LLM を活用した分節談話表示理論に基づく日本語談話構造解析

富崎智睦 [1] 三上燿輔 [1,2] 綿引周 [1] 谷中瞳 [1,2,3]
[1] 東京大学 [2] 理化学研究所 [3] 東北大学
{tomisaki-no-mail,ymikami,hyanaka}@is.s.u-tokyo.ac.jp
amanew@g.ecc.u-tokyo.ac.jp

概要

分節談話表示理論 (SDRT) は文章や会話中の談話関係を記述する理論である．英語では LLM をファインチューニングし，SDRT に基づく談話解析を高精度で行う手法が提案されている．しかし，日本語の SDRT コーパスは整備されておらず，日本語における同手法の有効性は自明でない．そこで，本研究では英語の SDRT コーパスを翻訳して日本語 SDRT コーパスを作成し，作成したコーパスを用いて日本語 SDRT 解析器を構築した．結果，日本語の解析器は英語と同等の性能を示した．また，談話標識などの特定の談話関係に頻出する言語表現に着目してアブレーション分析を行った結果，頻出表現が必ずしも談話関係の予測に寄与しないことが示唆された．

1 はじめに

会話や文章の意味を理解するためには，個々の文または節の意味だけではなく，それらがどのように関係し意味を成しているかを理解することが不可欠である．この文や節同士に成り立つ関係を談話関係といい，それらを識別するタスクは談話構造解析と呼ばれる．また，談話関係を記述するための理論の一つに分節談話表示理論 (Segmented Discourse Representation Theory; SDRT) [1] が挙げられる．

近年，大規模言語モデル (LLM) を利用して英語において高い精度での SDRT 解析を可能にする手法が提案されている．例えば，Bennis ら [2] は BERT の埋め込み表現を用いることで高精度の SDRT 解析を実現しており，Thompson ら [3] は LLM をファインチューニングして SDRT 解析器を構築することでさらなる高精度を達成している．

しかし，日本語の SDRT コーパスは十分に整備されておらず，日本語においてどの程度精度よく SDRT 解析ができるかは明らかでない．そこで，本研究ではまず，日本語の SDRT 解析器を構築するために，既存の英語 SDRT コーパスである Minecraft Structured Dialogue Corpus (MSDC) [4] を OpenAI API の GPT-5.1 を用いて翻訳し，日本語の SDRT コーパスを構築した．そして，Thompson [3] らの手法に従い，日本語に翻訳された MSDC コーパスを用いて Llama-3-8B と Llama-3-swallow-8B-0.1v をそれぞれファインチューニングし，2 種類の日本語 SDRT 解析器を構築した．また，Llama-3-8B を元の英語 MSDC コーパスでファインチューニングして英語 SDRT 解析器を構築し，日本語の解析器とその性能を比較した．

実験の結果，英語モデルも日本語モデルも全体としての性能は概ね同じであることを確認した．さらに，特定の談話関係に頻出する日本語の談話標識や終助詞に着目してアブレーション分析を行った結果，必ずしも頻出する言語表現が談話関係の予測に寄与していないことが示唆された．

2 背景

2.1 分節談話表示理論 (SDRT)

SDRT は文章や会話の談話構造を分析するための理論の一つである．SDRT において，談話は elementary discourse units (EDU) または elementary event units (EEU) と呼ばれる節の単位に分割される．EDU は会話や文章の節そのものを指すのに対し，EEU は会話が行われている状況や，その中で起こる出来事を表現している．例えば，表 1 における，節 2 が EDU で，節 5 が EEU である．そして談話全体は，EDU や EEU をノード，それらの間に成り立つ談話関係をエッジとした，弱連結有向非巡回グラフとして表される．

SDRT 以外の談話関係に関する代表的な理論としては，修辞構造理論 (Rhetorical Structure Theory; RST) [5] が挙げられる．RST は談話構造を木構造として表現している．RST における談話関係の定義に

表 1 MSDC コーパスの翻訳例，節 5 と節 7 については，単純に翻訳を行うと命令形の文として訳出されてしまうため，翻訳の対象外とした．

節ラベル	話し手	原文（英語）	訳文（日本語）
2	Builder	lets do this	やろうか
3	Architect	place a red block in front of the blue block	青いブロックの前に赤いブロックを置いてください
4	Architect	along the line toward the purple block	紫のブロックに向かう線に沿って
5	Builder	place red 4 1 0	place red 4 1 0
6	Architect	place another red block along that line	その線にそって，赤いブロックをもう一つおいてください
7	Builder	place red 3 1 0	place red 3 1 0
談話関係		Elaboration(3,4) Result(4,5) Result(5,6) Narration(3,6) Result(6,7)	

は未定義の概念が含まれているため，その厳密性が問題になる場合がある．それに対し SDRT は，各談話関係を形式意味論の理論に基づいて比較的厳密に定義できるという利点がある．さらに，SDRT は複数人での会話を扱うことができるという利点がある．

2.2 MSDC コーパス

SDRT に基づく談話関係がアノテートされたコーパスとして，Minecraft Dialogue Corpus (MDC) コーパスを元に作られた MSDC [4] がある．MDC には，ゲーム Minecraft の中で Builder と Architect と呼ばれる二人のプレイヤーが会話をしながら協調してタスクに取り組む様子が記録されている．具体的には，Architect が Builder に対して指示を出し，Builder がその指示に従って，ゲーム内で一定の構造物を構築するというタスクに取り組む．表 1 に MSDC コーパスの一部を示す．MSDC は 407 件の会話セッションからなり，それぞれの会話セッションには，タスクの中で交わされたチャットが節ごとに分割され EDU として含まれており，表 1 の節 2，3，4，6 が EDU に対応する．また節 5，7 はブロックの設置や破壊といった Builder がゲーム内で行った行為に関する記録であり，これは EEU に対応する．MSDC コーパスで使われている談話関係を付録の表 6 に示す．

2.3 SDRT 解析の方法論

大規模言語モデルを用いた代表的な SDRT 解析の手法として，BERTLine [2] と Llamipa [3] がある．

BERTLine BERTLine は EDU および EEU 間の接続と，その接続に関して成り立つ談話関係を同時に学習するマルチタスクアーキテクチャになっている．モデルは BERT の EDU および EEU ペアに対する埋め込み表現を利用して，EDU，EEU 間の接続および談話関係に関する予測を得ている．

Llamipa Llama Incremental Parser (Llamipa) は，Thompson らによって提案された SDRT 解析の手法であり，モデルは Llama のファインチューニングによって構築される．Llamipa は，予測対象の節と，対象に先行する直近の最大 15 個分の節，直近の 15 節に成り立つ談話関係を入力として受け取り，対象の節に対して成立する談話関係を出力する．先頭の節から順に談話関係を予測し，その結果を後続の予測に利用することで，テキスト全体の談話構造を予測することができる．結果として，Llamipa は BERTLine を上回る性能を達成したと報告されている．

3 日本語 SDRT 解析器の構築

本研究では MSDC コーパスを日本語に翻訳して日本語 SDRT コーパスを構築し，構築したコーパスを用いて Llamipa [3] の構築手法に従い LLM をファインチューニングすることで日本語の SDRT 解析器を構築する．

3.1 MSDC コーパスの翻訳

日本語 SDRT 解析器を構築するための学習データを用意するため，MSDC コーパスを LLM を用いて日本語に翻訳した．翻訳には，OpenAI API の GPT-5.1[1] を用いた．翻訳によって節の順番が前後しないように，節ごとに翻訳を行った．また，コンテキストを加味してできるだけ自然に対象の節を翻訳するため，GPT-5.1 を用いて翻訳を行う際，翻訳対象の節と，それが含まれている会話セッションをコンテキストとして与えた．ただし，MSDC の EEU については，単純に翻訳を行うと命令形の文として訳出されてしまうため，翻訳の対象外とした．翻訳の例を表 1 に示す．翻訳に用いたプロンプトは付録 A を参照のこと．

1） https://openai.com/ja-JP/index/gpt-5-system-card/

表 2　翻訳エラーの例

話し手	原文（英語）	訳文（日本語）	備考
Builder	“any color”	「どんな色でもいいよ」	「どんな色でもいい?」等が適切
Builder	“or still orange?”	「それともオレンジのままでいい?」	
Architect	“on the top of orange block”	「一番上のオレンジのブロックの上に」	「ブロックの上に」ではなく「ブロックの」等が適切
Architect	“put a yellow block to the left of it”	「その左側に黄色のブロックを置いてください」	
Architect	“On the top and bottom”	「上と下に,」	「上と下に」が重複
Architect	“we are going to add 2 block extensions”	「上と下に，ブロックを二つ分突き出させます」	

表 3　XCOMET-XL を用いた翻訳データの評価．0.0 から 1.0 の間のスコアで翻訳の品質を評価している．

評価データ	スコア
学習データ	0.84
テストデータ	0.85

表 4　構築した SDRT 解析器の性能

	英語	日本語	
ベースモデル	Llama3	Llama3	Llama3-swallow
適合率	0.80	0.79	0.80
再現率	0.79	0.78	0.79
F1 スコア	0.80	0.79	0.79

3.2　翻訳コーパスの評価

日本語に翻訳した MSDC コーパスについて，学習データ 305 件の会話セッションの内の 30 件，テストデータ 102 件のうち 10 件をサンプリングし，その品質を日本語母語話者である著者 1 名によって人手で評価した．結果として，一部軽微なエラーが見られたものの，日本語 SDRT 解析器の構築に十分な品質が担保されていると判断した．翻訳エラーの例を表 2 に示す．

また，コーパス全体の翻訳の品質を，機械翻訳の評価モデルである XCOMET-XL [6] を用いて評価した．XCOMET-XL は，対応する翻訳前のテキストと翻訳後のテキストを複数与えることで，その翻訳の質を 0.0 から 1.0 のスコアで評価することができる．日本語 SDRT 解析器の構築に用いた日本語 MSDC コーパスのテストデータと学習データを，それぞれ話し手の切り替わりごとに分割し，XCOMET-XL に与えて評価した．その結果を表 3 に示す．いずれも評価スコアは 0.85 程度となっており，この結果はある程度正しい翻訳がされていることを示す．

3.3　SDRT 解析器の学習

ファインチューニングのベースモデルとしては，Llama-3-8B [7] に加え Llama-3-swallow-8B-0.1v [8] を用いた．Llama-3-swallow-8B-0.1v は，Llama-3-8B に対し日本語能力を強化するための継続事前学習を行ったモデルである．学習においては，Llamipa [3] の構築手法に従い，QLoRA [9] で 3 エポック分の学習を行った．その他のハイパーパラメータについても，Llamipa [3] の設定に従った．

4　実験

4.1　実験設定

日本語の SDRT 解析器の比較対象として，元の英語 MSDC コーパスを用いて Llama-3-8B モデルをファインチューニングし，英語 SDRT 解析器を構築した．構築した英語または日本語 SDRT 解析器について，それぞれオリジナル MSDC コーパスのテストデータ，または翻訳されたテストデータを用いて評価した．具体的には SDRT の談話関係ごとに，適合率，再現率，F1 スコアを計算し比較した．また，談話関係の件数で重み付けした，適合率，再現率，F1 スコアを総合的な性能評価の指標として算出した．

学習および性能評価に用いた MSDC コーパスについて，学習データは 12594 件の EDU と 3439 件の EEU からなり，テストデータは 3886 件の EDU と 1048 件の EEU からなる．その他の統計情報は付録の表 6 に示す．

4.2　全体的な結果

英語と日本語それぞれの SDRT 解析器の性能を表 4 に示す．談話関係ごとの性能は付録の 7 に示す．総合的な性能としては，どのモデルも大きな違いは見られなかった．

また，Llama-3-8B に談話関係の定義をプロンプトとして与えて，ファインチューニングを行わず zero-shot プロンプティングで予測を行う場合と性能を比較した．結果，英語と日本語のいずれのデータに対しても F1 スコアは 0.02 となった．このように談話関係の予測性能がきわめて低かったことから，zero-shot プロンプティングによる SDRT 解析は困難であることが示唆された．

表 5 アブレーション前後での F1 スコア．出現回数の列は (言語表現が着目する談話関係の出現位置の列で指定されている位置に現れる回数)/(言語表現が出現する節の個数) という形式で表されている．出現位置は x が談話関係の第一引数，y が第二引数を表す．

言語表現	着目する談話関係	出現回数	出現位置	アブレーション前	アブレーション後
それから	Narration	36/54	y	0.88	0.88
そしたら	Narration	36/40	y	0.90	0.91
	Result	35/40	y	0.88	0.88
じゃあ	Narration	284/379	y	0.90	0.91
	Result	316/379	y	0.88	0.88
だから	Elaboration	41/47	y	0.76	0.75
でも	Contrast	46/54	y	0.79	0.19
よ	Correction	22/222	y	0.80	0.79
ね	Acknowledgement	145/369	y	0.86	0.85
か	Clarification-Q	24/60	y	0.71	0.73
	Confirmation-Q	8/60	y	0.92	0.92
	Question-answer pair	37/60	x	0.83	0.85

4.3 アブレーション分析

構築した Llama3 ベースの日本語 SDRT 解析器が，どの程度談話標識や終助詞といった特定の言語表現に依存して予測を行うのか調査するため，特定の言語表現を除去したときの談話関係の予測性能の変化を評価するアブレーションテストを行った．アブレーション対象の日本語談話標識として，MSDC コーパスのテストデータのうち 40 件以上の EDU に現れる接続詞を選定した．具体的には節の先頭に現れる，「それから」「そしたら」「じゃあ」「だから」「でも」をアブレーションの対象とした．

接続詞「それから」を含む節は MSDC 翻訳コーパスにおいて，談話関係 Narration と共起することが多かったことから，構築した SDRT 解析器は Narration を予測する際，この表現を一つの手がかりにしているという仮説が考えられる．同様に，「そしたら」と「じゃあ」は Result や Narration，「だから」は Elaboration，「でも」は Contrast と共起することが多く，これらの表現もそれぞれ共起する談話関係の予測に影響を与えていることが期待される．

また，終助詞の「よ」「ね」「か」もアブレーションの対象とした．終助詞の「よ」には，話し手と聞き手の情報の不一致を前提とし，聞き手が知らないことに注意を向けさせる働きがあり，終助詞の「ね」には話し手と聞き手の情報が一致していると想定される場合に，その内容を確認する働きがある [10]．このことから，「よ」は，談話関係 Correction を導き，「ね」は Acknowledgement を導くことが期待される．また，終助詞「か」は質問や疑問の意を表すことから，Clarification-Q や Confirmation-Q，Question-answer pair 等の関係を導くことが期待される．

アブレーションテストの結果を表 5 に示す．「でも」を削除した場合，Contrast 予測の F1 スコアが 0.79 から 0.19 に悪化した．しかし，接続詞「それから」「そしたら」「じゃあ」「だから」，終助詞「よ」「ね」「か」のアブレーションに関しては，各談話関係の予測性能に大きな変化は見られなかった．

これらの結果から，構築した日本語 SDRT 解析器は接続詞「でも」の有無に大きく依存して Contrast を予測していると言える．これに対して，その他の接続詞「それから」「そしたら」「じゃあ」「だから」や終助詞「よ」「ね」「か」については，F1 スコアに大きな違いは生じず，これらの表現が特定の談話関係の予測に寄与しているという仮説は確かめられなかった．これらの表現と共起しやすい談話関係は，特定の談話標識よりも節の意味内容や先行する節間の談話構造に依存して予測されるということが示唆される．

5 おわりに

本研究では LLM を用いて既存の英語 SDRT コーパスを日本語に翻訳し，翻訳された日本語の SDRT コーパスを用いて日本語の SDRT 解析器を構築し，その性能を英語の解析器と比較した．その結果，日本語の SDRT 解析は英語の SDRT 解析器と同等の性能を示した．また，日本語の談話標識や終助詞についてアブレーションテストを実施し，Contrast の予測に接続詞「でも」の有無が大きく関わっていることを示した．今後の課題として，SDRT コーパスの拡張とともに，SDRT 解析器の談話関係の予測に大きく寄与する表現とそうでない表現との違いに着目してさらなる分析を進める．

謝辞

本研究は JSPS 科研費 JP24H00809 の助成を受けたものである.

参考文献

[1] N. Asher and A. Lascarides. **Logics of Conversation.** Cambridge University Press, 2003.

[2] Zineb Bennis, Julie Hunter, and Nicholas Asher. A simple but effective model for attachment in discourse parsing with multi-task learning for relation labeling. In Andreas Vlachos and Isabelle Augenstein, editors, **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 3412–3417, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[3] Kate Thompson, Akshay Chaturvedi, Julie Hunter, and Nicholas Asher. Llamipa: An incremental discourse parser, 2024.

[4] Kate Thompson, Julie Hunter, and Nicholas Asher. Discourse structure for the Minecraft corpus. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 4957–4967, Torino, Italia, May 2024. ELRA and ICCL.

[5] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. **Text & Talk**, Vol. 8, pp. 243 – 281, 1988.

[6] Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. xCOMET: Transparent machine translation evaluation through fine-grained error detection. **Transactions of the Association for Computational Linguistics**, Vol. 12, pp. 979–995, 2024.

[7] Jauhri A. Pandey A. Kadian A. Al-Dahle A. Letman A. Mathur A. Schelten A. Yang A. Fan A. et al. Dubey, A. The llama 3 herd of models.

[8] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities, 2024.

[9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[10] 隆志益岡, 行則田窪. 基礎日本語文法. くろしお出版, 1989.

表 6　MSDC コーパスにおける各談話関係の件数と意味

談話関係	学習	テスト	意味
Acknowledgement(x,y)	2470	792	節 x の発話者の意図を節 y の発話者が受け入れたまたは達成したことを，節 y が含意する．
Alternation(x,y)	108	24	節 x と節 y のいずれかの内容が真である．
Clarification-Q(x,y)	560	134	節 y が節 x に関する質問で，何が起こったか，または，何が言われたかを明らかにする質問．
Comment(x,y)	866	382	節 y が節 x に関連する内容についての意見または評価を提供している．
Conditional(x,y)	39	13	節 x が仮定で，節 y がその仮定から導かれる帰結である．
Confirmation-Q(x,y)	551	159	節 y が，節 x の内容が正しいかどうかに関する質問である．
Continuation(x,y)	1220	321	節 x と節 y が同じトピックに属する．
Contrast(x,y)	223	68	節 x と節 y について，意味的な構造は似ているが，主題は対照的．
Correction(x,y)	1214	403	節 y が節 x の内容の一部を訂正している．
Elaboration(x,y)	2227	720	節 x で導入される出来事について，節 y がより多くの情報を提供している．
Explanation(x,y)	31	16	節 y が節 x で起こったことの理由や原因を述べている．
Narration(x,y)	2406	693	節 x と節 y で起こった出来事が，述べられたのと一致する順番で起きる．
Question-answer pair(x,y)	1100	290	節 x が質問で，節 y がそれに対する答えになっている．
Q-Elaboration(x,y)	130	38	節 y が節 x についてより多くの情報を得るためになされた質問である．
Result(x,y)	5857	1765	節 x の主要な出来事が，節 y で述べられる出来事の原因として理解される．
Sequence(x,y)	16	6	節 x と節 y で一連の行為，出来事を表している．

表 7　構築した SDRT 解析器の談話関係ごとの性能

	英語 (Llama3 ベース)			日本語 (Llama3 ベース)			日本語 (Llama3-swallow ベース)			
	適合率	再現率	F1 スコア	適合率	再現率	F1 スコア	適合率	再現率	F1 スコア	件数
Acknowledgement	0.86	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	789
Alternation	0.85	0.96	0.90	0.81	0.92	0.86	0.82	0.96	0.88	24
Clarification-Q	0.67	0.72	0.69	0.69	0.72	0.71	0.69	0.72	0.71	134
Comment	0.62	0.54	0.58	0.63	0.53	0.57	0.64	0.54	0.59	382
Conditional	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13
Confirmation-Q	0.97	0.89	0.93	0.94	0.89	0.91	0.93	0.89	0.91	159
Continuation	0.40	0.52	0.45	0.38	0.50	0.43	0.41	0.51	0.45	320
Contrast	0.88	0.66	0.75	0.88	0.69	0.77	0.84	0.64	0.73	67
Correction	0.73	0.71	0.72	0.78	0.72	0.75	0.78	0.71	0.74	403
Elaboration	0.78	0.76	0.77	0.77	0.75	0.76	0.75	0.78	0.76	720
Explanation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16
Narration	0.82	0.82	0.82	0.79	0.78	0.79	0.81	0.80	0.81	693
Question-answer_pair	0.85	0.77	0.81	0.83	0.79	0.81	0.86	0.81	0.83	289
Q-Elaboration	0.48	0.39	0.43	0.51	0.47	0.49	0.45	0.37	0.41	38
Result	0.90	0.91	0.90	0.90	0.90	0.90	0.90	0.91	0.91	1764
Sequence	0.50	0.17	0.25	1.00	0.17	0.29	1.00	0.17	0.29	6
関係ごとの重み付きスコア	0.80	0.79	0.79	0.79	0.78	0.79	0.80	0.79	0.79	

A　翻訳に用いたプロンプト

````
{"role": "system,
"content":
"""
## Task
You are given a multi-turn Minecraft chat log in JSON format,a target clause,
and your translation of the two previous clauses of the target clause.
The target clause is one of the clauses in the chat log.
Translate the clause into Japanese as possible as naturally.

## INPUT FORMAT
You are given the following JSON structure as input:

```
# Context
[{"text": "<clause1>","speaker": "<speaker1>"},{"text": "<clause2>","speaker": "<speaker2>"},...]
# Target clause
{"text": "<target_clause>","speaker": "<target_speaker>"}
```

## Rules
- No comments
- Translate only the target clause
- Ouptput only the translated text
- The transcript includes descriptions of in-game actions. Keep them as they are.
    - For example, if the text is "pick yellow 0 0 1, place red 1 0 0",
    Do not translate the action commands. Keep them in English
- Translate as naturally as possible while considering the context and previous translated clauses.
"""
}
{"role": "user", "content": "#Context\n" + <会話セッション全体>}
{"role: "user, "content" "# Target clause\n" + <翻訳対象の節>}
````