構造予測のための因果的注意機構を無効化したラベル付き教師あり LLM の微調整手法

矢野 憲[1] 高村 大也[1]
[1] 産業技術総合研究所 (AIST) 人工知能研究センター
{yano.ken,takamura.hiroya}@aist.go.jp

概要

デコーダ LLM は、要約、質問応答、論理推論など、幅広い自然言語タスクに適応できる汎用的な能力を有している。しかしながら、デコーダ LLM の汎用性は、学習時と推論時の両方で自己回帰的トークン生成が課す制約により、固有表現認識（NER）、関係抽出（RE）、イベント抽出などの構造予測タスクにおいて高い性能を出すことが難しい。本論文は、注意機構に焦点を当て、教師付きラベルを用いた識別的アプローチを採用することで、構造予測タスクにおける LLM の弱点を解決することを目的とする。具体的には、訓練時と推論時の両方で因果的注意マスクを除去することで、この弱点を克服できることを実証する。実験結果から、前述の手法と適切な LoRA（低ランク適応）設定で微調整したデコーダ LLM が、BC5CDR および CoNLL2003 NER タスクにおいて SOTA エンコーダモデルを上回る性能を発揮することが示された。

1 序論

NER は従来、BIO などのタグ付け方式を用いた系列ラベリング課題として扱われ、トークン分類を通じてエンティティのスパン検出と分類を行う。したがって、各トークンをその前後文脈を用いて正しく分類することが極めて重要である。BERT などのエンコーダ専用モデルは、マスクされたトークンをその前後文脈から推定するよう訓練されるため、学習されたトークン表現は NER などのトークン分類課題に極めて適している。一方、デコーダ LLM は、因果マスクを用いた自己回帰的なテキスト生成で訓練される。その結果、学習されたトークン表現には各トークンの右側文脈に関する情報が含まれない。しかしながら、膨大なテキストで訓練されたデコーダ LLM は、指示プロンプトを用いて様々な課題を解決でき、複数のタスクにわたって優れた能力を発揮する。しかし、エンティティスパンの正確な抽出とエンティティタイプの分類を必要とする NER などの構造予測タスクにおいて、デコーダ LLM を用いた生成による解決手法は十分な性能を示していない。

我々の研究課題への問いは以下の通りである：事前学習済みデコーダ LLM から因果的注意機構を除去しエンコーダモデルとして微調整した場合、NER タスクにおいて SOTA エンコーダ専用モデルと同等またはそれ以上の性能を達成できるか？それが可能な場合、デコーダ LLM のパラメータはどれくらい必要であるか？また、性能はモデルサイズの増加に伴い向上するか？

これらの問いに答えるため、因果的注意マスクを除去したデコーダ LLM の性能を、ベースラインとなるエンコーダモデルの性能と比較する実験を実施した。実験では、Gemma、Qwen2.5、Llama3.1、Llama3.2 の 4 つのモデルファミリーから、様々なパラメータサイズのデコードを 7 つ選択した。実験結果から、前述の手法と適切な LoRA 設定で微調整されたデコーダ LLM は、SOTA エンコーダモデルを上回る性能を発揮することが示された。

2 関連研究

固有表現認識（NER）は従来、系列ラベリング課題として扱われており、数多くの手法が提案されてきた [1, 2]。大規模なトランスフォーマーベースの事前学習済みモデルの登場により、性能向上のための標準手法として微調整が定着している [3, 4]。トランスフォーマーベースモデルを用いた主なアプローチは以下の通り分類できる：(1) BERT [4] やその派生モデル [5] などのエンコーダベースモデルを用いた教師あり微調整、 (2) T5 などのエンコーダ-デコーダモデルを用いた教師あり微調整 [6]、(3) LLM を用いたコンテキスト内学習 [7, 8]、 (4) LLM

を用いた教師あり微調整 [9, 10, 11]。デコーダ LLM による生成的な NER で、教師付きエンコーダベースラインを超える性能を示した研究も報告されている [12, 13, 14]。しかし、対象が医療など特定分野に限ったものや、結果の評価指標で、抽出したエンティティの位置を考慮していなかったり、エンティティの完全一致ではなく部分一致で評価しているケースも多い。例えば、GPT-NER [8] はエンコーダモデルと同等の性能を主張している。しかし、この手法には、一度に 1 つのエンティティタイプしか抽出できないことや、教師付きエンコーダモデルの性能に匹敵するには多数の文脈内サンプルが必要であることなど、いくつかの制限がある。NER などの構造化情報を LLM が生成するシリアル化されたテキスト形式に変換する際にも課題がある。制約のないテキスト生成では、出力のパースエラーが重大な懸念事項となる。これらの課題を克服するため、LLM から構造化した出力を生成する手法として、プロンプトエンジニアリング、制約付きデコーディング、文法ベース生成、ツール拡張生成などが存在する。LLM からの構造化出力は出力のパースエラーの軽減に寄与するが、推論時間に大きな影響を与える。

本論文は、Li や Dukić らの手法 [11, 10] と同様に、構造予測タスクを解決するため、LLM の因果的注意機構を完全注意に置き換えている方式を提案した。Dukić [10] らは、最高の性能を得るために、すべてのデコーダ層ではなく一部のデコーダ層を特定することで、因果的注意マスクを選択的に除去することを提案した。しかし、デコーダ層の数 n が大きい場合、この探索空間は膨大になる。探索空間を管理可能なサイズに縮小するため、Dukić [10] らはデコーダ層全体を 4 つの層グループ（各グループは連続する 8 層で構成、n=32 の場合）に分割することを提案した。これにより評価対象となる探索パターンは $2^4 = 16$ 通りに限定される。しかし、同数のデコーダ層を連続的にグループ化することが最良の NER 性能達成に最適かどうかは依然不明である。さらに、彼らのアブレーション実験では、異なるデコーダモデルや NER タスク全体で有意に優れた結果をもたらすデコーダ層の置換パターンが一貫して特定されなかった。したがって我々は、因果的注意マスクを全デコーダ層にわたり、完全注意マスクで置き換えることを選択した。

3 デコーダ LLM によるトークン分類

提案するトークン分類モデルは、事前学習済みデコーダ LLM で初期化し、全デコーダブロックの因果的注意マスクを完全注意マスクに置き換えることで定義される。デコーダ LLM を用いた NER 手法は、形式的には以下のように記述される。TClassifier を、事前学習済み LLM で初期化されたトークン分類モデルと定義し、因果言語モデルの出力に代えてトークン分類モデルの出力を用いる。入力トークンを $\{x_i | 1 \leq i \leq T\}$ とすると、トークン分類は以下のように定義される。

$$token\_clf = \text{TClassifier.init}(LLM) \quad (1)$$

$$\boldsymbol{l_t} = token\_clf(x_t | x_{1 \leq i \leq t}) \quad (2)$$

TClassifier は、出力部分を除いてデコーダ LLM の事前学習済みパラメータ全てで初期化される。$\boldsymbol{l_t}$ は固有表現ラベルに対するロジットである。次に、デコーダ LLM の全デコーダブロックにおいて因果的注意 (CA) を完全注意（FA）に置き換えたトークン分類モデルを FA-TClassifier として、以下のように定義する。

$$token\_clf = \text{FA-TClassifier.init}(LMM) \quad (3)$$

$$\boldsymbol{l'_t} = token\_clf(x_t | x_{1 \leq i \leq T}) \quad (4)$$

ここでは、固有表現ラベルに対するロジット $\boldsymbol{l'_t}$ は入力トークン全体が条件となる。学習には、正解トークンラベルと予測トークンラベルから計算した交差エントロピー損失を用いる。以降の議論では、表記上の煩雑さを避けるため、デコーダ LLM で初期化するトークン分類モデル FA-TClassifier と TClassifier をそれぞれ FA-LLM と CA-LLM と表記する。

4 実験手法

我々の実験では 2 つの NER データセットを使用した。一つは BC5CDR [15] であり、もう一つは CoNLL03 [16] である。BC5CDR には化学物質と疾患の 2 種類のエンティティが定義され、CoNLL2003 は人物、組織、場所、その他（miscellaneous）の 4 種類のエンティティが定義されている。各データセットの詳細は付録 A に記載されている。各モデルは対応する訓練データと検証データのみを用いて訓練し、対応するテストデータで性能を評価した。ベースラインとして、実験では以下のエンコーダモデルを使用した。BERT (bert-base-cased) [4]、RoBERTa (roberta-base) [5]、DeBERTa

(deberta-v3-large) [17]。デコーダ LLM については、以下の 4 つの LLM モデルファミリーを選択した。Gemma、Qwen2.5、Llama3.2、Llama3.1。また、各モデルファミリーから、異なるサイズのモデルを１つまたは２つ選択した。具体的には、以下の 7 つの様々なサイズのデコーダ LLM で評価した：Gemma-2B、Gemma-7B、Qwen2.5-0.5B、Qwen2.5-7B、Llama3.2-1B、Llama3.2-3B、Llama3.1-8B.

系列ラベリング学習の詳細

FA-LLM およびエンコーダモデルのベースラインは、最後の隠れ層の上にトークン分類器を追加することで系列ラベリング器として訓練される。トークン分類器はランダム値で初期化され、微調整中にベースである LLM のパラメータと共に訓練した。我々の実験では、ベースラインのエンコーダモデルはパラメータ効率化微調整（PEFT）技術を用いずに訓練した一方、デコーダ LLM は LoRA [18] を用いて訓練した。LoRA を用いて FA-LLM を微調整する理由は、FA-LLM は一般にベースラインのエンコーダモデルよりもはるかに大きな容量を持つため、訓練データへの過学習を緩和することにある。LoRA を用いることで、FA-LLM の学習可能パラメータを、エンコーダモデルの全パラメータサイズと比較してもはるかに小さい規模に削減可能である。さらに LoRA を使用することで、141GB のメモリを搭載した単一の H200 GPU 上で、デバイスバッチサイズを 8 にした場合に、全ての FA-LLM の微調整を実行することが可能であった。設定した学習パラメータと LoRA 構成の詳細は付録 B に記載されている。実験では、全ての FA-LLM に対して固定した LoRA 構成を用いて微調整した。ただし、異なる LoRA 構成が結果に与える影響を調査するため、FA-Llama3.2-3B および FA-Llama3.1-8B において異なる LoRA 構成を用いた追加実験を実施した。

5　結果

表 1 は、各種 FA-LLM とベースラインエンコーダモデルのテストデータに対するマイクロ F1 スコアを示す。系列ラベリング結果の評価には seqeval を用いた。「訓練可能パラメータ数」の列は訓練可能パラメータ数を示し、「パラメータ数」の列はモデルサイズを示す。LoRA を採用することで、FA-LLM の学習可能パラメータ数は、LoRA なしに訓練されたエンコーダモデルよりも大幅に低減されることが分かる。最高スコアは太字で、次点スコアは下線付きで表示している。10 億を超えるパラメータサイズを持つ全ての FA-LLM が、BC5CDR および CoNLL2003 データセットの両方でエンコーダモデルを上回る性能を発揮することを確認した。エンコーダモデルと同等のサイズである FA-Qwen2.5-0.5 のみが、それらに及ばない結果となった。これは、より大規模なモデルほど性能向上の可能性が高いことを示している。また、モデルサイズが類似している FA-LLM でも、異なるモデルファミリーに属する場合、性能が同等ではないことがわかる。例えば、FA-Llama3.2-3B の結果は、より大規模な FA-Gemma-7B や FA-Qwen2.5-7B よりもはるかに優れている。

次に、異なる LoRA 構成が性能に与える影響を調査した。ベースラインや他の FA-LLM の結果を上回る性能を示した FA-Llama3.2-3B と FA-Llama3.1-8B を使用した。表 2 は、これら 2 つのモデルに対する 4 種類の異なる LoRA 構成の結果を示しており、PEFT を使用せずにこれらの 2 つのモデルを訓練した場合の結果も併せて示している。表中の「r」「alpha」「dropout」「modules」の列は、それぞれ LoRA の構成項目であるランク、LoRA スケーリング、ドロップアウト確率、対象モジュールを示している。これら 2 つの FA-LLM は、表の 3 番目または 4 番目の構成で LoRA を用いて訓練した場合に最高の性能を示した。さらに、LoRA を使用しない学習では、両モデルとも性能が低下した。

これらの結果は、LoRA を用いた微調整が LoRA を用いない微調整と比較して NER 性能を大幅に改善することを示している。また、32 から 128 の比較的大きな $rank$ と $alpha = 2 \times rank$ を組み合わせ、全ての線形モジュールを対象とした LoRA 構成が、より優れた性能を達成するための最適な設定であることを示している。

6　エラー分析

表 6 は、DeBERTa、FA-Llama3.1-8B、および CA-Llama3.1-8B の BC5CDR と CoNLL2003 に対するエラー分析を示している。CA-Llama3.1-8B は因果的注意機構で訓練され、FA-Llama3.1-8B と CA-Llama3.1-8B 共に表 5 に示す LoRA 構成を用いて微調整された。NER エラーは 3 つのクラス（SPAN エラー、TYPE エラー、DETECTION エラー）に分類され、各エラークラスはさらに詳細なエラータイプに細分

表 1: ベースラインエンコーダモデルと完全注意機構によるデコーダ LLM のマイクロ F1 スコアの比較。「訓練可能パラメータ数」の列は、LoRA で微調整されたモデルの学習可能パラメータのサイズと、「パラメータ数」の列で指定された総パラメータサイズに対する割合を示す。各データセットにおける最高の結果は**太字**で、次点の結果は下線で示した。

Model	BC5CDR Test F1	CONLL2003 Test F1	PEFT	訓練可能パラメータ数	パラメータ数
BERT	0.858	0.908	-	107M	107M
RoBERTa	0.880	0.915	-	124M	124M
DeBERTa	0.882	0.911	-	405M	405M
FA-Gemma-2B	0.886	0.922	LoRA	39.2M (1.54%)	2,545M
FA-Gemma-7B	0.890	0.927	LoRA	100.0M (1.16%)	8,638M
FA-Qwen2.5-0.5B	0.863	0.891	LoRA	17.6M (3.44%)	511M
FA-Qwen2.5-7B	0.888	0.920	LoRA	80.8M (1.13%)	7,152M
FA-Llama3.2-1B	0.889	0.920	LoRA	22.6M (1.79%)	1,258M
FA-Llama3.2-3B	<u>0.899</u>	**0.935**	LoRA	48.6M (1.49%)	3,261M
FA-Llama3.1-8B	**0.904**	<u>0.929</u>	LoRA	83.9M (1.11%)	7,589M

表 2: 4 つの異なる LoRA 構成での FA-Llama3.2-3B および FA-Llama3.1-8B のマイクロ F1 スコア。「r」と「alpha」の列は LoRA パラメータのランクと LoRA スケーリング値を示す。「modules」の列は対象モジュールを示し、「all linear」は全線形モジュール（q_proj, k_proj, v_proj, o_proj, down_proj, gate_proj, up_proj）を意味する。

Model	BC5CDR Test F1	CoNLL2003 Test F1	r	alpha	dropout %	modules	訓練可能パラメータ数
FA-Llama3.2-3B	0.883	0.914	8	16	0.05	q,k,v,o	4.60M (0.143%)
	0.889	0.918	16	32	0.05	q,k,v,o	9.19M (0.285%)
	0.898	0.928	64	128	0.05	all linear	97.30M (2.940%)
	0.900	0.927	128	256	0	all linear	195.00M (5.710%)
	0.864	0.913	-	-	-	-	3.26B (100.0%)
FA-Llama3.1-8B	0.888	0.919	8	16	0.05	q,k,v,o	6.84M (0.091%)
	0.893	0.922	16	32	0.05	q,k,v,o	13.70M (0.182%)
	0.904	0.929	64	128	0.05	all linear	168.00M (2.190%)
	0.900	0.932	128	256	0	all linear	336.00M (4.280%)
	0.878	0.924	-	-	-	-	7.58B (100.0%)

化して示した。両方の分析において、各エラータイプの発生件数が DeBERTa よりも FA-Llama3.1-8B で少ないことを確認した。これは、FA-Llama3.1-8B が DeBERTa よりも堅牢な固有表現認識知識を獲得したことを示している。CA-Llama3.1-8B は両方の分析で性能が低く、因果的注意機構が NER のボトルネックであることを裏付けている。FA-Llama3.1-8B と DeBERTa の分析において、BC5CDR では CoNLL2003 に比べて TYPE の誤り数が大幅に少なく、CoNLL2003 ではエンティティタイプ間の曖昧性がより大きいことを示している。上記の結果と分析は、デコーダ LLM が NER のような構造化予測タスクで示す弱点を、我々が提案する手法によって改善可能であることを裏付けている。

7 結論

デコーダ LLM が構造予測タスクで示す弱点は、訓練時と推論時の両方で因果的注意マスクを除去することで克服できることを示した。実験結果から、提案した手法で微調整したデコーダは、SOTA エンコーダモデルよりも優れた性能を達成することが明らかになった。実際、LoRA で微調整した 10 億パラメータ規模を超える全ての FA-LLM が、BC5CDR および CoNLL2003 のデータセットにおいてエンコーダモデルの結果を上回ることを確認した。LoRA 設定が性能に大きく影響するため、最良の性能を達成するには最適な設定の探索が不可欠である。今後の課題として、このような最適な LoRA 設定を自動取得する手法の開発が含まれる。

謝辞

この成果は，産総研政策予算プロジェクト「フィジカル領域の生成 AI 基盤モデルに関する研究開発」の結果得られたものである．

参考文献

[1] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[2] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In Katrin Erk and Noah A. Smith, editors, **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

[3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.

[6] Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. Instructuie: Multi-task instruction tuning for unified information extraction, 2023.

[7] Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. Few-NERD: A few-shot named entity recognition dataset. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 3198–3213, Online, August 2021. Association for Computational Linguistics.

[8] Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. GPT-NER: Named entity recognition via large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, **Findings of the Association for Computational Linguistics: NAACL 2025**, pp. 4257–4275, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[9] Weilu Xu, Renfei Dang, and Shujian Huang. LLM's weakness in NER doesn't stop it from enhancing a stronger SLM. In Adam Anderson, Shai Gordin, Bin Li, Yudong Liu, Marco C. Passarotti, and Rachele Sprugnoli, editors, **Proceedings of the Second Workshop on Ancient Language Processing**, pp. 170–175, The Albuquerque Convention Center, Laguna, May 2025. Association for Computational Linguistics.

[10] David Dukić and Jan Šnajder. Looking right is sometimes right: Investigating the capabilities of decoder-only llms for sequence labeling, 2024.

[11] Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu lee Wang, Qing Li, and Xiaoqin Zhong. Label supervised llama finetuning, 2023.

[12] Vipina Kuttichi Keloth, Yan Hu, Qianqian Xie, Xueqing Peng, Yan Wang, Andrew Zheng, Melih Selek, Kalpana Raja, Chih-Hsuan Wei, Qiao Jin, Zhiyong Lu, Qingyu Chen, and Hua Xu. Advancing entity recognition in biomedicine via instruction tuning of large language models. **Bioinformatics**, Vol. 40, , 2024.

[13] Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Bowen Yan, and Min Zhang. Rethinking negative instances for generative named entity recognition. **ArXiv**, Vol. abs/2402.16602, , 2024.

[14] Mingchen Li, Huixue Zhou, Han Yang, and Rui Zhang. Rt: a retrieving and chain-of-thought framework for few-shot medical named entity recognition. **Journal of the American Medical Informatics Association : JAMIA**, Vol. 31, pp. 1929 – 1938, 2024.

[15] Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. **Database: The Journal of Biological Databases and Curation**, Vol. 2016, , 2016.

[16] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In **Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003**, pp. 142–147, 2003.

[17] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

[18] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

A NER データセットの詳細

表 3: BC5CDR [15] および CoNLL03 [16] のサンプル数の統計

Dataset	Train	Val	Test	Named entity types
BC5CDR	5,228	5,330	5,865	Chemical, Disease
CoNLL03	14,041	3,250	3,453	PER,ORG,LOC,MISC

B 学習パラメータ

表 4: 学習パラメータ

	Encoders	FA-LLMs
per_device_batch_size	8	8
gradient_accumulation_steps	1	1
num_train_epochs	10	10
learning_rate	5E-05	5E-05
weight_decay	0.001	0.001
warmup_ratio	0	0
lr_scheduler_type	cosine	cosine
gradient_checkpointing	False	False
model_max_length	512	512
bf16	false	false
use_lora	No	Yes

表 5: LoRA 設定

LoRA configuration item	Value
lora_r	32
lora_alpha	64
lora_dropout	0.05
lora_bias	none
lora_taget_modules	q_proj,k_proj,v_proj,o_proj down_proj,gate_proj,up_proj

C エラー分析

エラーは、正解ラベルと予測ラベルを比較して計算した。各ラベルは (開始位置, 終了位置, エンティティタイプ) の形式でエンティティラベルのリストとして表され、開始位置と終了位置はエンティティスパンの開始点と終点、エンティティタイプはエンティティの種類を示す。SPAN エラーにおいて、開始または終了の位置が一致するがスパン長が一致しない場合に、too_short または too_long に分類される。一方、開始と終了の両位置が一致せず、かつスパン長の差が 2 単語以内の場合、boundary_off に分類される。それ以外の場合は completely_wrong に分類した。

表 6: DeBERTa、FA-Llama3.1-8B、および CA-Llama3.1-8B における BC5CDR と CoNLL2003 の誤り分析

	BC5CDR			**CoNLL2003**		
	DeBERTa	FA-Llama3.1-8B	CA-Llama3.1-8B	DeBERTa	FA-Llama3.1-8B	CA-Llama3.1-8B
Total Prediction	10502	10442	11383	5800	5775	5912
Exact Matches	8705	9007	7247	5177	5283	4219
Total Errors	1797	1435	4136	623	492	1693
SPAN ERRORS						
boudary_off	5 (0.3%)	4 (0.3%)	1 (0.0%)	0 (0.0%)	1 (0.2%)	0 (0.0%)
too_short	198 (11.0%)	195 (13.6%)	284 (6.9%)	52 (8.3%)	47 (9.6%)	49 (2.9%)
too_long	244 (13.6%)	169 (11.8%)	138 (3.3%)	75 (12.0%)	44 (8.9%)	29 (1.7%)
completely_wrong	7 (0.4%)	6 (0.4%)	10 (0.2%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
TYPE ERRORS						
correct_span_wrong_type	39 (2.2%)	14 (1.0%)	195 (4.7%)	236 (37.9%)	176 (35.8%)	734 (43.4%)
DETECTION ERRORS						
false_positives	693 (38.6%)	633 (44.1%)	1574 (38.1%)	152 (24.4%)	127 (25.8%)	264 (15.6%)
false_negatives	611 (34.0%)	414 (28.9%)	1934 (46.8%)	108 (17.3%)	97 (19.7%)	617 (36.4%)