多観点検証に基づく要約における忠実性向上

酒向 颯也 [1]　小杉 哲 [1]　船越 孝太郎 [1]　奥村 学 [1]
[1] 東京科学大学
{sakou,kosugi,funakoshi,oku}@lr.first.iir.isct.ac.jp

概要

大規模言語モデルによる要約生成では，流暢さが向上する一方で，入力文書に基づかないハルシネーションの問題が依然として大きな課題である．本研究では，複数の検証エージェントが異なる観点から要約を確認し，指摘に基づいて反復的に修正を行う多観点からの検証に基づく要約フレームワークを提案する．各エージェントは事実整合性，数値・時間的一致性，固有表現の一致性などを独立に検証し，各エージェントからの指摘を統合し，入力文書に対する忠実性に特化した反復的な修正を適用する．これにより，流暢さを維持しつつ，事実の誤りのみを効率的に排除する．複数の要約データセットを用いた評価により，提案手法が従来手法と比較して多くのデータセットにおいて要約の忠実性を向上させる傾向にあることを示した．

1　はじめに

大規模言語モデル (LLM) の発展により，要約タスクは大きな進展を遂げている．特に生成型要約では，長文記事の内容を簡潔かつ流暢な文章として生成できるようになり，ニュース要約，文書検索支援，専門文書の理解補助など，多様な応用が期待されている．しかしその一方で，LLM による要約生成には，入力文書に含まれない内容や事実と異なる情報を生成してしまうハルシネーションと呼ばれる問題が依然として存在する．要約におけるハルシネーションは発見が難しく，入力文書への忠実性が損なわれることで，ユーザーの誤解や誤った意思決定につながる恐れが指摘されている [1]．

従来の要約手法は，単一モデルによる一回限りの生成が主流であり，生成結果の妥当性や事実整合性が十分に検証されないままであった．近年になり，最終的な要約を出力する前に，生成途中または生成結果に対して追加の評価を行い，その結果を用いて要約を修正・改善する手法も提案されている [2]．この手法は，単一モデルによる一回限りの生成に比べ，生成内容を自己点検・再考する点で有効である一方，単一の評価観点・単一のプロンプトによって要約全体を判断しているため，数値情報の誤り，固有表現の不一致，照応関係の破綻など，性質の異なるハルシネーションを同時に十分捉えることが難しく，修正の根拠が曖昧になる場合がある．

このような背景から，本研究では要約生成において，評価・修正過程を単一のものとして扱うのではなく，複数の観点に基づく検証を明確に分離し，複数の検証結果を総合して適用する枠組みを提案する．具体的には，事実整合性，数値および時間情報の一致性，固有表現の一致性，照応関係の一致性といった異なる観点に着目した複数の検証エージェントを導入し，各エージェントが指摘した問題を統合的に扱うことで，ハルシネーションの発生個所をより精密に修正することを目指す．さらに，検証結果を一括して要約全体に反映させるのではなく，必要な修正に限定して反復的に要約を更新することで，内容の過度な更新を避けつつ忠実性の向上を図る．

2　関連研究

LLM による要約におけるハルシネーションの問題に対し，近年は評価手法および抑制手法の両面から研究が進められている．本節では，要約の忠実性評価に関する研究と，生成過程に検証を組み込む手法に焦点を当てる．

要約の忠実性を向上させる代表的な反復的修正手法として，SummIt [2] が挙げられる．SummIt は，ChatGPT を用いて要約の生成・自己評価・修正を繰り返すことで忠実性を高めるフレームワークである．しかし，SummIt の修正プロセスは要約全体に対する汎用的なフィードバックに依存しており，本研究が着目するような「数値情報の誤り」や「照応関係の破綻」といった特定の性質を持つハルシネーションを個別に切り分けて検証・修正する仕組みにはなっていない．

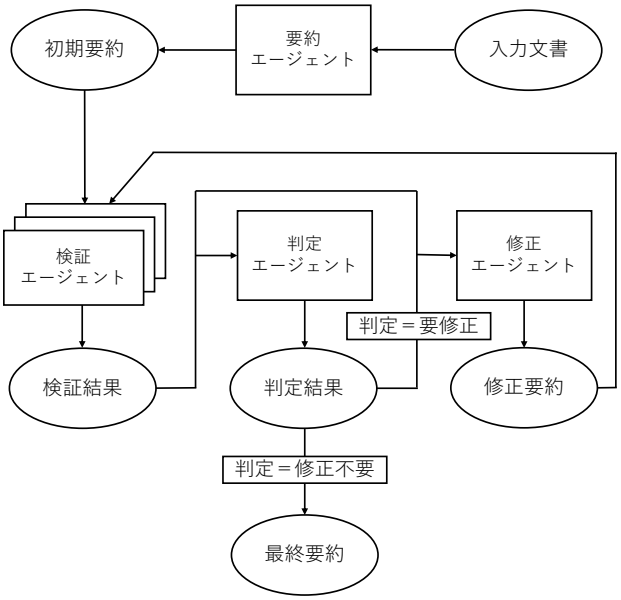

図 1　本アーキテクチャの概略図

生成プロセスにおいてモデルの内部状態を利用する手法としては，HALLUCANA [3] が提案されている．HALLUCANA は LLM の隠れ層にある内部表現を利用し，生成中あるいは生成前にハルシネーションを検知して介入を行う．この手法は忠実性の向上に有効であるが，モデルの内部パラメータへのアクセスを前提としているため， API モデルへの適用が困難であるという制約がある．

また, 事後的な検証手法として Chain-of-Verification (CoVe) [4] が提案されている．CoVe は生成された内容に対して検証質問を立案・回答することで自己修正を行うが，基本的には一連の検証ステップを一度に行う設計となっている．

これらの研究に対し，本研究はハルシネーションを事実整合性，数値・時間時間情報の一致性，固有表現の一致性，照応関係の一致性といった複数の具体的な観点に分解し，それぞれに特化した検証エージェントを導入する点に特徴がある．各観点からの指摘を独立に抽出し，それらを統合して忠実性に特化した精密な修正を反復的に適用することで，既存手法では困難であった，要約文に含まれる個々の事実誤りを観点ごとに切り分け，誤りのある個所のみを特定して修正することを目指す．

3　手法

提案手法の全体構成を図 1 に示す．本フレームワークでは，まず入力として与えられた元記事から要約エージェントにより初期要約を生成する．生成された要約は検証エージェントに入力され，複数の検証エージェントがそれぞれ異なる観点から要約の問題点を出力する．その後，判定エージェントが出力された問題点に基づいて要約の修正が必要か否かを判断する．判定の結果，要約に修正が必要であると判断された場合には，修正エージェントが検証結果をもとに要約を改善する．修正後の要約は再び検証エージェントに入力され，同様の検証および判定プロセスが繰り返される．一方で，判定エージェントによって修正が不要であると判定された場合には，その要約を最終要約として出力する．

このように検証観点を分離することで，単一モデルによる一括判定と比べて，誤りの見落としを低減し，判定過程の解釈性を高めることを目的とする．

3.1　検証エージェント

本研究では，要約に含まれる代表的な事実誤りの種類に対応するため，以下の 4 種類の検証エージェントを設計した．すべての検証エージェントは，元記事と要約を入力とし，該当する不整合や問題点を自然言語で列挙する．

Faithfulness Checker：　要約中の記述のうち，元記事によって直接裏付けられていない主張を検出する検証エージェントである．具体的には，要約が記事に含まれない新たな事実や推測を付加していないか，あるいは記事中で明示されていない因果関係や評価を含んでいないかを確認し，根拠が不十分な主張を問題点として列挙する．

Numeric Consistency Checker：　数値情報の整合性に着目した検証エージェントである．記事中に出現する日付，数量，割合，金額などの数値表現と，要約中の対応する数値が一致しているかを検証し，不一致や誤った変換が見られる場合に指摘する．

Named Entity Consistency Checker：　人名，地名，組織名などの固有名詞の不一致を検出する検証エージェントである．要約中に出現する固有名詞が，記事中に存在しないものや，記事中の別の固有表現と混同されている場合に，それを問題点として報告する．

Pronoun Consistency Checker：　要約中の代名詞の参照関係の妥当性を検証する検証エージェントである．具体的には，「彼」「彼女」「それ」などの代名詞が指している対象が記事中に明確に存在するか，また参照先が曖昧または誤っていないかを判定し，不整合がある場合に指摘する．

3.2 判定エージェント

各検証エージェントの出力をもとに，要約全体として修正が必要かどうかを判定するために，判定用の LLM を用いる．本研究では，検証エージェントの出力の統合方法および判定手順の違いにより，以下の 4 通りのアーキテクチャを検討する．

まず，検証エージェントの出力を一つの判定 LLM にまとめて入力する方式（unite）と，各検証エージェントの出力をそれぞれ独立した判定 LLM に入力し，最終的に AND 条件で統合する方式（divide）の 2 通りを比較する．次に，判定 LLM に入力する情報として，検証エージェントの出力をそのまま用いる方式（raw）と，抽出用 LLM を用いて検証エージェント出力から修正が必要な課題点のみを抽出した上で判定 LLM に入力する方式（extracted）の 2 通りを検討する．以上より，判定方法としては $2 \times 2 = 4$ 通りのアーキテクチャーが存在する．

本研究では，各データセットの Validation set 1000 件を用いてこれら 4 通りの構成を比較し，最も性能の高かったアーキテクチャを最終的な手法として採用する．また，計算量および実用性を考慮し，各要約に対する最大ループ回数は 3 回に制限した．

4 実験

4.1 実験設定

本研究では，要約生成および各種検証・修正を行う基盤モデルとして `Llama-3.1-8B-Instruct` を用いた．生成時の確率的ばらつきを排除し，各アーキテクチャ間の差異を公平に比較するため，すべての実験において温度パラメータは 0 に設定した．実験では，まず入力記事から初期要約（Draft）を一度生成し，その共通の初期要約を起点として，各アーキテクチャによる検証および修正プロセスを適用した．なお，初期要約の生成に用いるプロンプト設計および生成手順は，先行研究である SummIt [2] に従った．

4.2 データセット

評価には，事実一貫性および要約難易度の異なる複数の代表的データセットを用いた．具体的には，ニュース記事要約として CNN/DailyMail（CNN/DM）[5] および XSum [6]，対話要約として DialogSum [7]，長文科学文書要約として arXiv [8] を使用した．使用した各データセットのサンプルサイズおよび文書・要約長の統計情報を表 1 に示す．

表 1 評価に使用したテストデータセットの統計情報

Dataset	#Samples	平均文書単語数	平均要約単語数
CNN/DM	1,000	683.0	55.6
XSum	1,000	380.0	21.1
DialogSum	1,000	134.7	18.8
arXiv	1,000	5,737.7	163.1

4.3 評価指標

生成要約の品質評価には，忠実性評価指標として FactCC [9]，FactKB [10]，SummaC [11]，MiniCheck [12]，AlignScore [13] を採用した．なお，FactCC および FactKB は入力長が最大 512 トークンに制限されているため，長文文書を対象とする arXiv データセットに対しては，評価を行っていない．また，要約の内容網羅性を測るために，ROUGE [14] スコアを併せて報告する．

4.4 実験結果

まず，各データセットの Validation set を用いて，提案手法における 4 通りのアーキテクチャ（unite/divide × raw/extracted）を比較し，SummaC, MiniCheck, AlignScore のスコアに基づいて最終構成を決定した．その結果，CNN/DM では extracted-divide，XSum では raw-unite，DialogSum では extracted-unite，arXiv では raw-divide が最良となった．以降の評価では，データセットごとに選定されたこれらの構成を Ours として用いた．ただし，arXiv 以外のデータセットでは 4 通りのアーキテクチャでスコア，最終要約までのループ数の分布，共に大きな差は見られなかった．

Test set における自動評価結果を表 2 に示す．比較対象として，初期要約（Draft）および反復的修正手法である SummIt を用いた．全体として，Ours は複数のデータセットにおいて忠実性評価指標の改善を示しており，特に SummaC，MiniCheck，AlignScore において向上が確認されるケースが多い．一方で，ROUGE スコアは必ずしも一貫して向上せず，CNN/DM, Xsum, DialogSum では基本的に修正前の Draft の方が高くなった．

CNN/DM では，Ours は SummaC, MiniCheck, AlignScore を改善し，ROUGE は Draft および SummIt と同程度の値となった．XSum では，Ours が FactCC, FactKB，MiniCheck，AlignScore を改善した．Dialog-

表 2　実験結果：忠実性評価指標，ROUGE，および平均出力単語数（Len）の比較　（FactCC および FactKB は入力長制限のため，arXiv に対しては評価を行っていない．）

Dataset	Method	FactCC	FactKB	Summac	MiniCheck	AlignScore	R-1	R-2	R-L	Len
CNN/DM	Draft	**17.96**	**98.46**	41.01	73.93	87.62	37.71	**14.97**	**23.88**	111.1
	SummIt	17.68	98.38	40.71	73.80	87.30	**37.72**	14.89	23.85	109.6
	Ours	16.87	98.41	**41.23**	**74.60**	**87.72**	37.56	14.91	23.77	111.2
XSum	Draft	23.39	65.81	23.33	75.54	88.15	**28.77**	**7.84**	**20.51**	34.0
	SummIt	22.79	66.38	**23.90**	75.00	88.10	28.35	7.71	20.17	36.8
	Ours	**25.03**	**69.53**	23.80	**77.00**	**88.45**	27.68	7.41	19.61	36.0
DialogSum	Draft	15.07	92.25	25.79	**62.04**	**90.21**	**28.97**	**10.36**	**22.17**	55.4
	SummIt	15.06	**92.28**	25.80	61.96	90.21	28.91	10.34	22.13	55.5
	Ours	**17.02**	86.78	**26.73**	61.84	89.73	28.19	10.04	21.56	59.1
arXiv	Draft	–	–	**59.24**	50.15	80.76	34.30	13.98	19.20	455.0
	SummIt	–	–	58.42	50.92	80.46	34.99	14.09	19.57	431.4
	Ours	–	–	59.16	**51.28**	**81.19**	**35.48**	**14.47**	**19.74**	410.9

Sum では，Ours が FactCC および SummaC を改善したが，MiniCheck，AlignScore，ROUGE では Draft または SummIt が高い値を示した．arXiv では，Ours が MiniCheck，AlignScore で最良の結果を示した．

以上から，提案手法はデータセットによって挙動に差はあるものの，複数の条件において忠実性評価指標の改善を達成していることが確認できる．

4.5　アブレーション結果

提案手法における各検証エージェントの寄与を分析するため，アブレーション実験を実施した．具体的には，各検証エージェントを単独で用いた構成と，全検証エージェントを用いた構成から一つずつ検証エージェントを除外した構成の，合計 8 通りの設定について比較を行った．定量的な評価結果の詳細は付録 A に示す．

アブレーション実験より，すべての検証エージェントを用いた構成は多くの指標で安定した性能を示した．一方，XSum と DialogSum においては，一部の意味的忠実性指標で Numeric Checker を除外した構成が高いスコアを示す傾向が見られた．また，単一検証エージェントのみを用いた場合，性能は全体的に低下する傾向にあった．

5　考察

本手法では，初期要約を起点として，検証・判定・修正を反復的に適用する．本節では，初期要約に対する最初のループ（loop 1）に着目し，修正要否の判定結果と要約品質の関係を分析する．loop 1 における OK/NG 判定別の初期要約のスコア（付録 B 参照）から，すべてのデータセットにおいて，OK と判定された要約は MiniCheck を中心とする忠実性指標で一貫して高い平均スコアを示した．この結果は，本研究で用いた検証エージェントが，入力文書との事実整合性という観点から，要約の品質を一定程度適切に識別できていることを示している．

一方で，実験結果では，検証エージェントによる判定が妥当であるにもかかわらず，最終的な要約性能の改善が限定的なケースも観測された．特に，一部のデータセットや評価指標においては，loop 1 で NG と判定された要約が必ずしも十分に改善されず，最終的な忠実性スコアが Draft や既存手法と大きく変わらない場合がある．このことは，本フレームワークにおける性能向上のボトルネックが，「誤りの検出」では必ずしもなく，「検出された問題点をどのように修正に反映するか」という修正過程に存在する可能性を示唆している．

以上より，本研究の結果は，複数の検証エージェントによる判定自体の有効性を支持する一方で，今後の課題として，修正用 LLM の設計や，指摘内容と修正操作の対応付けをより明示的に行う仕組みの必要性を示している．

6　おわりに

本研究では，要約に含まれるハルシネーションを抑制するため，事実整合性・数値/時間・固有表現・照応関係の 4 観点に特化した検証エージェントを導入し，指摘に基づいて反復的に修正する枠組みを提案した．複数データセットでの評価により，提案手法が忠実性指標を改善することを確認した．一方で，複数の判定エージェントを含めた反復処理により計算量は増加するため，今後は収束判定の改良やエージェントの軽量化，および人手評価による修正品質の分析を進める．

参考文献

[1] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. **ACM Trans. Inf. Syst.**, Vol. 43, No. 2, 2025.

[2] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. SummIt: Iterative text summarization via ChatGPT. In **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 10644–10657, 2023.

[3] Tianyi Li, Erenay Dayanik, Shubhi Tyagi, and Andrea Pierleoni. HALLUCANA: Fixing LLM hallucination with a canary lookahead. In **Findings of the Association for Computational Linguistics: NAACL 2025**, pp. 213–230, 2025.

[4] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 3563–3578, 2024.

[5] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In **Advances in Neural Information Processing Systems**, Vol. 28, 2015.

[6] Hayate Iso, Chao Qiao, and Hang Li. Fact-based Text Editing. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 171–182, 2020.

[7] Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. DialogSum: A real-life scenario dialogue summarization dataset. In **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 5062–5074, 2021.

[8] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 615–621, 2018.

[9] Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 9332–9346, 2020.

[10] Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. FactKB: Generalizable factuality evaluation using language models enhanced with factual knowledge. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 933–952, 2023.

[11] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 163–177, 2022.

[12] Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 8818–8847, 2024.

[13] Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 11328–11348, 2023.

[14] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, 2004.

A　Ablation 実験の結果

表 3　Ablation 実験の結果　（FactCC および FactKB は入力長制限のため，arXiv に対しては評価を行っていない.）

Dataset	Variant	FactCC	FactKB	Summac	MiniCheck	AlignScore	R-1	R-2	R-L
CNN/DM	w/ Faithfulness Checker	17.09	98.34	41.08	74.39	87.68	37.56	14.89	23.77
	w/ Numeric Checker	18.04	98.46	41.00	74.08	87.64	37.71	14.97	23.87
	w/ Entity Checker	17.73	**98.48**	41.19	74.20	87.72	37.71	**15.00**	23.89
	w/ Pronoun Checker	**18.13**	98.45	41.06	73.85	87.60	**37.72**	14.98	**23.89**
	Full model (Ours)	16.87	98.41	41.23	**74.60**	87.72	37.56	14.91	23.77
	w/o Faithfulness Checker	17.42	98.47	41.10	74.02	87.61	37.69	14.98	23.87
	w/o Numeric Checker	16.94	98.41	41.23	74.58	**87.76**	37.55	14.90	23.76
	w/o Entity Checker	17.02	98.34	41.09	74.49	87.68	37.58	14.91	23.78
	w/o Pronoun Checker	16.86	98.41	**41.24**	74.57	87.73	37.58	14.92	23.79
XSum	w/ Faithfulness Checker	23.57	66.93	23.75	76.36	88.78	28.44	7.76	20.32
	w/ Numeric Checker	23.31	65.94	23.40	75.58	88.25	28.73	**7.84**	20.48
	w/ Entity Checker	23.30	66.07	23.38	75.88	88.39	28.74	7.83	20.50
	w/ Pronoun Checker	23.57	65.72	23.33	75.57	88.09	**28.76**	**7.84**	**20.51**
	Full model (Ours)	**25.03**	**69.53**	23.80	77.00	88.45	27.68	7.41	19.61
	w/o Faithfulness Checker	23.73	66.89	23.55	75.95	88.31	28.58	7.79	20.38
	w/o Numeric Checker	24.51	69.42	23.87	76.97	**88.90**	27.61	7.37	19.73
	w/o Entity Checker	24.05	68.48	**24.01**	77.21	88.86	27.88	7.42	19.94
	w/o Pronoun Checker	24.01	68.52	23.82	**77.59**	88.46	27.71	7.34	19.85
DialogSum	w/ Faithfulness Checker	15.34	88.78	25.95	61.86	90.18	28.63	10.22	21.93
	w/ Numeric Checker	15.19	91.76	25.80	62.09	90.18	28.97	**10.34**	**22.17**
	w/ Entity Checker	15.02	91.99	25.88	61.89	90.20	28.92	**10.34**	22.14
	w/ Pronoun Checker	15.01	**92.09**	25.78	61.91	90.29	**28.94**	10.33	22.14
	Full model (Ours)	**17.02**	86.78	**26.73**	61.84	89.73	28.19	10.04	21.56
	w/o Faithfulness Checker	14.84	90.72	26.09	61.96	90.18	28.88	10.30	22.07
	w/o Numeric Checker	16.78	87.84	26.19	**62.54**	**90.46**	28.40	10.02	21.65
	w/o Entity Checker	16.92	88.85	26.19	61.63	89.98	28.45	10.11	21.74
	w/o Pronoun Checker	16.32	87.32	26.17	62.28	90.03	28.31	10.04	21.65
arXiv	w/ Faithfulness Checker	–	–	59.12	51.21	81.16	**35.54**	**14.47**	**19.76**
	w/ Numeric Checker	–	–	59.22	50.15	80.75	34.30	13.98	19.20
	w/ Entity Checker	–	–	**59.24**	50.15	80.76	34.30	13.98	19.20
	w/ Pronoun Checker	–	–	**59.24**	50.15	80.76	34.30	13.98	19.20
	Full model (Ours)	–	–	59.16	**51.28**	**81.19**	35.48	14.47	19.74
	w/o Faithfulness Checker	–	–	**59.24**	50.15	80.76	34.30	13.98	19.20
	w/o Numeric Checker	–	–	59.15	51.21	81.17	35.47	14.47	19.74
	w/o Entity Checker	–	–	59.16	**51.28**	**81.19**	35.48	14.47	19.74
	w/o Pronoun Checker	–	–	59.12	51.20	81.16	**35.54**	**14.47**	**19.76**

B　検証エージェントの判定の妥当性

表 4　loop1 における OK/NG 判定別初期要約スコア比較　（FactCC および FactKB は入力長制限のため，arXiv に対しては評価を行っていない.）

Dataset	判定	FactCC	FactKB	SummaC	MiniCheck	AlignScore	ROUGE-1	ROUGE-2	ROUGE-L
CNN/DM	loop1 OK	**20.00**	**98.46**	**41.19**	**75.21**	**88.30**	37.40	14.93	**23.96**
	loop1 NG	15.46	**98.46**	40.75	72.32	86.76	**38.08**	**15.00**	23.75
XSum	loop1 OK	**24.46**	**68.05**	**23.95**	**79.98**	87.59	28.46	**7.87**	20.16
	loop1 NG	22.60	64.04	22.85	72.12	**88.56**	**29.02**	7.83	**20.78**
DialogSum	loop1 OK	14.79	**93.26**	**25.89**	**63.88**	90.14	28.86	**10.59**	**22.21**
	loop1 NG	**15.57**	90.57	25.64	59.10	**90.37**	**29.19**	10.00	22.12
arXiv	loop1 OK	–	–	**60.29**	**52.89**	**82.04**	**34.95**	**14.06**	**19.44**
	loop1 NG	–	–	58.22	47.60	79.68	33.86	13.96	19.03