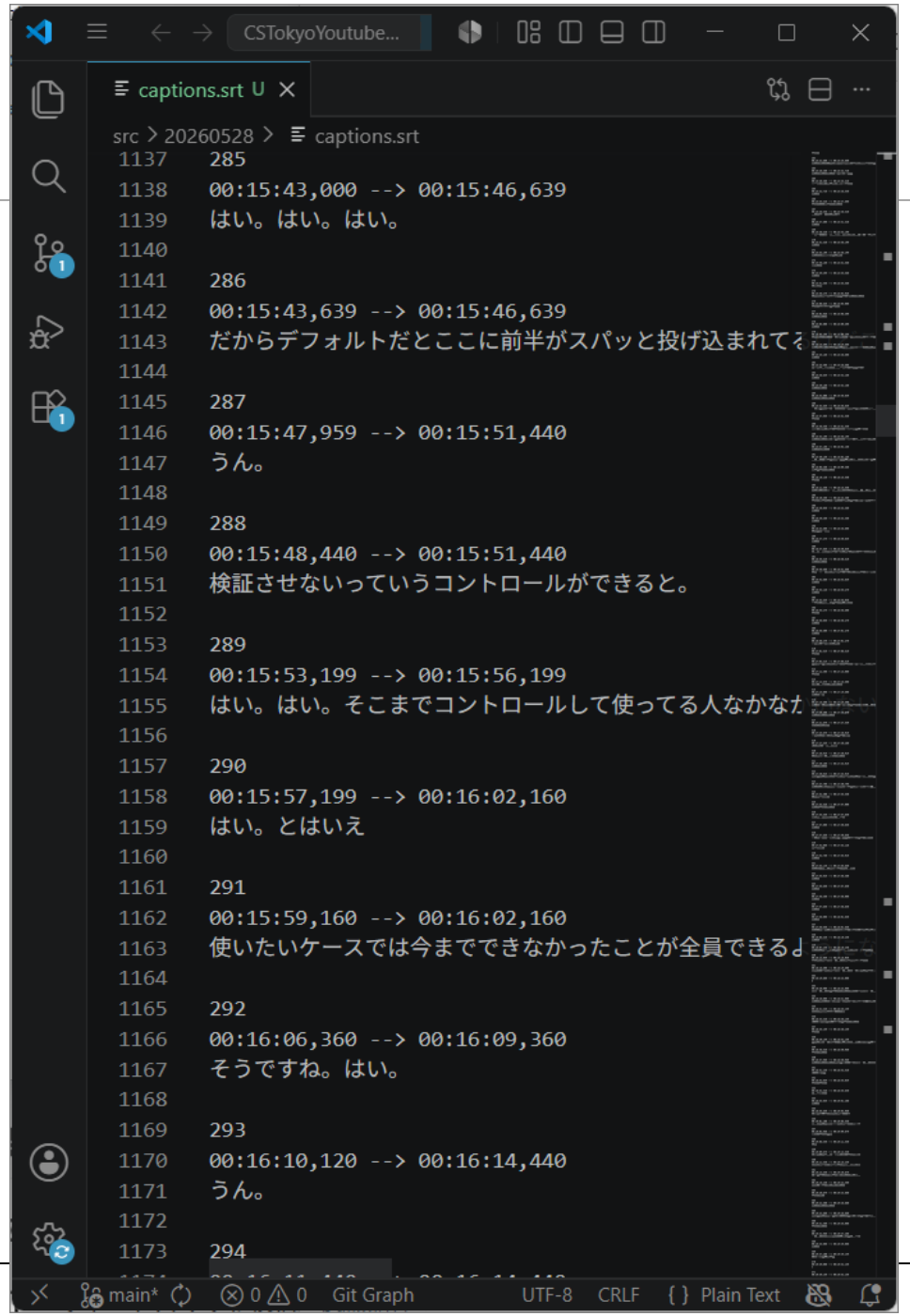
文字起こしデータを入手する

- C# Tokyo はイベントはライブ配信で限定公開し、を編集し後日動画として公開している
- ライブ配信時の文字起こしが YouTube 上に残っている
- 「字幕」内の 日本語（自動字幕起こし）からデータが取得できる



文字起こしデータ

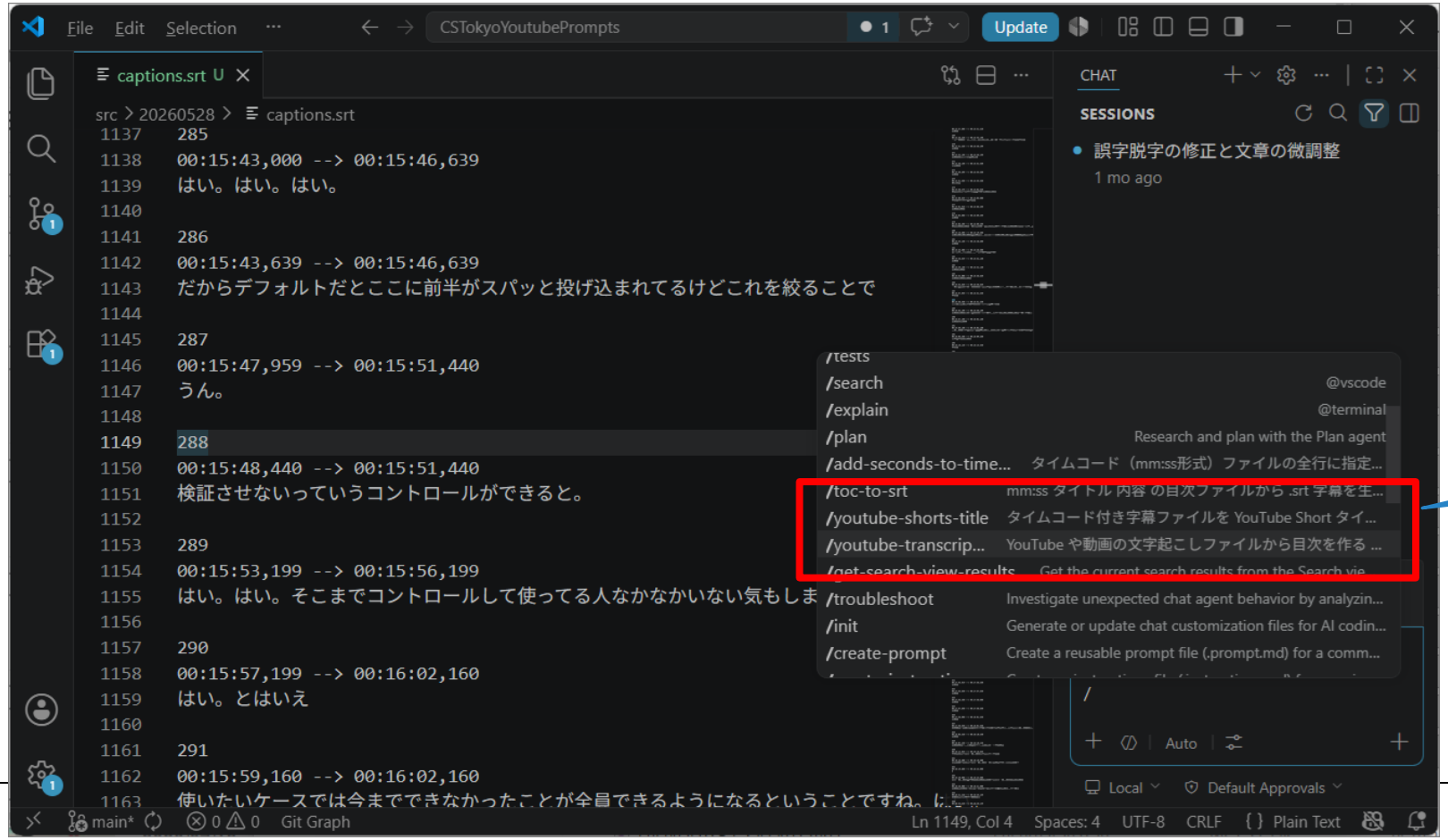
- 形式は中身を確認しやすい .srt を選択
  (個人の好みです)



GitHub Copilot で加工する

文字起こしを Skill で加工

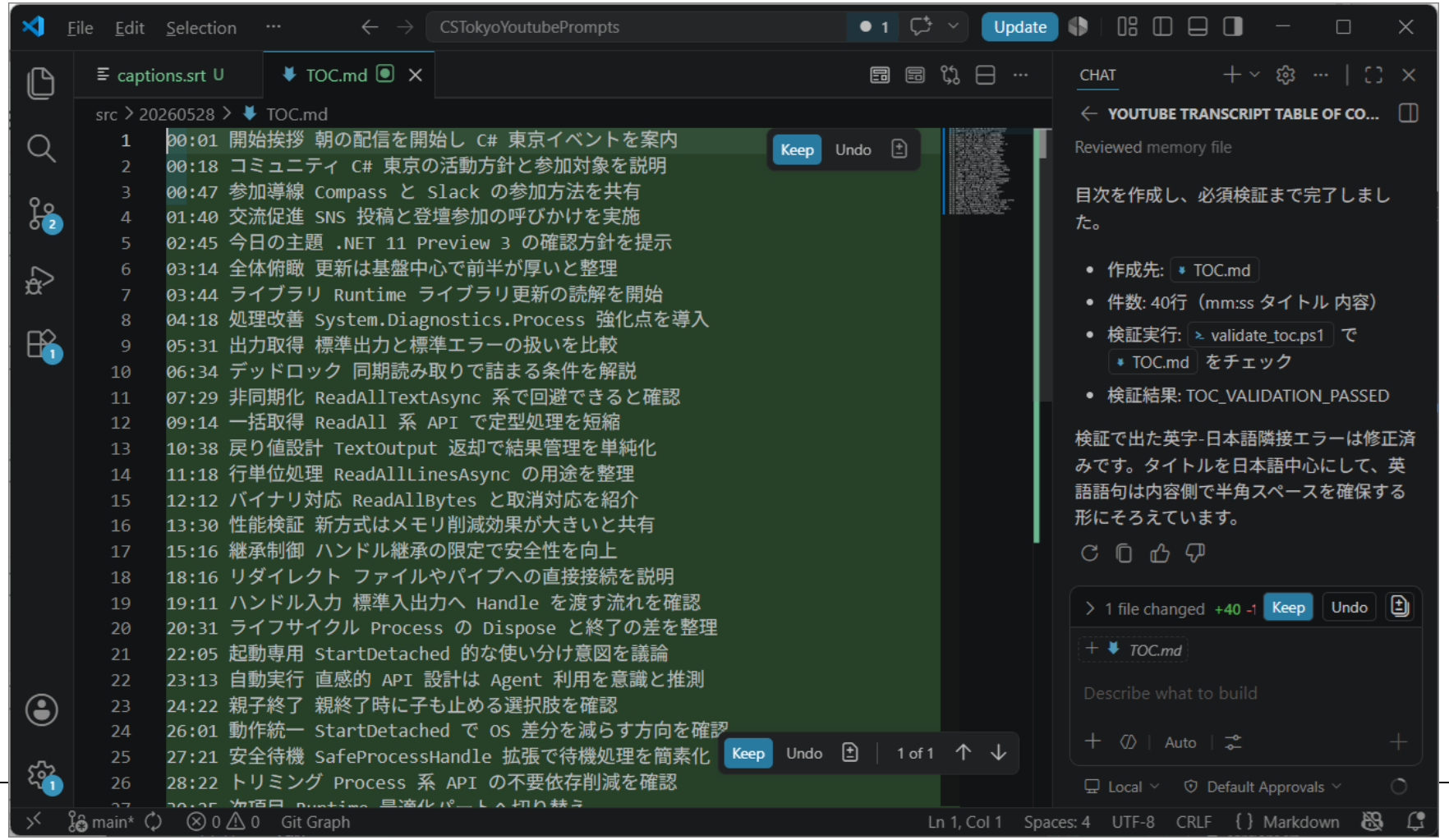
- 文字起こしデータの加工にも GitHub Copilot は使える
- プログラミング内容なら固有名詞・用語の補正も得意
- ※自動字幕は用語のカタカナ化や不正確さがある



加工用の Skill を用意している

文字起こしを Skill で加工

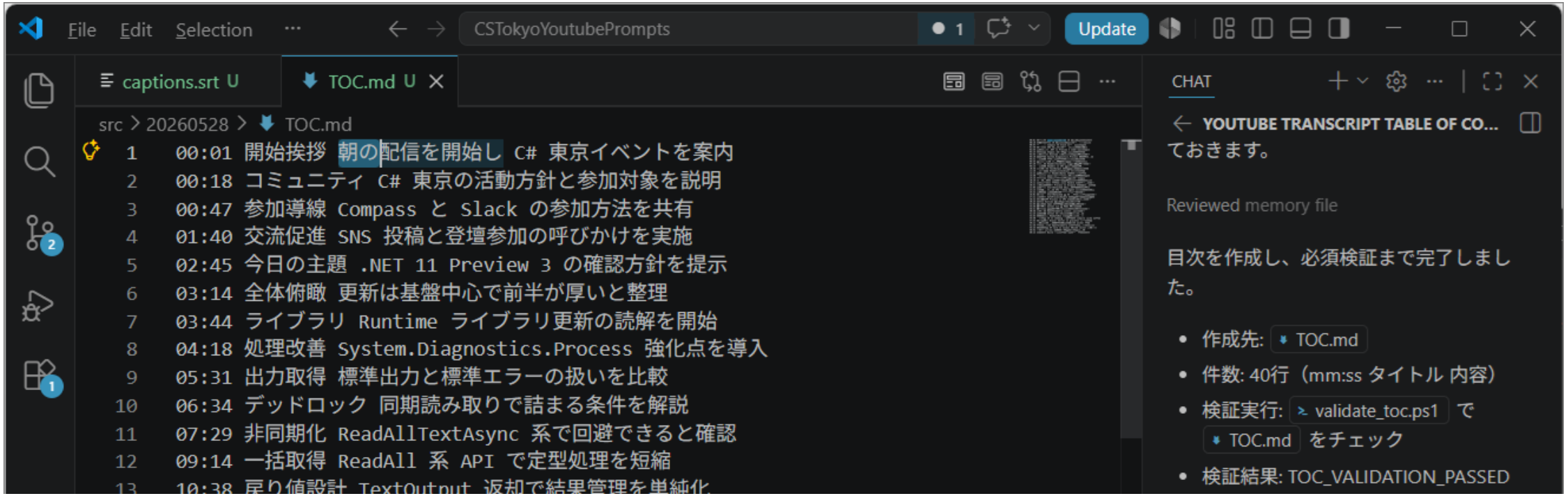
- VS Code で開いたまま加工できてとても自然
- Skill を作れば一発。数分ごとの見出しを自動生成



課題：タイトルがわかりにくい

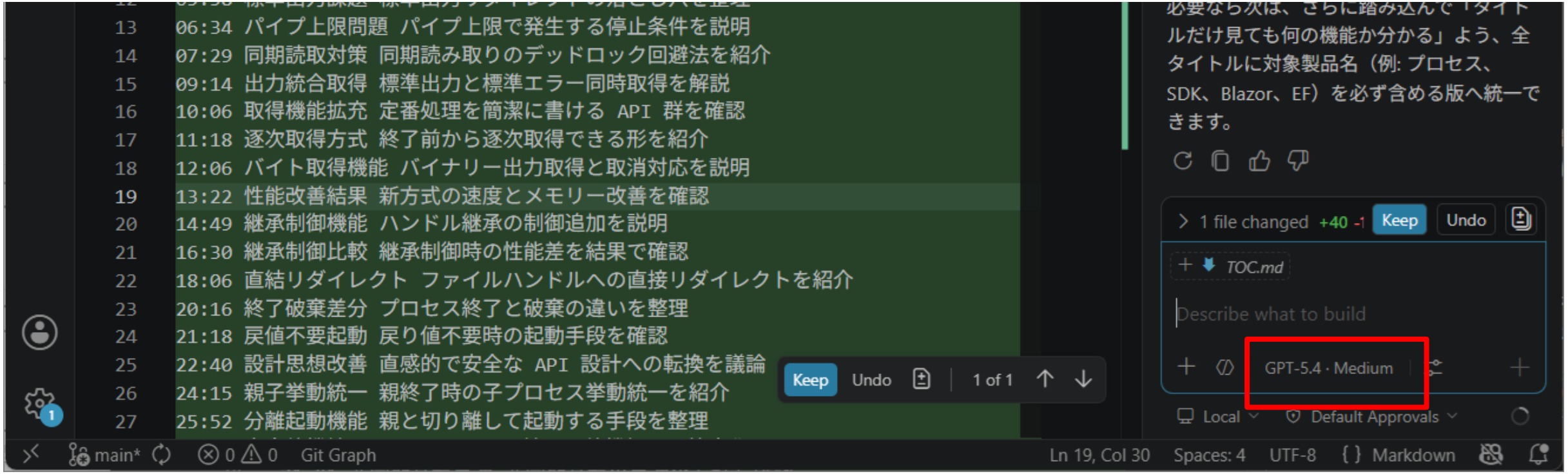
- 生成された見出しが少しわかりにくかった
- 「何についての話か」が抜け落ちていた
- 例：「処理改善」だけでは何の処理かわからない （動画には複数の技術要素があるのに主語が不明）

  ……追加プロンプトで再挑戦しても、まだいまいち



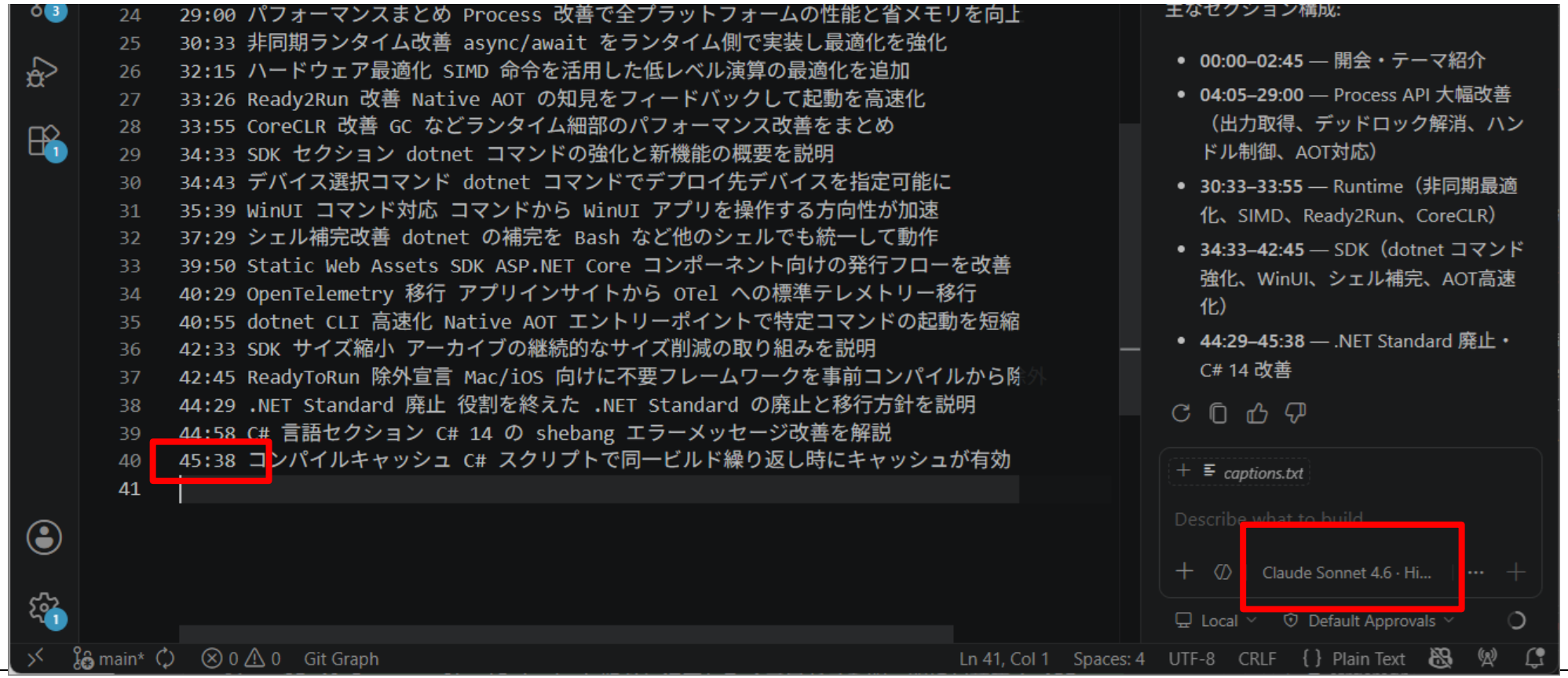
モデルを変更してみる（GPT 系）

- GPT 系の少し良いモデルに変えてみる
- 見出しの精度は改善されるか
- 結果は……無念。まだいまいちだった



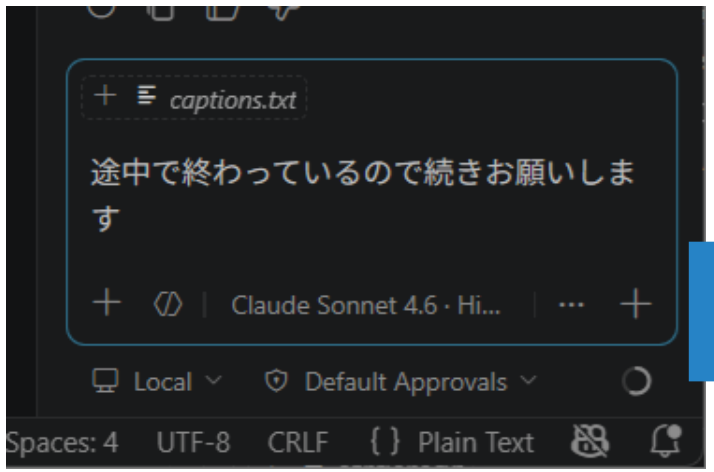
Claude 系で大きく改善

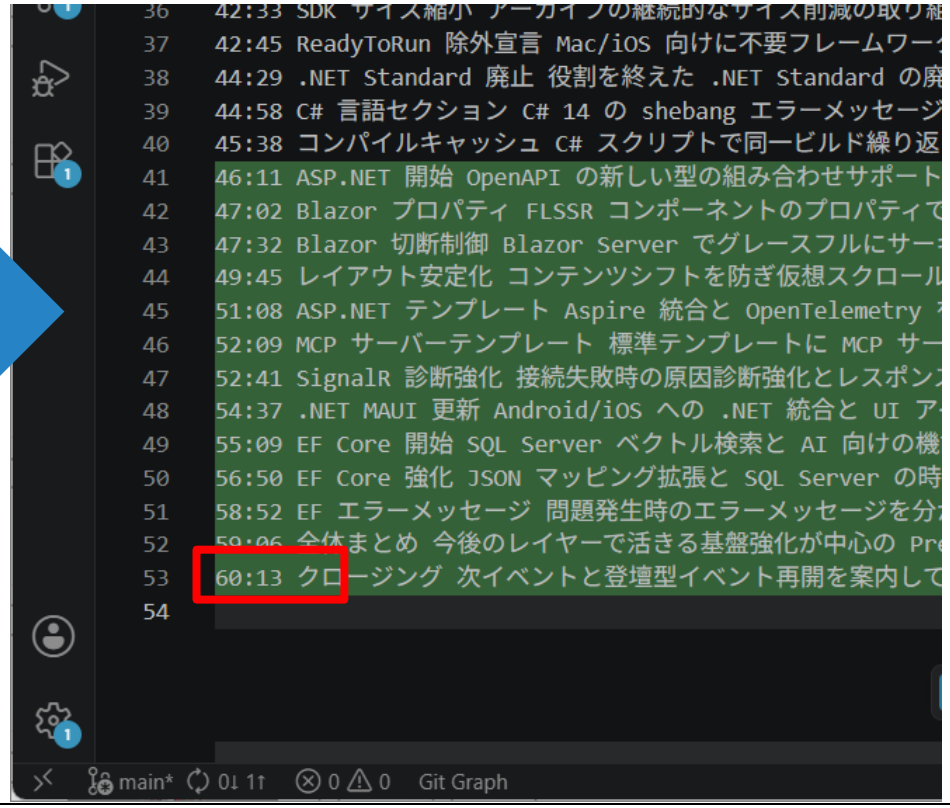
- さらにモデルを変更（Claude 系）、 かなりいい仕上がりになった
- これがプログラミング用 AI の力！
- しかし、最後までやり切れず（60分中45分あたりまで）（対処は次ページ）



アドリブ力も必要

- 「続きもお願い」と頼めば最後まで生成してくれた
- このあたりは AI の使い手側にもアドリブ力が必要
- 途中で止まっても、追加依頼でカバーできる





タイムテーブルを調整する

タイムスタンプを一括調整

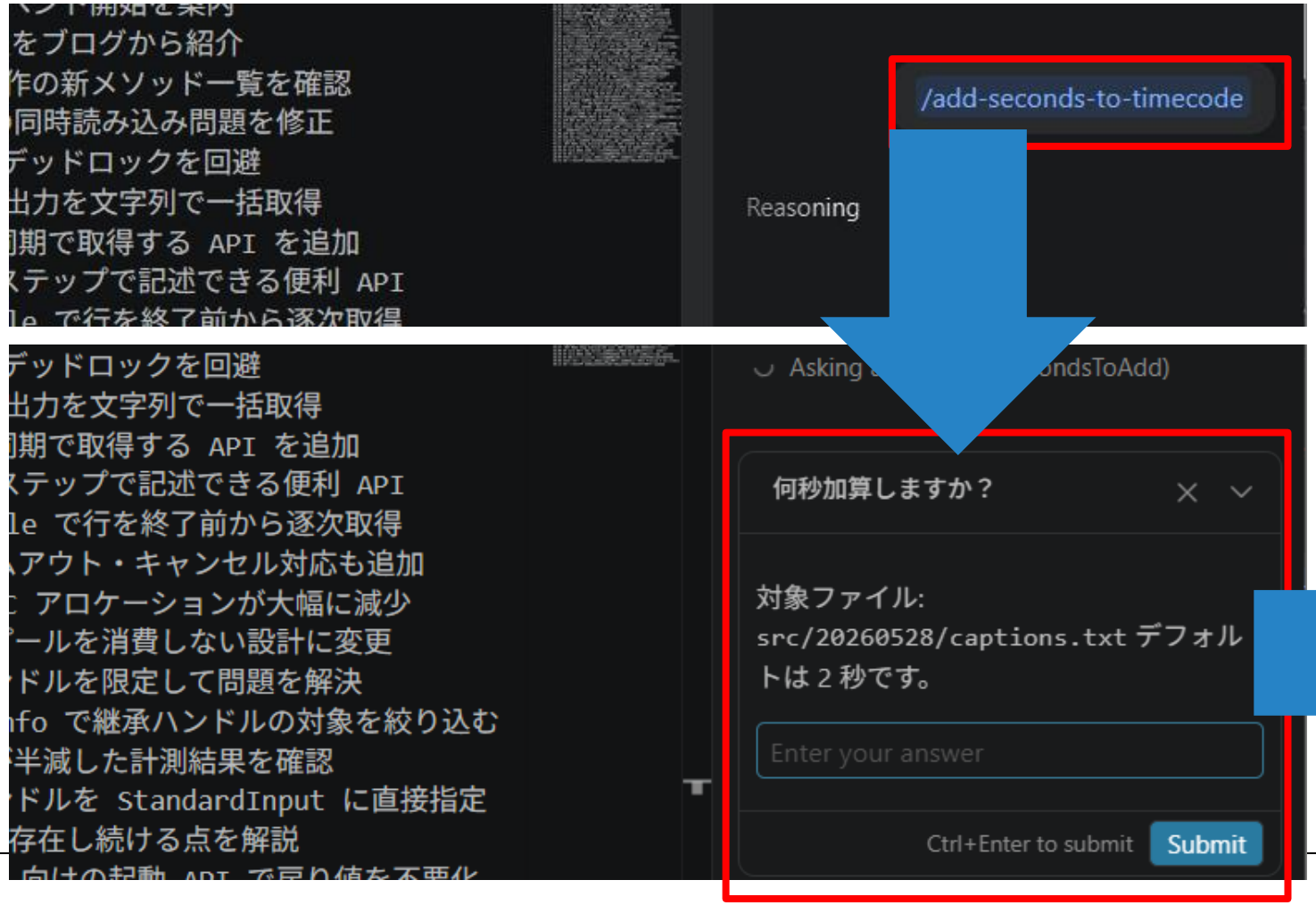
動画にするときにはオープニング（タイトルロゴ約3秒）を入れる

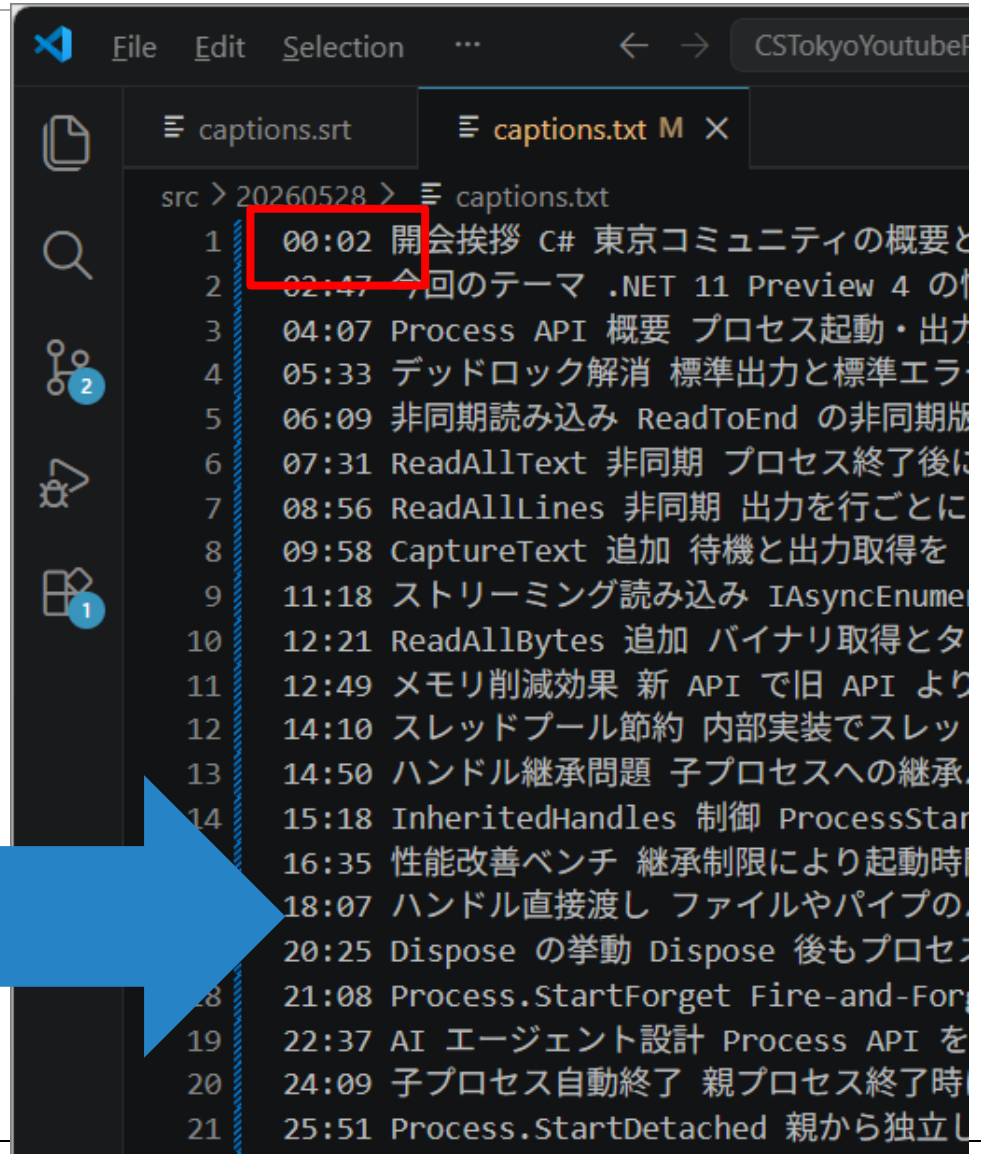
その結果、編集後は文字起こしと数秒ずれる

タイムスタンプの一括調整が必要になる

タイムスタンプを一括調整

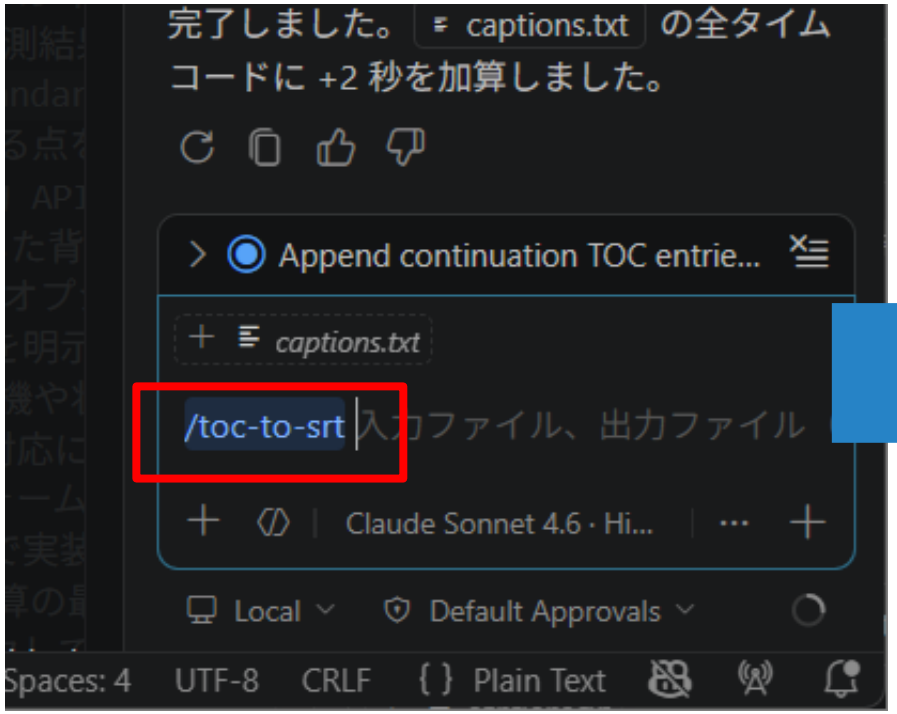
- これも GitHub Copilot の Skill で一発
- 何秒ずらすか聞いてくれて、完璧に調整できた

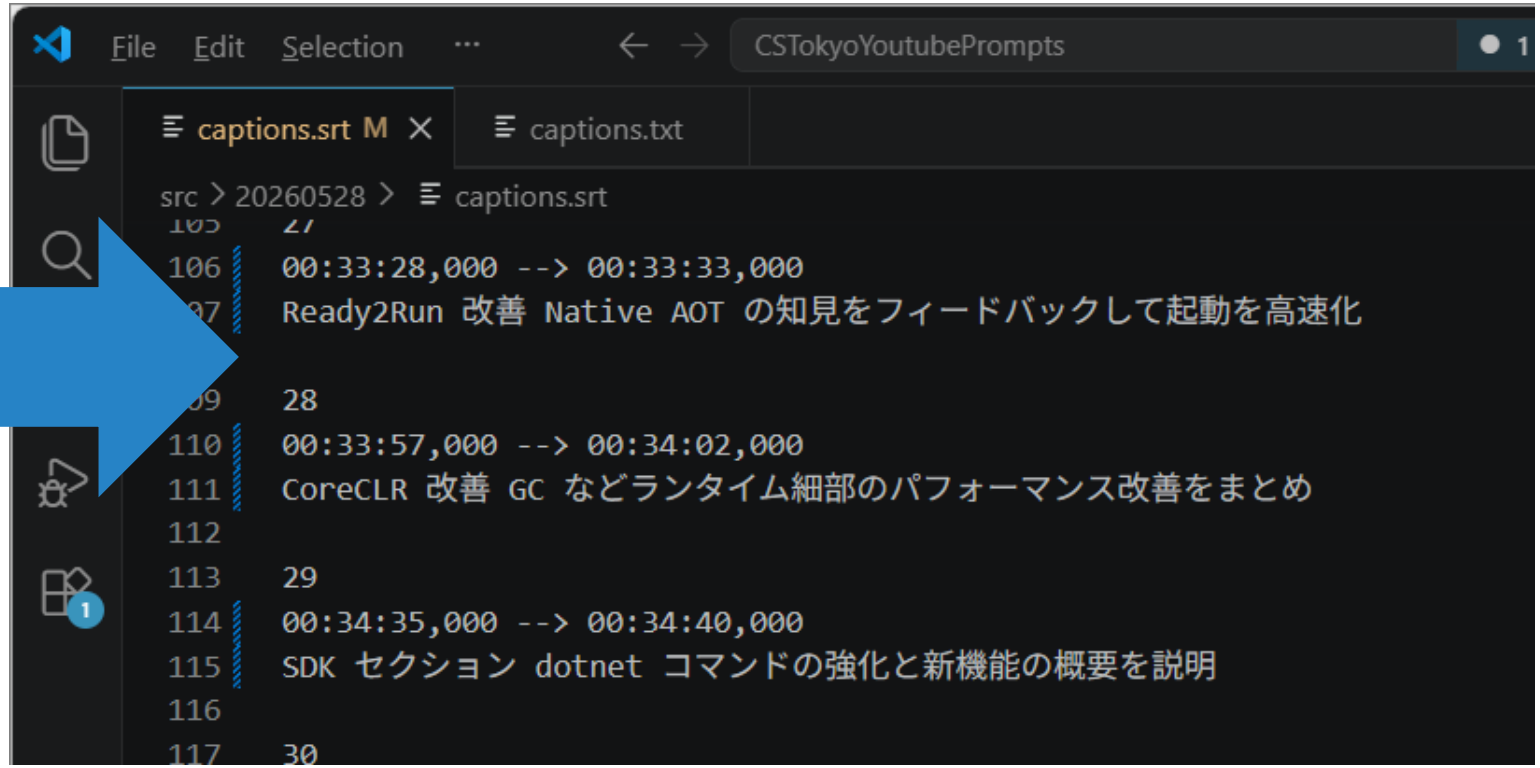




ファイル形式を変換する

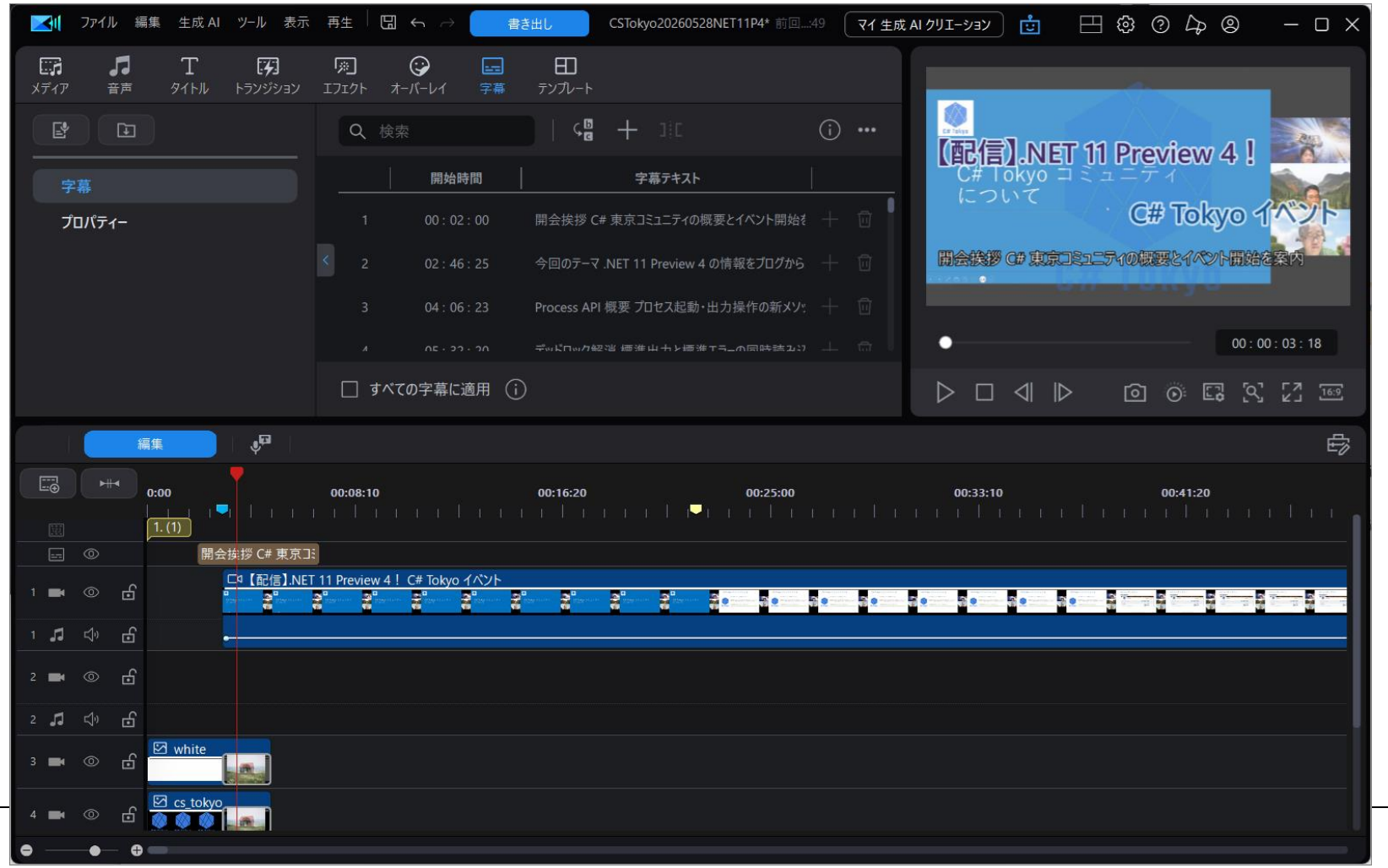
- 字幕・文字起こしデータは、しょせんテキストデータ
- テキスト変換なら言語変換が得意な Copilot で十分
- ここでも Skill を用意している





ファイル形式を変換する

- 変換成功。動画編集ソフトにインポートできる
- かなり実用的な仕上がり



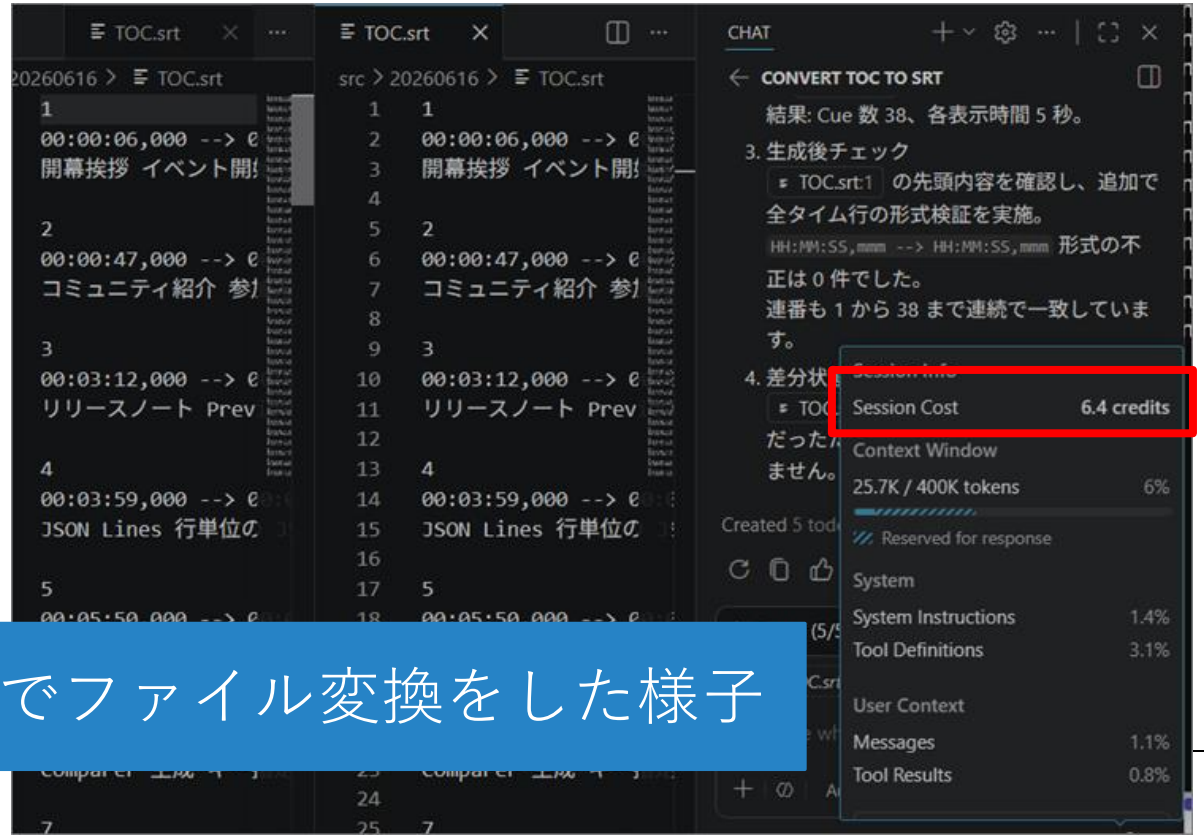
アポカリプス

2026/06 世界は AI Credits に包まれた

前セクションは 2026/05 に作成した手法です。

実際のところ、GitHub Copilot による書式変換は今でも可能。ただし、AI Credits を消費してしまう。

※2026/06 の GitHub Copilot の価格変更で AI Credits という単位での従量料金になり、多くの人がこれまでの数倍～数十倍の料金となってしまい、非常に話題になっています。



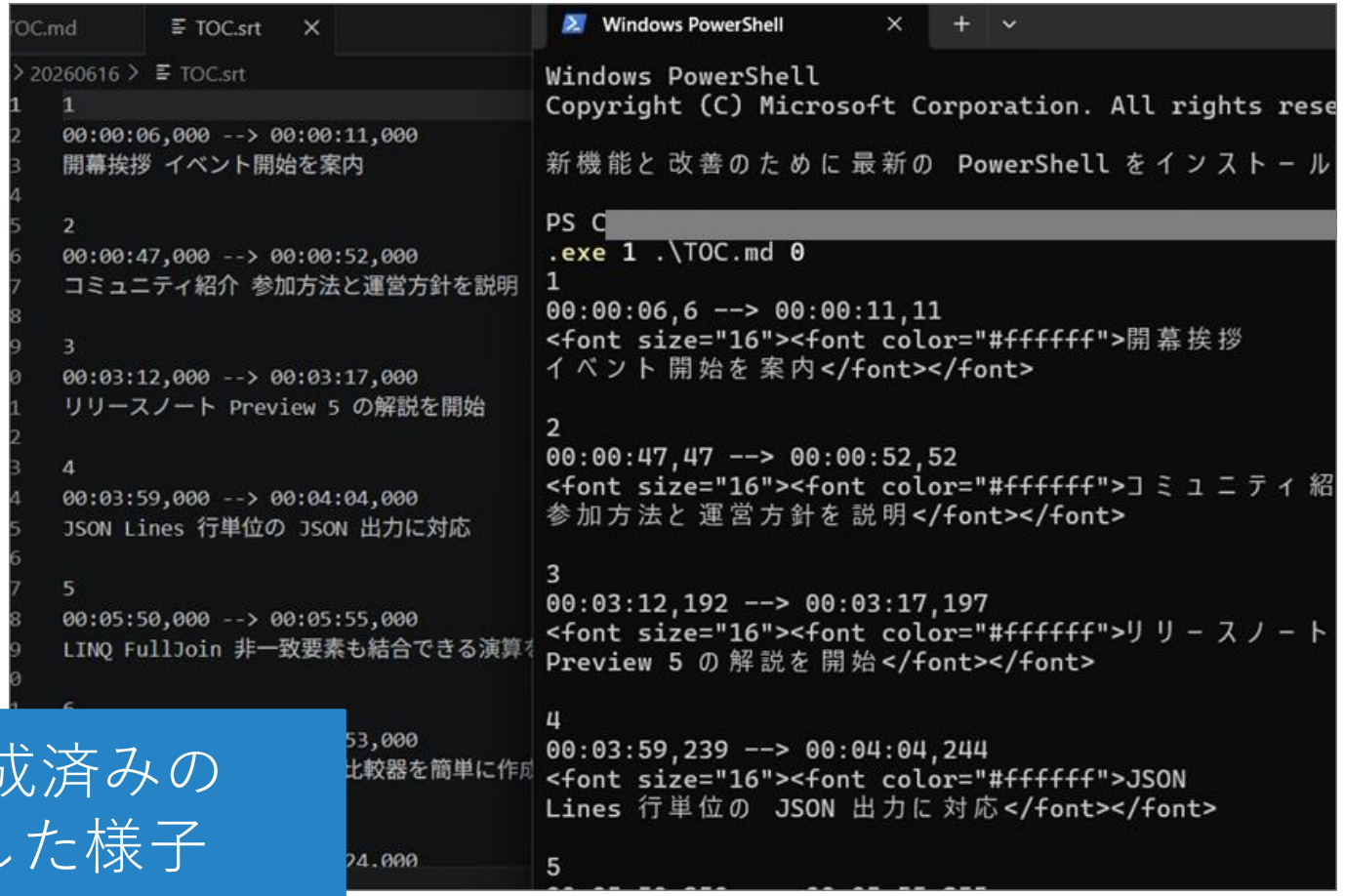
GitHub Copilot でファイル変換をした様子

ポストアポカリプスの救世主 C# コード

そこで登場させたいのが、C# コードです。
ファイル変換**ツールを作っておき**、ツール側で変換することで、

AI Credits 消費はなんと驚愕の **0** 。

すごくないですか？ 画期的です。



C# コードで書いた事前作成済みの
ツールでファイル変換をした様子

C# コード

小さな C# ツールで変換ルールを固定化し、毎回 AI に頼らない再利用可能な処理にします。

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ToSrt;

internal class ToSrt
{
    internal static string Format(long no, Line line, long start, long span)
    {
        var time_start_second = line.Time + start;
        var time_end_second = time_start_second + span;
        var time_start = Commons.TimeFormat(time_start_second);
        var time_end = Commons.TimeFormat(time_end_second);
        var title = line.Title.Trim();
        var description = string.Join(" ", line.Description).Trim();

        var result = @$"{no}
{time_start},{time_start_second} --> {time_end},{time_end_second}
<font size=""16""><font color=""#ffffff"">{title}
{description}</font></font>";
        return result;
    }
}
```

変換工程の

AI Credits 消費 0

変換ツールを作成してコスト削減

皆様のメリット

コスト予測：従量料金の増加を抑え、見積りしやすい

品質安定：同じルールで変換し、表記ゆれを低減

横展開：一度作ったツールを複数動画へ再利用

https://github.com/m-ishizaki/ToSrt

まとめ

結論：Copilot で動画制作もできる

- 「プログラミング専用」ではない：規約上も制限はなく、一般的な作業にも素直に応えてくれる
- Skill 化で作業が一気に効率化：文字起こし加工・タイムスタンプ調整・形式変換まで一発
- モデル選びとアドリブ力が鍵：Claude 系で品質が大きく向上し、実用レベルに到達

ありがとうございました

石崎 充良