字幕データの構造化形式の違いが大規模言語モデルの番組概要文生成性能に与える影響の分析

安田有希 [1]　望月貴裕 [1]
[1]NHK 放送技術研究所
{yasuda.y-hk,mochizuki.t-fm}@nhk.or.jp

概要

本研究では，放送字幕を入力として番組概要文を生成するタスクにおいて、入力フォーマットが大規模言語モデル（LLM）の生成性能, 安定性に与える影響を分析した。字幕を TSV、YAML、MARKDOWN、JSON、PLAIN の 5 形式に変換し、Qwen3-4B-Instruct を LoRA によって微調整して比較評価を行った。その結果、構造化のレベルが高い入力ほど生成性能が向上し、特に JSON 形式が最も高い精度を示すことを確認した。一方で、構造化レベルが低い PLAIN 形式では、言語モデルの破綻などを起因とする性能の劣化が顕著であった。

1　はじめに

電子番組表（EPG）における番組概要文の自動生成は、放送業務の効率化、検索性の向上、および視聴支援の観点から重要な課題である。近年、自然言語生成能力に優れた大規模言語モデル（Large Language Models; LLM）を用いて，動画の発話内容を書き起こしたテキストデータから要約文を生成する試みが報告されている [1, 2]。そこで我々は、出演者のセリフ、ナレーション、効果音の説明などから成る字幕放送用テキストデータ（以下「字幕データ」）を入力とした、LLM による番組概要文自動生成技術の開発に取り組んでいる。字幕データは一般的な自然言語文とは異なり、各字幕テキストに開始・終了時刻が付与されており、字幕テキスト間の連続性や関係性を反映した「時系列構造」を有している。しかし、多くの LLM はテキストチャット形式を前提として設計されているため、このような時系列構造を持つ字幕データは、必ずしも適切に解釈されるとは限らない。高精度な番組概要文生成技術を実現するためには、字幕データをどのような入力で LLM に与えるべきか、精査する必要がある。本研究では、放送字幕から EPG 番組概要文を自動生成するタスクを設定し、字幕の構造化形式の違いが生成性能に与える影響を比較する。

2　関連研究

放送字幕に関する従来研究は、主に聴覚障害者支援やアーカイブ利活用を目的とし、字幕生成・補正に焦点を当てた研究、あるいは字幕データを映像要約の補助的入力として利用する研究が中心である [3, 4, 5, 6]。これらのアプローチは、字幕データを機械学習により活用するという点で本研究と関連する。しかし、番組概要文そのものを生成対象とする研究は限られている。

また、本研究で扱う「時系列構造を持つ字幕データからの番組概要文生成」は、広くは Data-to-Text タスクの一種として位置づけられる [7, 8, 9]。Data-to-Text は、グラフや表形式などの構造自体に意味を有するデータ（以下「構造化データ」）から自然言語文を生成する問題として、自然言語処理分野における重要な研究テーマの一つである。

近年では、LLM の性能が入力プロンプトの設計に大きく依存することは、多くの研究および実務報告により指摘されており、自然言語入力を前提としたモデルに対して、構造化データをどのように入力するかという観点が改めて注目されている [10, 11, 12, 13, 14]。これらの研究の多くはチャット形式の質問応答タスクを対象としており、放送字幕のような「時刻情報・データ間の連続性・関係性」が混在した構造化データについて、構造化のレベルの違いが生成性能に与える影響を定量的に分析した研究はほとんど存在しない。

本研究は、Qwen3-4B-Instruct[15] に対して字幕データを複数の構造化データで与え、番組概要文生成における性能を比較することで、この研究ギャップを補完するものである。

3 実験

表 1 構造化データの例

構造化形式	字幕の表現
TSV	Start\tEnd\tText\n 0:00:41.26\t0:00:43.93\t まるめて ギュ。\n 0:00:41.26\t0:00:43.93\t しゅうちゅう。\n ... \t0:09:11.83\t0:09:14.13\ t てれますわ。
YAML	@subtitle\n Start: 0:00:41.26\n End: 0:00:43.93\n Text: まるめて ギュ。\n @subtitle\n Start: 0:00:41.26\n End: 0:00:43.93\n Text: しゅうちゅう。\n ... @subtitle\n Start: 0:09:11.83\n End: 0:09:14.13\n Text: てれますわ。
MARKDOWN	| Start | End | Text |\n |——|——|——|\n | 0:00:41.26 | 0:00:43.93 | まるめて ギュ。 |\n | 0:00:41.26 | 0:00:43.93 | しゅうちゅう。 |\n ... | 0:09:11.83 | 0:09:14.13 | てれますわ。 |
JSON	{\"Start\": \"0:00:41.26\", \"End\": \"0:00:43.93\", \"Text\": \"まるめて ギュ。\"}\n {\"Start\": \"0:00:41.26\", \"End\": \"0:00:43.93\", \"Text\": \"しゅうちゅう。\"}\n ... {\"Start\": \"0:09:11.83\", \"End\": \"0:09:14.13\", \"Text\": \"てれますわ。\"}
PLAIN	0:00:41.26 0:00:43.93 まるめて ギュ。 0:00:41.26 0:00:43.93 しゅうちゅう。 ... 0:09:11.83 0:09:14.13 てれますわ。

3.1 データセット

本研究では、NHK 総合および NHK E テレで放送された日本語テレビ番組の字幕データと、対応する EPG 番組概要文からなる 64,212 件のペアデータを用いた。字幕データは、字幕の開始・終了タイムコード、および字幕本文から構成される。

生成対象である番組概要文は、概ね 80〜150 文字程度の短文であり、字幕全体から番組の要点や主題を抽出して簡潔に要約するタスクである。全データからランダムに 99% を訓練データ、残りの 1% をテストデータとして使用した。これは，大規模データに対する LLM 微調整の計算コストを考慮しつつ，十分な統計的傾向を確認するためである。

3.2 データ構造形式

字幕データを以下の 5 種類の形式に変換し、構造化のレベルの違いが生成性能に与える影響を比較した。

- TSV: 各発話を行単位で分離し、タイムコードおよび本文をタブ区切りで表現。
- YAML: 発話情報を階層構造として記述し、属性を key-value 形式で付与。
- MARKDOWN: 改行や「ー」、「｜」を用いて発話単位をテーブル形式で明示。
- JSON: 各発話を {Start, End, Text} を要素とするオブジェクト配列として表現。
- PLAIN: タイムコード、字幕本文をスペースで連結した構造化レベルが低いデータ形式。

これらは構造化のレベルが異なる。例えば、データをスペースで区切っているのみの PLAIN に比べて、各発話を行やテーブルで表現している TSV および MARKDOWN は構造化レベルが高く、階層化およびオブジェクト配列化されている YAML や JSON はさらに構造化レベルが高い。このような構造化レベルの違いが番組概要文の生成品質にどのように寄与するかを分析することを目的とする。表 1 に各構造化形式の字幕表現例を示す。

3.3 モデルと学習設定

モデルには Qwen3-4B-Instruct を用い、LoRA による微調整を行った [16]。LoRA の rank・alpha、学習率などのハイパーパラメータは事前検証により設定し、すべての構造化形式で同一条件とした。

- Learning rate: 3×10^{-5}
- Batch size: 2
- LoRA rank: 16
- LoRA alpha: 32
- Epoch: 15
- 最大入力トークン長: 4096

学習は構造化形式ごとに独立して実施し、形式間の

影響が混在しないよう配慮した。LLM への入力プロンプトには構造化の情報を端的に説明する文章と番組タイトルを付け加えた。

3.4 評価指標

本研究では生成した番組概要文に対し、ROUGE-L、BLEU、BERTScore の 3 指標を用いて評価した [17, 18, 19]。なかでも、要約タスクの一般的な指標である ROUGE-L に焦点をあてて分析を行った。

4 実験結果と分析

4.1 実験結果

表 2 に構造化形式ごとの評価結果を示す。JSON は ROUGE-L、BLEU、BERTScore のすべてにおいて最も高い値を示した。一方、PLAIN は全指標で最も低い性能であった。

表 2 実験結果

構造化形式 / 評価指標	ROUGE-L	BERTScore	BLEU
TSV	0.411	0.759	0.272
YAML	0.422	0.763	0.276
MARKDOWN	0.415	0.760	0.265
JSON	**0.432**	**0.767**	**0.289**
PLAIN	0.334	0.727	0.171

4.2 評価値の分布に対する分析

図 1 に，各構造化形式における ROUGE-L の分布を示す箱ひげ図および蜂群図を示す。縦軸は ROUGE-L の値、横軸は構造化形式を表す。箱ひげ図では、第 1 四分位数から第 3 四分位数の範囲を色付けして示している。図中の各点は個々のテストサンプルを表し、通常点と異なる大きさで描画された点は外れ値を示す。蜂群図では可視性向上のため，各構造化形式についてテストサンプルからランダムに 500 件を抽出して表示している。なお、紙面の都合上、BLEU および BERTScore の分布は付録 A に示す。

全体的な傾向として、JSON や YAML といった構造化のレベルが高いデータは他の評価指標において分布幅が広いものの中央値が高い。一方、PLAIN は分布が最も狭く、低スコア領域にサンプルが集中している。TSV、MARKDOWN といった構造化レベルが中程度のデータは、構造化のレベルが高いデータに比べてやや分布幅が狭く、一部のサンプルで高スコアを達成している。

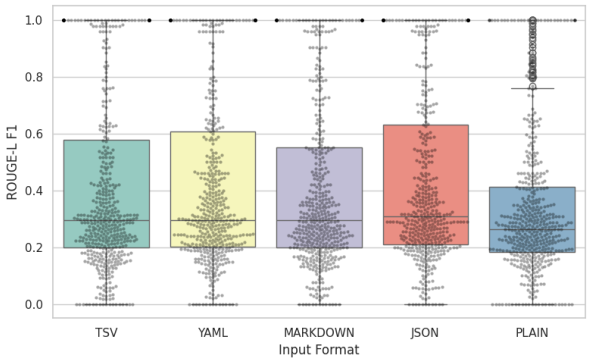

図 1 ROUGE-L 分布図

4.3 エラーの傾向

実験結果にもとづき、最も性能の良かった JSON と最も性能の悪かった PLAIN のエラーを比較、分析した。まず、ROUGE-L スコアにもとづき、0、0–0.2、0.2–0.4、0.4–0.6 の 4 区間に分け、各区間から 20 件ずつサンプルを抽出した。そのうえで、出力結果と参照概要文を比較し、エラーを人手で 7 種類に分類した。図 2、図 3、図 4、図 5 にそれぞれの区間におけるエラーの種類を示す。

ROUGE-L=0 の完全失敗区間では、PLAIN において出演者名の羅列など、番組概要文として成立していない出力が多数観測された。固有名詞の誤りも多く、構造化レベルが低いデータでは字幕構造を理解できない可能性が高いことが示唆される。

ROUGE-L が 0–0.2 および 0.2–0.4 の中間区間では、両形式ともに不自然な日本語表現が見られたが、言語モデルの破綻（繰り返しや言語混在）は PLAIN でより多く発生した。これは、構造化レベルの入力が言語生成の安定性にも影響を与える可能性を示している。

ROUGE-L が 0.4–0.6 の高スコア区間では、両形式とも日本語としての破綻はほぼ消失し、主に「どの情報をどの程度記述するか」といった高次の要約判断に関する誤りが中心となった。

5 考察

本研究の結果から、字幕を入力とする番組概要文生成においては、構造化形式における構造化レベルが、モデルの生成性能および安定性を大きく左右することが示された。特に JSON は、他の手法に比べてすべての指標で分散が高いものの高い精度を一貫して示しており、構造化レベルが高い入力が LLM による文脈理解を促進させることが確認された。

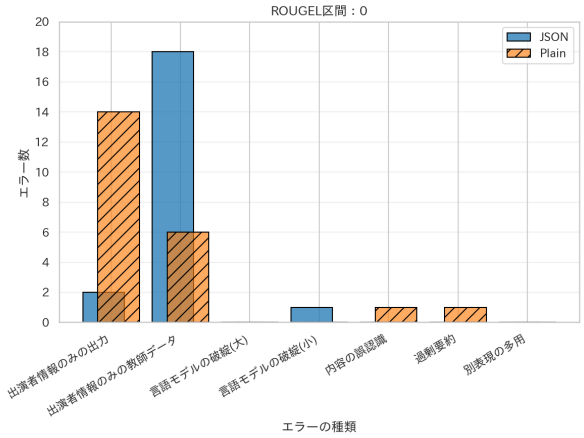

図 2　エラーの種類 (ROUGE-L：0)

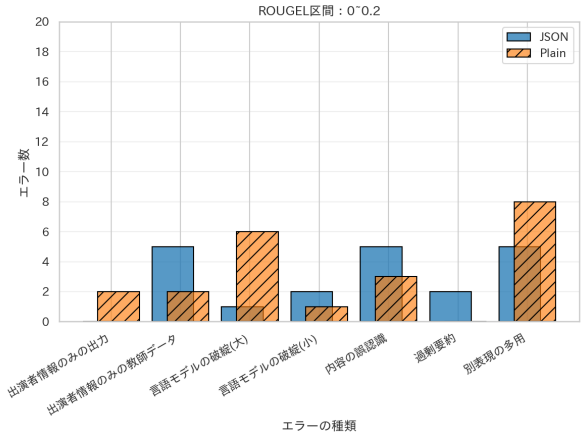

図 3　エラーの種類 (ROUGE-L:0-0.2)

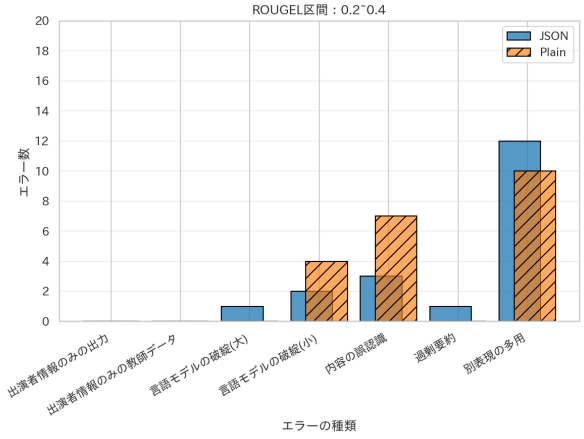

図 4　エラーの種類 (ROUGE-L:0.2-0.4)

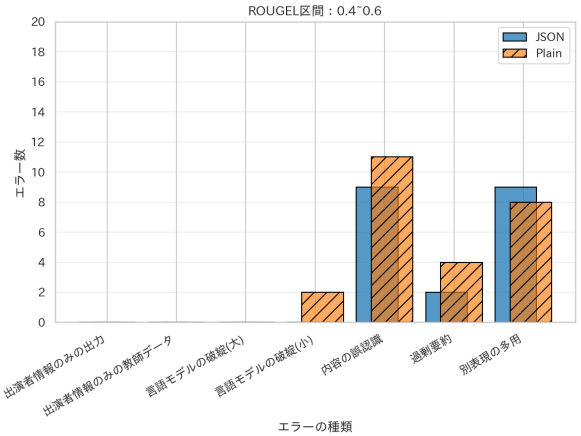

図 5　エラーの種類 (ROUGE-L：0.4-0.6)

JSON が優れた性能を示した要因として、発話単位や時刻情報が key/value により明確に区別され、字幕に内在する時系列構造が配列として保持される点が挙げられる。このような表現は、LLM にとって各情報の役割を誤解しにくく、要約対象となる文脈を一貫して解釈する助けとなると考えられる。一方、PLAIN では構造情報が失われるため、文脈理解の不安定化や言語モデルの破綻が生じやすい。

以上より、昨今の LLM を用いた字幕処理においては、「自然文として整形された入力」よりも、「構造が保持された入力」を用いることが、生成性能と効率の両立に重要である可能性が示唆された。

6　おわりに

本研究では、放送字幕を入力とした番組概要文生成タスクにおいて、構造化形式の違いが LLM の生成性能、安定性に与える影響を分析した。TSV、YAML、MARKDOWN、JSON、PLAIN の 5 形式を統一条件下で比較した結果、構造化の明示性が高い入力ほど性能が安定し、特に JSON は最も一貫した生成品質を示すことを確認した。

本研究は、映像を入力とするマルチモーダル要約に向けた前段階として、字幕という言語情報に着目し、構造化データ設計の重要性を示した点に位置づけられる。今後は、TOON や XML などの新たな構造化形式との比較、構造情報を Embedding 化する層をモデルに結合するといった、モデルがより直接的に構造情を扱う手法の検討、およびさらに大規模な LLM を用いた再検証を通じて、字幕処理および番組概要文生成手法のさらなる高度化を目指す。

参考文献

[1] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025.

[2] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025.

[3] K. Penyameen, G.M Siva Suriya Rajan, A. Arshath Ahamed, S. Yugesh Ram, J John Shiny, and A Periya Nayaki. Ai-based automated subtitle generation system for multilingual video transcription and embedding. In **2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)**, pp. 1096–1101, 2025.

[4] Jakob Poncelet and Hugo Van hamme. Leveraging broadcast media subtitle transcripts for automatic speech recognition and subtitling, 2025.

[5] Ádám Varga, Balázs Tarján, Zoltán Tobler, György Szaszák, Tibor Fegyó, Csaba Bordás, and Péter Mihajlik. Automatic close captioning for live hungarian television broadcast speech: A fast and resource-efficient approach. In Andrey Ronzhin, Rodmonga Potapova, and Nikos Fakotakis, editors, **Speech and Computer**, pp. 105–112, Cham, 2015. Springer International Publishing.

[6] Prashant Giridhar Shambharkar and Ruchi Goel. Analysis of real time video summarization using subtitles. In **2021 International Conference on Industrial Electronics Research and Applications (ICIERA)**, pp. 1–4, 2021.

[7] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In Jian Su, Kevin Duh, and Xavier Carreras, editors, **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 1203–1213, Austin, Texas, November 2016. Association for Computational Linguistics.

[8] Ehud Reiter. An architecture for data-to-text systems. In Stephan Busemann, editor, **Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)**, pp. 97–104, Saarbrücken, Germany, June 2007. DFKI GmbH.

[9] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 2253–2263, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[10] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. Does prompt formatting have any impact on llm performance?, 2024.

[11] Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. ProSA: Assessing and understanding the prompt sensitivity of LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 1950–1976, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[12] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In **Proceedings of the 17th ACM International Conference on Web Search and Data Mining**, WSDM '24, p. 645–654, New York, NY, USA, 2024. Association for Computing Machinery.

[13] Dehai Min, Nan Hu, Rihui Jin, Nuo Lin, Jiaoyan Chen, Yongrui Chen, Yu Li, Guilin Qi, Yun Li, Nijun Li, and Qianren Wang. Exploring the impact of table-to-text methods on augmenting LLM-based question answering with domain hybrid data. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)**, pp. 464–482, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[14] Zdeněk Kasner and Ondrej Dusek. Beyond traditional benchmarks: Analyzing behaviors of open LLMs on data-to-text generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 12045–12072, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[15] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. **ICLR**, Vol. 1, No. 2, p. 3, 2022.

[17] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In **Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics**, pp. 150–157, 2003.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

A　BLEU と BERTScore の分布

図 6、図 7 に各構造化形式における BLEU と BERTScore の分布を示す。

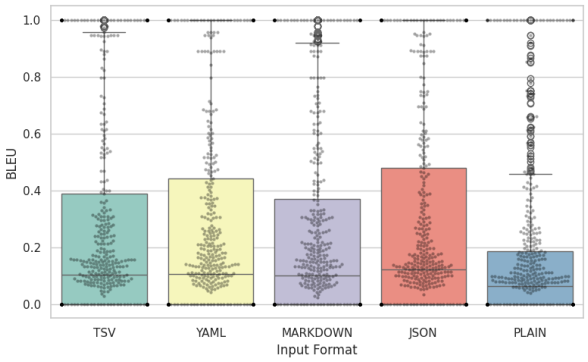

図 6　BLEU 分布図

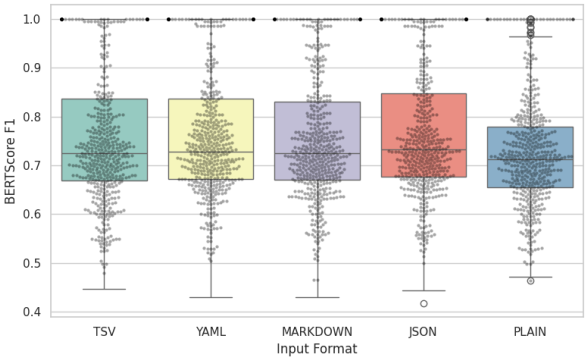

図 7　BERTScore 分布図