日本語エンティティ曖昧性解消の体系的評価

澤田 悠治 [1]　安井 雄一郎 [2]　渡辺太郎 [1]
[1] 奈良先端科学技術大学院大学 [2] 日本経済新聞社
{sawada.yuya.sr7,taro}@is.naist.jp
yuichiro.yasui@nex.com

概要

エンティティリンキングは，テキスト上のエンティティを知識ベースのエントリとリンキングすることで，幅広い応用で活用される基盤技術である．しかしながら，最新モデルを用いたエンティティリンキング性能やその性質は十分に評価されているとは言えず，特に日本語の専門ドメインにおける最新モデルの汎用性は明らかではない．本研究では，ニュース，行政，地理の 3 ドメインからなる日本語評価ベンチマークを構築し，Text Embedding モデルや Dual Encoder，LLM リランキング等の体系的比較を行った．実験の結果，Text Embedding モデルが全ドメインで約 80%の R@10 を達成し，リランキングにより最大 5 ポイントの精度向上を確認した．

1　はじめに

エンティティリンキング（Entity Linking; EL）は，テキスト中の特定のエンティティを指し示す表現（メンション）を，Wikidata [1] などの知識ベース（KB）上の適切なエントリに紐付けるタスクである．EL システムは，表記揺れや文脈に依存する曖昧性のある表記を正規化するのみならず，近年では質問応答や LLM の知識編集などのアプリケーションにおいて，外部知識を検索・参照するための基盤技術として重要な役割を担っている．

一般的な EL システムは，テキストからメンションを抽出する「メンション抽出」と，抽出したメンションを KB 上の正しいエントリに特定する「エンティティ曖昧性解消（Entity Disambiguation; ED）」の 2 段階のパイプラインで構成されている．メンション抽出は，spaCy[1)] や GiNZA[2)] などの既存 NLP ツールの活用することで任意の固有表現タイプの抽出が可能であるため，EL 研究の多くは，文脈に基づいて

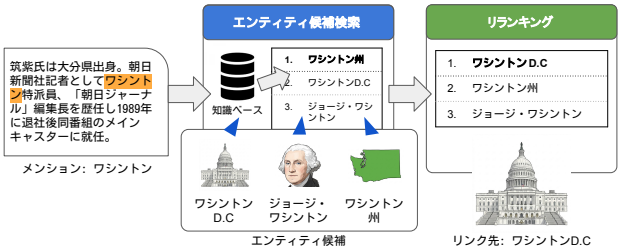

図 1　エンティティリンキングの概要図

正しいリンク先を特定する ED タスクに焦点を当てている．ED タスクでは，図 1 に示すように，メンションのリンク先になりえるエントリを抽出するエンティティ候補検索，およびエンティティ候補の中から適当なリンク先を特定するリランキングという 2 つのサブタスクで構成されている．エンティティ候補検索では，BERT などの事前学習済みモデルをベースとし Dense Passage Retriever [2] が，リランキングでは LLM を用いた全候補を考慮した生成型手法 [3, 4] が提案されており，ZELDA などのエンティティ曖昧性解消のベンチマーク [5] において 80%程度の性能を実現している．

一方で，EL システムやベンチマークの多くは英語テキストを対象に構築されており，日本語の専門ドメインにおける最新モデルの汎用性は明らかではない．日本語テキストについては，Mewsli-9 などの多言語 EL タスク [6, 7] において検証が行われている一方で，より専門的な語彙を含む文書や，時間軸によって異なるエンティティに対し，どのモデルが最良であるかは明らかではなく，体系的な検証が求められている．

そこで本研究では，既存の日本語データセットを基盤とした多ドメインの評価ベンチマークを構築し，従来モデルの体系的な評価を行う．具体的には，ニュース，行政，地理とった異なる 3 つのドメインと投稿時期の異なるデータセットを選定し，評価用ベンチマークを整備した．本ベンチマークを用いた実験では，E5 などの Text Embedding モデ

1）https://spacy.io/models/ja
2）https://github.com/megagonlabs/ginza

ル，BERT をベースにした Dual Encoder モデル，および LLM ベースのリランキングモデルを採用し，各データセットを用いて学習・評価を行うことで，その性能を包括的に比較する．全データセットにおいて Text Embedding モデルが 80%程度の Recall@10 という高い検索精度を示し，さらにリランキングモデルを組み合わせることで，精度が最大で 5 ポイント改善することを確認した．

2 システム概要

本研究で対象とするエンティティ曖昧性解消（ED）タスクは，一般的にエンティティ候補検索とリランキングの 2 段階のパイプラインで構成される．第 1 段階の候補検索モジュールでは，膨大な知識ベースの中からメンションのリンク先候補となるエントリを絞り込む．第 2 段階のリランキングモジュールでは，絞り込まれた限定的な候補集合に対して，より複雑な文脈理解が可能なモデルを適用することで，最終的な正解エントリを特定する．

2.1 エンティティ候補検索

エンティティ候補検索では，与えられたメンションとその周辺文脈に基づき，KB に含まれる全エントリの中から，正解の可能性がある候補集合を抽出する．具体的には，テキスト中のメンション m とその周辺文脈 c に対して，特定のエントリ e がリンク先となる確率 $P(e \mid m, c)$ を計算し，知識ベースの全エントリ E の中から確率が上位の k 個のエントリを候補集合 $E_c = \{e_1, e_2, ..., e_k\}$ として抽出する．

従来のエンティティ候補検索では，Wikipedia や YAGO におけるメンションとエントリ間のリンク頻度に基づく事前確率 $P(e \mid m)$ [8] やページランク [9] が主に利用されてきた．近年では，Text Embedding モデル [10] や Dual Encoder モデル [2] を用いた Dense Passage Retriever (DPR) [11] が導入され，メンションの周辺文脈 c を含めた情報を一括して符号化することで，メンションの周辺文脈とエントリの概要文との意味的な類似度の計算により，文脈に即した適切な候補集合 E_c を特定する．

2.2 リランキング

リランキングでは，前段のエンティティ候補検索によって絞り込まれた候補集合 E_c の中から，文脈に最も合致するエントリ e' を特定する．本稿では，この確率 $P(e \mid m, c, E_c)$ の算出アプローチとして「識別型」と「生成型」の 2 種類を検討する．

$$e' = \arg\max_{e \in E_c} P(e \mid m, c, E_c) \tag{1}$$

識別型リランキングでは，候補集合 E_c 内の各エントリ e に対して文脈との適合度を示すスコア s_e を付与し，その中から最も確率の高いエントリを選択する．本手法は，候補集合をクラスとしたマルチクラス分類問題として，以下の Softmax 関数を用いて定式化される．

$$P(e \mid m, c, E_c) = \frac{\exp(s_e)}{\sum_{e' \in E_c} \exp(s_{e'})} \tag{2}$$

Cross-Encoder [2] では，メンション m，周辺文脈 c，および各エントリの概要文 d_e を一括してモデルへ入力し，その出力ベクトルを 1 次元の線形層を通じてスコアリングする（$s_e = \text{Linear}(\text{Encoder}(m, c, d_e))$）．Fevry ら [12] は，メンション m，周辺文脈 c を符号化したベクトルと，分類層の重みとして保持される各エントリの埋め込みベクトルとの内積をスコア s_e として用いる．

生成型リランキングでは，生成型言語モデルを用い，正解エントリを識別するトークン系列 $y = \{y_1, y_2, \ldots, y_T\}$（エンティティ名 [3] や，多肢選択形式における選択肢番号 [4] など）を直接的に生成する．ここでは，候補集合 E_c 全体の情報を符号化関数 $\text{Enc}(\cdot)$ を介して入力として与えることで，全候補の差異を考慮した推論を行う．

$$P(y \mid m, c, E_c) = \prod_{t=1}^{T} P(y_t \mid y_{<t}, \text{Enc}(m, c, E_c)) \tag{3}$$

3 評価実験

本稿では，既存の日本語 EL データセットを用いて，候補検索およびリランキングモジュールの各モデルの性能を比較する．また，エンティティの種類によるリンキングの難易度を調査するため，固有表現タイプおよび訓練データでの出現の有無による性能差を分析する．

3.1 データセット

本稿で採用する日本語 EL データセットの一覧を表 1 に示す．本研究では，ドメイン汎用性に加え，エンティティの時系列的変化に対する頑健性を検証するため，性質の異なる複数のデータセットを選定

表 1 データセット一覧．*は自動生成により作成したデータセットを表す．

	ドメイン	文書数	文字数	メンション w/ QID	メンション NIL	エンティティ
Jawikify [13]	新聞記事	340	1703.7	18,740	19,349	5,779
CADEL [14]	行政文書	3,852	54.7	6,462	1,620	1,830
ATD-MCL [15]	ブログ	100	1415.5	2,224	0	1,190
Mewsli-9 [16]	WikiNews*	3410	598.6	34,463	0	13,663
Nikkei [17]	新聞記事	3024	387.3	35,483	4,641	7,085

した．例えば，Jawikify と Nikkei はどちらも新聞記事をソースとしているが，Jawikify が 2001–2005 年に発行された記事を用いているのに対し，Nikkei は 2022 年時の記事を対象としており，メンションが指し示すエントリは発行時期によって異なる可能性がある．

これらのデータセットの利用にあたっては，分割済みのデータが公開されているものはそのまま採用し，未分割のデータセットについては，7:1:2 の比率で訓練用，開発用，評価用に分割して使用した．リンク先となる知識ベース（KB）については，地理ドメインを扱う ATD-MCL を除く全てのデータセットにおいて，日本語 Wikidata に統一した．Jawikify については，元のデータに含まれる Wikipedia のページ ID から MediaWiki API[3] を用いて Wikidata ID を取得し，対応する Wikidata ID が存在しないページ ID は NIL に変換した．ATD-MCL については，中谷ら [15] による OpenStreetMap[4] のエントリ辞書を KB として使用する．

3.2 実験設定

本稿では，候補検索，リランキングの各段階におけるモデルの性能を比較する．評価に用いる各モデルは entity-linkings[5] をもとに作成した．

候補検索 エンティティ候補検索では，以下の 3 つのアプローチを比較検証の対象とする．Text Embedding モデルには，多言語対応モデルである Multilinglal E5 [18] と，日本語特化モデルである Ruri-v3[19] を採用する．Dual Encoder においては，東北大の日本語 BERT モデルに加え，Ruri-v3 の基盤モデルである ModernBERT モデル [20] を比較対象として選定し，検索精度の差異を比較する．これらのモデルの学習にあたっては，Wu ら [2] と同様に，まずミニバッチ内のサンプルのみを用いて学習し，その後ミニバッチに Hard Negative を加えて再学習する 2 段階のプロセスを採用した．また，ベースラインとして，文字列類似度に基づく BM25 を用いる．加えて，日本語 Wikipedia のダンプデータから抽出したメンションのリンク先統計を抽出し，これに基づく事前確率 $P(e|m)$ を比較対象とする．評価指標には，検索上位 k 件の中に正解リンクが含まれている割合を示す Recall@k を採用する．

リランキング リランキングでは，前段の候補検索で作成した ruri-v3-130m[6] を用い，抽出された上位 30 件のエントリを候補集合 E_c として使用する．ここでは，識別型と生成型のそれぞれにおいて以下の 2 つのモデルを比較対象とする．識別型リランキングでは，Cross-Encoder [2] および Fevry ら [12] におけるスパン分類手法を採用し，基盤モデルにはいずれも ModernBERT-130m [7] を使用する．生成型リランキングにおいては，LLM を用いた Zero-shot 手法である ChatEL [4]（GPT-4o-mini[8]）と比較する．また，これらの手法に対するベースラインとして，ruri-v3-130m から再度 Hard Negative をサンプリングして学習したモデルを追加する．評価指標には Top-1 Accuracy を採用し，モデルが最も高いスコアを付与したエントリと正解エントリの整合性を比較する．

3.3 実験結果

候補検索 候補検索の実験結果を表 2 に示す．DPR は全てのデータセットにおいて 90 ポイント程度の R@50 を示しており，候補検索モジュールによって正解のリンク先 ID が上位 50 件に絞りこめることを示している．Mewsli-9 においては，Prior と同程度の再現率を示しているものの，その他のデータセットでは Prior を上回っており，モデル学習によってエンティティとの意味的類似性を捉えていることを表している．特に，Text Embedding モデルは全てのデータセットにおいて 90 ポイント程度の R@50 を示したことから，候補検索モジュールは，全データセットにおいてエンティティ候補を上位 50 件まで絞りこめることを示唆している．

次に，日本語 Wikipedia dump データ[9] を使ったタスク事前学習による ruri-v3-30m の性能差を表 3 に

3） https://www.mediawiki.org/wiki/API
4） https://www.openstreetmap.org/
5） https://github.com/naist-nlp/entity-linkings
6） https://huggingface.co/cl-nagoya/ruri-v3-130m
7） https://huggingface.co/sbintuitions/modernbert-ja-130m
8） https://platform.openai.com/docs/models/gpt-4.1-mini
9） 本稿では，中谷ら [21] が収集した 2024 年 9 月 1 日の日本語 Wikipedia の記事の冒頭を使用する．

表 2　候補検索モデルの性能

	CADEL			Jawikify			ATD-MCL			Mewsli-9			Nikkei		
	R@1	R@10	R@50	R@1	R@10	R@50	R@1	R@10	R@50	R@1	R@10	R@50	R@1	R@10	R@50
BM25	16.55	29.57	40.60	7.99	17.13	24.49	18.60	42.89	55.58	20.08	43.10	60.11	16.03	33.33	44.92
Prior	41.65	51.61	53.52	35.48	46.25	47.11	–	–	–	69.71	80.01	80.24	51.05	69.21	69.29
Text Embedding															
mE5-small-130m	59.01	79.06	85.39	57.73	75.80	83.68	**45.73**	69.80	76.81	68.63	82.28	86.02	78.24	89.46	93.64
mE5-base-355m	62.38	83.27	89.41	25.24	44.44	60.30	38.51	65.43	74.84	60.74	77.45	82.64	74.29	89.29	94.22
ruri–v3-30m	46.75	72.18	82.30	70.31	90.40	94.70	26.70	59.96	77.46	59.96	78.81	84.23	84.24	95.24	97.77
ruri-v3-130m	62.38	**83.27**	**89.41**	**77.25**	**92.35**	**95.60**	**45.73**	**78.56**	**87.09**	**69.92**	**84.84**	**88.55**	**87.18**	**97.43**	**98.80**
Dual Encoder															
bert-base-110m	57.28	77.93	84.86	42.20	66.05	79.18	8.75	25.38	39.17	63.95	78.83	82.43	79.45	91.17	94.74
bert-large-340m	62.90	81.36	87.37	50.86	76.24	87.15	10.72	31.95	43.98	66.58	81.55	85.66	80.32	91.75	95.28
modernbert-30m	37.10	58.56	69.28	67.63	84.13	89.84	17.94	44.20	59.08	59.96	78.81	84.23	76.66	90.76	95.11
modernbert-130m	40.84	61.33	71.37	75.84	91.23	94.98	2.63	14.22	29.32	67.02	83.54	87.36	81.71	95.22	97.65

表 3　Wikipedia 事前学習による ruri-v3-30m の R@10

	CADEL	Jawikify	ATD-MCL	Mewsli-9	Nikkei
Train data	72.18	90.40	59.96	78.81	95.24
Jawiki	65.15	43.02	67.18	84.29	81.27
Jawiki → Train data	82.88	88.95	68.71	**86.02**	90.27

表 4　各データセットの候補リランキング精度

	CADEL	ATD-MCL	Mewsli-9
ruri-v3-130m	64.73	51.42	75.29
Fevry	31.22	12.69	–
Cross Encoder	67.06	44.42	–
ChatEL	**67.56**	56.02	68.40

示す．ATD-MCL，CADEL，Mewsli-9 は，Wikipedia 記事の追加学習によって 8 ポイント以上の再現率の改善が見られる一方，Jawikify と Nikkei は 2 ポイント程度低下する傾向が見られた．これは，Jawikify と Nikkei は，訓練用データと評価用データのメンション被覆率は Wikipedia 記事との被覆率を上回っており，頻出エンティティが多く出現する新聞ドメインにおいては，Wikipedia の事前学習が不要である可能性がある．

候補リランキング　リランキングの実験結果を表 4 に示す．リランキングによって候補検索モジュールから 3 ポイント程度の精度が改善した一方で，一部のモデルおよびデータセットで性能の低下が見られた．実際に Fevry ら [12] や Wu ら [2] はモデル学習に Wikipedia の記事データを用いていることから，リランキングモデルの構築にはデータセットの訓練データではリソースが不十分であった可能性がある．

表 5　ruri-v3-130m による固有表現タイプ別の R@1．括弧内の数字は評価データに出現するメンション頻度を示す．

	Known		Unknown	
	QID	NIL	QID	NIL
人名 (840)	80.0	97.1	53.9	61.2
組織名 (655)	85.7	40.0	56.2	63.0
地名 (991)	93.0	64.7	77.0	67.9
施設名 (330)	100.0	77.8	60.0	68.1
製品名 (1,799)	90.5	75.9	55.4	60.5
イベント名 (176)	54.2	16.7	50.9	61.3
自然物名 (193)	84.7	100.0	65.8	48.0

3.4　各固有表現タイプに対する性能差

Jawikify の 7 種類の固有表現タイプ別の性能を表 5 にの R @ 1 を示す．すべての固有表現タイプにおいて，訓練データに出現し，かつ Wikidata ID をもつ事例は 80 ポイント程度の再現率を示す一方，それ以外の事例においてはこれらの再現率を大きく下回った．特に，イベント名は訓練データで出現した事例においても 50 ポイント程度の再現率を示している．これは，“東京オリンピック” や “ワールドカップ” のような，テキストの時間軸によってリンク先が異なるメンションが性能低下を招いている可能性が考えられる．以上の結果から，既存の候補検索およびリランキングモジュールは，日本語においても有用な性能を示したものの，未知のエンティティ，NIL および時間変化に関わるメンションに対して依然として課題が見られる．これらの特徴をもつメンションに対するリンキング性能の改善と，時間変化に伴う EL システムの性能維持は今後の課題である．

参考文献

[1] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. **Commun. ACM**, Vol. 57, No. 10, p. 78–85, September 2014.

[2] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6397–6407, Online, November 2020. Association for Computational Linguistics.

[3] Junxiong Wang, Ali Mousavi, Omar Attia, Ronak Pradeep, Saloni Potdar, Alexander Rush, Umar Farooq Minhas, and Yunyao Li. Entity disambiguation via fusion entity decoding. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 6524–6536, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[4] Yifan Ding, Qingkai Zeng, and Tim Weninger. ChatEL: Entity linking with chatbots. 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. 3086–3097, Torino, Italia, May 2024. ELRA and ICCL.

[5] Marcel Milich and Alan Akbik. ZELDA: A comprehensive benchmark for supervised entity disambiguation. In Andreas Vlachos and Isabelle Augenstein, editors, **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 2061–2072, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[6] David Kubeša and Milan Straka. Damuel: A large multilingual dataset for entity linking, 2023.

[7] Jan A. Botha, Zifei Shan, and Daniel Gillick. Entity Linking in 100 Languages. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 7833–7845, Online, November 2020. Association for Computational Linguistics.

[8] Onoe Yasumasa and Durrett Greg. Fine-grained entity typing for domain independent entity linking. In **Proceedings of the AAAI Conference on Artificial Intelligence, 34 Volume**, June 2020.

[9] Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In Rada Mihalcea, Joyce Chai, and Anoop Sarkar, editors, **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 238–243, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[10] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. **arXiv preprint arXiv:2212.03533**, 2022.

[11] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics.

[12] Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. Empirical evaluation of pretraining strategies for supervised entity linking, 2020.

[13] Davaajav Jargalsaikhan, 岡崎直観, 松田耕史, 乾健太郎. 日本語 wikification コーパスの構築に向けて. 言語処理学会第 22 回年次大会 (NLP2016), March 2016.

[14] 翔平東山, 将夫出内, 将夫内山. 日本語エンティティリンキングのための行政機関ウェブ文書コーパスの構築. 情報処理学会研究報告, Vol. 2024-NL-260, No. 10, pp. 1–15, jun 2024.

[15] Hibiki Nakatani, Hiroki Teranishi, Shohei Higashiyama, Yuya Sawada, Hiroki Ouchi, and Taro Watanabe. A text embedding model with contrastive example mining for point-of-interest geocoding. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 7279–7291, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[16] Jan A. Botha, Zifei Shan, and Daniel Gillick. Entity Linking in 100 Languages. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 7833–7845, Online, November 2020. Association for Computational Linguistics.

[17] 澤田悠冶, 安井雄一郎, 大内啓樹, 渡辺太郎, 石井昌之, 石原祥太郎, 山田剛, 進藤裕之. 企業名の類似度に基づく日経企業 id リンキングシステムの構築と分析. 自然言語処理, Vol. 31, No. 3, pp. 1330–1355, 2024.

[18] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. **arXiv preprint arXiv:2402.05672**, 2024.

[19] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings, 2024.

[20] Hayato Tsukagoshi, Shengzhe Li, Akihiko Fukuchi, and Tomohide Shibata. ModernBERT-Ja. https://huggingface.co/collections/sbintuitions/modernbert-ja-67b68fe891132877cf67aa0a, 2025.

[21] 中谷響, 安井雄一郎, 若本亮佑, 石井昌之, 大内啓樹, 渡辺太郎. Wikidata に基づく大規模ジオコーディングデータセット. 言語処理学会第 31 回年次大会 (NLP2025), March 2025.

表 6 候補検索およびリランキングモジュール学習時のハイパーパラメータ

	候補検索	リランキング
batch size (train)	64 (16)	2
batch size (eval)	128 (32)	4
learning rate	1e−5	1e−5
epochs	10	10
optimizer	AdamW	AdamW
eps	1e−6	1e−6
scheduler	linear	linear
warmup ratio	0.1	0.1
weight decay	0.01	0.01
max grad norm	0.00	0.00
β	[0.9, 0.98]	[0.9, 0.98]

A 実験設定の詳細

候補検索とリランキングの学習時に使用したハイパーパラメータを表 6 に示す．候補検索とリランキングモジュールはいずれも最大トークンサイズを 128 に固定し，メンションの前後 250 文字の周辺テキストをモデル入力とする．候補検索においては，エンティティの説明文の入力も同様に 128 トークンとする．