マスク予測モデルを用いた軽量なハルシネーションのスパン検出手法

山田美優 [1]　荒瀬由紀 [1]
[1] 東京科学大学
yamada.m.ee1b@m.isct.ac.jp, arase@c.titech.ac.jp

概要

大規模言語モデル (LLM) の生成における hallucination が問題となっている。既存研究の多くは、テキスト全体に hallucination が含まれるか否かの二値分類を行っているが、長いテキストに対してはより詳細な判断が必要である。しかし、hallucination の箇所検出には大規模な LLM を用いる手法が多く、軽量なモデルによる検出が求められている。本研究では、詳細な hallucination 検出を軽量なモデルで可能にするために、マスク予測モデルを用いたスパン検出手法を提案する。検出には hallucination が入力文章から推論できない情報であることを利用する。出力文章をチャンクに分割し、各チャンクをマスクして入力文章と照合してマスク予測を行うことで hallucination のスパン検出を行う。実験では fine-tuning によって hallucination スパンの検出性能を通常のトークン分類よりも向上させ、高い再現率を達成できることが示された。

1　はじめに

大規模言語モデル（LLM）は、生成タスクにおける高い汎用性によって急速に普及した。一方で、事実に基づかない情報を生成する hallucination 問題は依然として深刻な課題である [1]。LLM の回答は流暢であるため、ユーザーがその真偽を判別することは困難であり、誤情報を信じて拡散してしまう危険性が高い。そのため、hallucination を含む情報を見逃さずに検出することが重要であり、特に誤情報の流通を防ぐ観点からは再現率の高い検出が求められる。

既存の hallucination 検出手法の多くは、判断対象のテキストに hallucination が含まれているか否かの二値分類手法がほとんどである [2, 3]。しかし、判断対象のテキストが長い場合、テキスト単位の二値分類では文章の事実性を計測するのには不十分である [4]。そのため、より詳細な事実性の評価を行うために、本研究ではテキスト中の hallucination が含まれるスパンを特定する手法を提案する。hallucination のスパン検出にはパラメータ数の大きい LLM が用いられることが多い [5, 6, 7]。しかし、hallucination のスパン検出をより広く実用化するためには、高速な計算が可能な軽量なモデルで検出を行うことが望ましい。そのため、本研究では軽量なエンコーダモデルのマスク予測を用いて hallucination のスパン検出を行う手法を提案する。

本研究では、LLM による条件付き言語生成を対象とし、hallucination を入力に基づかない、または入力と反する情報を出力したスパンと定義する [6]。本研究では「入力文章に基づいて生成された内容であれば、その一部を隠しても入力文章から再構成できる」という性質を利用する。具体的には、出力文章中の各チャンクをマスクし、入力文章に照合することでマスクスパンを予測させ、hallucination のスパン検出を行う手法を提案する。予測されたスパンが元のチャンクと異なる場合、入力文章から推論できる内容と出力文章に記述されている内容が食い違っていることになるため、そのチャンクは hallucination を含むと判断する。

実験の結果、使用したモデルは軽量なものであったが、hallucination のスパン検出において再現率を向上し、有効性が示された。本研究の実験に用いたコードは以下のレポジトリにて公開している。(https://github.com/miyu-y/nlp2026_spandetect)

2　提案手法

入力文章に基づいて生成されたチャンクであれば、そのチャンクをマスクしても、入力文章中の情報から同様の内容を予測できると考えられる。一方で、hallucination は入力文章に含まれない情報で

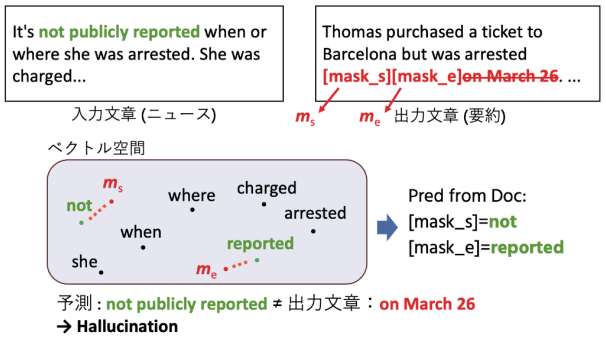

図 1　提案手法の概要

あるため、入力文章から対応する内容を予測することは困難である。提案手法では出力文章をチャンクに分割し（2.1 節）、図 1 に示す通り各チャンクをマスクして入力文章と照合するマスク予測を行う（2.2 節）。そしてマスク予測の際に得られた値を特徴量として用い、線形分類器でそのチャンクが hallucination を含むか否かを判定する（2.3 節）。

2.1　チャンク分割

どの範囲をマスクするかを決定するために、テキストをチャンク単位に分割する必要がある。トークン単位など細かい分割では推論回数が増えすぎてしまい、文単位など粗い分割では検出の細かさが十分ではない。そこで以下の例に示すように、テキストのチャンク分割に SRL (Semantic Role Labeling : 意味役割分析) を用いる。SRL は、テキスト中の述語とその引数を抽出するものであり、テキストの意味構造を捉えることができるため、hallucination スパン検出でも用いられている [8]。

- 原文 : Keonna Thomas was charged with attempting to travel to Syria.
- 分割結果 : [Keonna Thomas] [was charged] [with attempting] [to travel] [to Syria] .

文中に複数の動詞が含まれる場合、それらすべての分析結果を統合して、最も細かいチャンク分割を行う。具体的には、ある分析結果で分割されたチャンクが他の分析結果でさらに分割されている場合、より細かい分割の方を採用するものとする。どの分析結果にも含まれなかったトークン (接続語など) は、マスク対象から除外する。また、受動態や進行形等の動詞二つが連続する場合など、分割した結果マスク対象のチャンクが短すぎる場合は、隣接するチャンクと結合する。

2.2　マスク予測による入出力文章の照合

マスクした出力文章中のチャンクを入力文章に照合するため、NPM (NonParametric Masked Language Model) [9] に基づくマスク予測手法を提案する。

2.2.1　マスクスパン予測

マスクした出力文章中のチャンクと入力文章を照合することで、チャンクに対応する可能性のあるスパン候補を獲得する。まず出力文章内のチャンクを 2 つの<mask>トークンに変換する。先頭のトークンを<mask_s>、末尾のトークンを<mask_e>と表記する。次に、エンコーダで出力文章全体の埋め込みを作成し、それぞれの<mask>トークンに対応する埋め込みを得る $(\boldsymbol{m}_s, \boldsymbol{m}_e)$。続いてエンコーダで入力文章の各トークンの埋め込みを得る $(\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_i, \ldots, \boldsymbol{d}_n)$。

これら埋め込みを用い、以下の手順でマスクスパンを予測する。まず、各<mask>トークンの埋め込み $(\boldsymbol{m}_s, \boldsymbol{m}_e)$ と、入力文の各トークンの埋め込み $\boldsymbol{d}_k$ との類似度を計算する。ここで、類似度計算には以下のコサイン類似度を用いる。

$$\mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}||||\boldsymbol{y}||} \tag{1}$$

次に、スパンの開始位置 i と終了位置 j のすべての組み合わせ（ただし $i \leq j$）に対し、以下の予測スコア pred_score を算出する。

$$\text{pred\_score} = \mathrm{sim}(\boldsymbol{m}_s, \boldsymbol{d}_i) + \mathrm{sim}(\boldsymbol{m}_e, \boldsymbol{d}_j) \tag{2}$$

このスコアを最大化するインデックスの組 (i^*, j^*) を選択し、対応するトークン列 $\boldsymbol{c}_{pred}$ を対応する入力スパンの候補とする。

$$\begin{aligned} i^*, j^* &= \underset{i \leq j}{\mathrm{argmax}}(\text{pred\_score}) \\ \boldsymbol{c}_{pred} &= (\boldsymbol{d}_{i^*}, \boldsymbol{d}_{i^*+1}, \ldots, \boldsymbol{d}_{j^*}) \end{aligned} \tag{3}$$

2.2.2　Hallucination スパンへの適応

Hallucination スパン検出では、hallucination を含むスパンと含まないスパンが混在する。そこでマスクしたチャンクが hallucination を含む場合と含まない場合に適応するように fine-tuning を行う。Hallucination を含まないチャンクの場合は、NPM と同様にマスクしたチャンクに対応するスパンを入力文章から正しく予測できるように訓練する。一方で、hallucination を含むチャンクの場合は、対応するスパンが入力文章に存在しないはずである。そこ

でマスクしたチャンクとは異なる情報を含むスパンを予測するように訓練する。

この学習を実現するため、以下の手順で Fine-tuning 用データを構築する。出力文章の各チャンクについて、入力文章中の類似したチャンクを (Y^+)、そうでないチャンクを (Y^-) とする。類似したチャンクは、チャンクの 2/3 以上の n-gram が一致しているものとする。[1)] さらに、Y^+ および Y^- に含まれる各チャンクの開始トークンと終了トークンを、それぞれ Y^+_s, Y^+_e および Y^-_s, Y^-_e としてまとめる。

訓練データに含まれるチャンクのうち、Y^+ の要素数が 0 でないものを用いて fine-tuning を行う。Fine-tuning によって、hallucination を含まないチャンクをマスクした場合は、Y^+ のチャンクが Y^- のチャンクよりも類似度が高くなるように、hallucination を含むチャンクをマスクした場合は、Y^- のチャンクが Y^+ のチャンクよりも類似度が高くなるようにする。損失関数は以下の margin ranking loss を用いる（γ はマージンの値）。

$$L = (1-\ell)\cdot(L^p) + \ell\cdot(L^n),$$

$$L^p = \sum_{t\in\{s,e\}} \max\Big(0,\ \gamma - \big[\max_{y\in Y^+_t} \mathrm{sim}(\boldsymbol{m}_t, y) - \max_{y\in Y^-_t} \mathrm{sim}(\boldsymbol{m}_t, y)\big]\Big),$$

$$L^n = \sum_{t\in\{s,e\}} \max\Big(0,\ \gamma - \big[\max_{y\in Y^-_t} \mathrm{sim}(\boldsymbol{m}_t, y) - \max_{y\in Y^+_t} \mathrm{sim}(\boldsymbol{m}_t, y)\big]\Big).$$

ここで、ℓ はマスクしたチャンクが hallucination を含む場合に 1、含まない場合に 0 となるラベルである。

2.3 Hallucination スパンの判定

予測したマスクスパンについて、それらが hallucination を含むか否かを線形分類器を用いて分類する。pred_score が上位 5 つの予測スパンの中から、出力チャンクとの類似度が最も高いものを分類に用いる。スパン・チャンクの埋め込みは、それぞれの各トークンの埋め込みの平均とし、類似度の計算は式 1 を用いる。分類器の特徴量として以下のものを用いる。

予測スコア　式 2 で計算される pred_score を特徴量として用いる。pred_score の値はチャンク内の最大値を用い、値が低いほどマスクスパンを埋めにくいことを意味し、hallucination を含む可能性が高い。

予測スパンとチャンクの類似度　予測スパンとチャンクの類似度が低いほど内容が異なることを示唆し、入力文章から推論した内容と食い違うことになり、hallucination を含む可能性が高い。

その他の特徴量　以下の特徴量も併せて用いる。

- マスクされたチャンクのトークン長
- マスクされたチャンクの開始位置と終了位置
- マスクされたチャンクが何番目の文か

類似度はチャンクが長いほど小さくなる傾向があるため、チャンクのトークン長を特徴量として用いる。また、一般的に hallucination は出力文章の後半に多く発生する傾向があるため [10]、チャンクの開始位置と終了位置、チャンクが何番目の文かも特徴量として用いる。

1) Y^+ を出力文章中のチャンクと完全一致のものに限定すると、Y^+ の要素数が少なくなってしまったり、0 になってしまうチャンクが多かったため、類似しているチャンクを含めるようにした。

表 1　データセットの事例数 (() 内は hallucination の割合)

	QA	要約	全体
Train (FT)	4, 134 (8.6%)	3, 858 (3.0%)	7, 992 (5.9%)
Train (分類)	500 (9.9%)	500 (3.5%)	1, 000 (6.7%)
Dev	400 (9.0%)	400 (3.3%)	800 (6.1%)
Test	160 (22.4%)	204 (12.9%)	364 (17.7%)

3 実験

提案手法の有効性を評価するため RAGTruth[6] を用いて評価実験を行った。

3.1 データセット

RAGTruth には 1 つの入力文章に対してそれぞれ異なる 6 つの LLM[11, 12, 13] が生成した文章が含まれており、hallucination スパンのラベルが人手アノテーションによって付与されている。

データセットは QA、Data-to-text、ニュース要約の 3 つのタスクから構成されており、今回は QA とニュース要約のみを実験に使用した。[2)] QA タスクの入力文章は MS MARCO[14] に含まれる passage と question、出力は answer である。ニュース要約タスクの入力文章は CNN/Daily Mail dataset[15] などに含まれるニュース記事、出力はその要約である。

RAGTruth の訓練データから、線形分類器の訓練用に 1, 000 事例 (各タスク 500 事例ずつ)、検証データとして 800 事例 (各タスク 400 事例ずつ) を抽出し、残りを fine-tuning 用の訓練データとして使用した。RAGTruth のテストデータの中には hallucination が全く含まれていないものもあるが、本実験では hallucination を含む事例のみを用いた。表 1 に各データセットの事例数と hallucination の文字単位の

2) Data-to-text タスクは入力文章が json 形式と特殊なため、今回は使用しなかった。

表 2　各手法の hallucination スパン検出性能 (P: Precision, R: Recall, F1: F1 スコア)

手法	QA			要約			全体		
	P	R	F1	P	R	F1	P	R	F1
Llama	70.1	35.6	**47.2**	75.3	37.9	**50.4**	72.0	36.5	**48.4**
Token 分類	**79.5**	31.1	44.7	**86.0**	1.9	3.7	**79.7**	20.4	32.5
提案手法	32.1	**64.6**	42.9	22.9	**51.4**	31.7	28.5	**59.8**	38.6

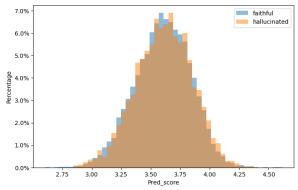

図 2　Fine-tuning 前の pred_score の分布

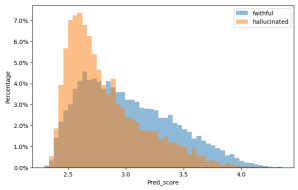

図 3　Fine-tuning 後の pred_score の分布

割合を示す。

3.2　提案手法の実装

SRL によるチャンク分割には AllenNLP の BERT-based SRL モデルを用いた [16]。Fine-tuning には ModernBERT-large (0.4B) [17] を用い、学習率 1e-5、エポック数 5 で学習を行った。各事例について、出力文章のチャンクで Y^+ の要素数が 0 でないものを、全チャンクのうち上限 15%までランダムに抽出してマスクし fine-tuning を行った。損失関数の margin の値は 0.2 とした。

3.3　比較手法

比較手法として、ModernBERT-large のトークン分類モデルを用いたハルシネーションスパン検出の性能を評価した。入力文章と出力文章を結合して入力し、出力文章の各トークンが hallucination を含むか否かを分類する。学習率 1e-6、エポック数 5 で学習を行った。

さらに、パラメータ数の大きい LLM の性能と比較するために Llama-3.1-8B-Instruct[18] を fine-tuning して hallucination スパン検出を行った。入力文章と出力文章を入力し、出力文章中の hallucination スパンを json 形式で出力させる。プロンプトは AppendixB に示す。学習率 2e-5、エポック数 1 で学習を行った。[3)] いずれのモデルにおいても、Early Stopping を用いて、検証データの loss が 2 エポック連続で改善しなかった場合に学習を停止した。

4　実験結果と分析

表 2 に各手法の hallucination スパン検出性能を示す。出力文章内のあるチャンクが hallucination を含むと判定された場合、そのチャンク内の全文字を hallucination とみなす。評価指標は RAGTruth に倣い、文字単位の Precision、Recall、F1 スコアとする。

表 2 より、Token 分類手法および Llama はいずれも高い Precision を示した一方で、Recall が低い結果となった。一方で、提案手法は高い Recall を示し、F1 スコアも全体で Token 分類手法より 6.1 ポイント高い。QA タスクにおいては、ModernBERT はパラメータ数が Llama の 20 分の 1 程であるにもかかわらず、Llama に対し良好な F1 スコアを達成している。提案手法は Precision が低いが、hallucination スパンを見逃しにくい手法であると言える。この Precision の低下は、チャンク単位で判定を行っていることに起因すると考えられる。具体的には、部分的に hallucination を含む場合でも、チャンク全体が誤検出として評価されるため、Precision が低下する傾向がある。

図 2 と図 3 に、fine-tuning 前後の pred_score の分布を示す。Fine-tuning 前は両図とも 2 つの分布が完全に重なっていたが、fine-tuning 後は hallucination を含むチャンク（オレンジ色）と含まないチャンク（青色）で分布に顕著な違いが見られるようになった。Fine-tuning によって、hallucination を含まないチャンクのスコアは高い値に維持したまま、hallucination を含むチャンクのスコアが低くなるようにすることを目指していた。しかし、Hallucination を含まないチャンクのスコアも低下する傾向となっており、これも Precision が低い原因の一つであると考えられる。一方で、fine-tuning により、hallucination を含むチャンクのスコア分布が低い値にシフトし、高い Recall が達成されたと考えられる。

5　おわりに

本研究では、出力文章中のチャンクを入力文章に照合するマスク予測モデルによる hallucination のスパン検出手法を提案した。実験の結果、提案手法は Token 分類手法およびパラメータ数の大きい Llama モデルと比較して顕著に高い Recall を達成した。一方で性能はまだ十分ではなく、今後は、損失関数や予測手法の改善を行い、性能向上を目指す。

3)　RAGTruth の論文内の実験で用いられている設定に従った。

謝辞

本研究は、JST 経済安全保障重要技術育成プログラム JPMJKP24C3 の支援を受けたものである。本研究は、東京科学大学のスーパーコンピュータ TSUBAME4.0 を利用して実施した。

参考文献

[1] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. **arXiv preprint arXiv:2307.10169**, 2023.

[2] Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. On large language models' hallucination with regard to known facts. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, 2024.

[3] Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. Refchecker: Reference-based fine-grained hallucination checker and benchmark for large language models. **arXiv preprint arXiv:2405.14486**, 2024.

[4] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, 2023.

[5] Hsuan Su, Ting-Yao Hu, Hema Swetha Koppula, Kundan Krishna, Hadi Pouransari, Cheng-Yu Hsieh, Cem Koc, Joseph Yitan Cheng, Oncel Tuzel, and Raviteja Vemulapalli. Learning to reason for hallucination span detection. **https://arxiv.org/abs/2510.02173**, 2025.

[6] Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2024.

[7] Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. In **First Conference on Language Modeling**, 2024.

[8] Passant Elchafei and Mervat Abu Elkheir. Hallucination detectives at SemEval-2025 task 3: Span-level hallucination detection for LLM-generated answers. In **Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)**, 2025.

[9] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In **Findings of the Association for Computational Linguistics: ACL 2023**, 2023.

[10] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. **https://arxiv.org/abs/2305.13534**, 2023.

[11] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal et al. Anadkat. Gpt-4 technical report. **arXiv preprint arXiv:2303.08774**, 2023.

[12] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, and Lucile et al. Saulnier. Mistral 7b. **arXiv preprint arXiv:2310.06825**, 2023.

[13] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti et al. Bhosale. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023.

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. 2016.

[15] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer- generator networks. In **In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2017.

[16] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. 2019.

[17] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. **https://arxiv.org/abs/2412.13663**, 2024.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. **https://arxiv.org/abs/2407.21783**, 2024.

表 3　タイプ別の hallucination チャンク数

タイプ	QA	要約	全体
Evident Conflict	2,349	5,812	8,161
Subtle Conflict	0	339	339
Evident Baseless Info	23,441	10,820	34,261
Subtle Baseless Info	5,545	1,020	6,565

表 4　タイプ別の検出性能 (Recall)

タイプ	QA	要約	全体
Evident Conflict	42.9	40.9	41.5
Subtle Conflict	-	41.6	41.6
Evident Baseless Info	68.8	63.9	67.3
Subtle Baseless Info	73.2	76.5	73.7

表 5　Llama-3.1-8B-Instruct で用いた hallucination スパン検出のプロンプト

QA

Below is a question:
[質問]
Below are related passages:
[入力文章]
Below is an answer:
[出力文章]
[指示]

要約

Below is the original news:
[入力文章]
Below is a summary of the news:
[出力文章]
[指示]

指示

Your task is to determine whether the answer contains either or both of the following two types of hallucinations:
1. conflict: instances where the answer presents direct contraction or opposition to the passages;
2. baseless info: instances where the answer includes information which is not substantiated by or inferred from the passages.
Then, compile the labeled hallucinated spans into a JSON dict, with a key "hallucination list" and its value is a list of hallucinated spans. If there exist potential hallucinations, the output should be in the following JSON format: {{“hallucination list”: [hallucination span1, hallucination span2, ...]}}. Otherwise, leave the value as a empty list as following: {{“hallucination list”: []}}.
Output:

A　Hallucination のタイプ別の分析

本研究で用いた RAGTruth データセットには、hallucination のタイプ別のラベルが付与されている。本節では、提案手法の hallucination スパン検出性能をタイプ別に分析する。RAGTruth データセットにおける hallucination のタイプは以下の 4 つである。

- Evident Conflict : 数値や名前の間違いなど、入力と明らかに異なる情報
- Subtle Conflict : 入力が意図した意味とは異なる情報の提供、入力とニュアンスが異なる記述など
- Evident Introduction of Baseless Information : 入力に含まれる情報では裏付けられない情報など
- Subtle Introduction of Baseless Information : 検証不可能な情報や主観的な意見など

表 3 に各タイプの hallucination チャンク数を示す。Conflict タイプよりも Baseless Information タイプの方が、Subtle タイプよりも Evident タイプの方が多くなっている。表 4 に各タイプ別の再現率 (Recall) を示す。提案手法は Conflict タイプよりも Baseless Information タイプの方が高い Recall を示した。これは、Conflict タイプは入力文章と一部の数値や単語などが異なっている場合が多いが、そういった数値の違いなどは埋め込みの類似度にあまり影響を与えないためだと考えられる。一方で、Baseless Information タイプは入力文章に記述されていない情報が含まれている。そのため、入力文章からマスクスパンを予測する際に、入力文章から予測できない場合が多く、提案手法が有効に働いたと考えられる。

B　Llama-3.1-8B-Instruct のプロンプト

表 5 に Llama-3.1-8B-Instruct を用いた hallucination スパン検出のプロンプトを示す。プロンプトは、RAGTruth の論文内で実験に用いられているものと同じものを用いた。