定義文自動生成による
専門分野向けエンティティリンキングの精度向上

石垣達也 [1]　上原由衣 [1]　劉珊珊 [1]　松本裕治 [2]　高村大也 [1]
[1] 産業技術総合研究所　[2] 理化学研究所
{ishigaki.tatsuya, yui.uehara, shanshan.liu, takamura.hiroya}@aist.go.jp
yuji.matsumoto@riken.jp

概要

本研究では，専門分野向けのエンティティリンキング (EL) を複数の知識ベースから獲得した定義文を用いて学習する枠組みを扱う．EL は入力文中の言及 (メンション) を知識ベース中のエンティティに紐付ける問題である．候補エンティティの定義文を埋め込み表現で表し，リンクするか否かを判定する分類器の学習に用いる手法が知られている．しかし，定義文は知識ベース中のすべてのエンティティに付与されているとは限らず，欠損が生じていることがしばしばある．そこで，欠損した定義文を言語生成技術により補完することを提案する．より具体的には，複数の知識ベースを利用し，片方の知識ベースの定義文からもう片方の知識ベースの定義文を生成することにより，欠損した定義文を互いに補い合う．また，元々の定義文および生成した疑似定義文を有効活用するためのエンティティリンキングモデルのアーキテクチャについて調査を行った．MedMentions を用いた実験において，疑似定義文も活用するモデルが既存手法よりも有意に上回る性能を示した．

1　はじめに

エンティティリンキング (EL) は入力文に含まれる固有名詞などの言及 (メンション) を，知識ベース中のエンティティに対応付ける問題である．Wikipedia などの一般分野の知識ベースにリンクする設定 [1, 2] のほか，分野特化の設定 [3] が存在する．本研究では特に生物医学の UMLS にリンクする後者の設定を扱う．どちらの設定でも，標準的なアプローチでは，1) 入力文の言及を検出し，2) 知識ベースから候補エンティティを抽出した上で，3) 候補をスコアリングする．本研究では特にスコアリング部分に着目し，言及と候補エンティティを適切に埋め込み表現に変換し，それらがリンクするか否か正しく判定する分類器の構築を目指す．

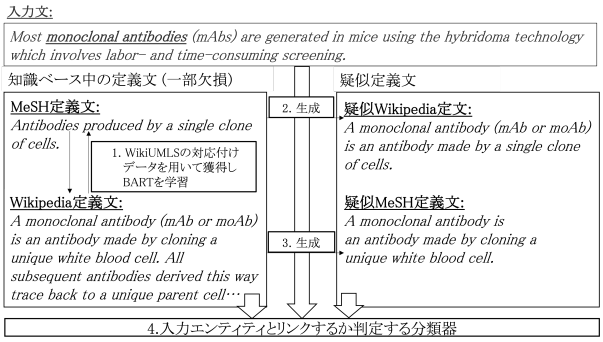

図 1: 疑似定義文を用い EL モデルを学習する枠組み．

従来，EL モデルでは言及と候補エンティティそれぞれを埋め込みベクトルとして表現し，類似度や教師あり学習を用いてリンクするか否か判定する手法がよく用いられている．言及は主に BERT [4] 等でベクトル化するのが標準的であるが，候補エンティティのベクトル化にはエンティティ名を用いる手法やグラフ構造を用いる手法など様々な方法がある．また，知識ベースの各エンティティの定義文 [1)] を用いる手法もあり，その有効性が知られている [2]．しかし，定義文は必ずしもすべてのエンティティに対し用意されているわけではない．そこで，本研究では言語生成技術により定義文を生成し EL モデルの学習に用いる枠組みを提案する．とくに複数の知識ベースから定義文を取得し，欠損する定義文を言語生成により互いに補完する．

図 1 に示すように，提案する枠組みは，まず UMLS と MeSH，UMLS と Wikipedia の対応付けデータ [5] から，UMLS エンティティに対し MeSH および Wikipedia に含まれる定義文を取得する．ただし，

1) 一文からなることが多い短いテキストなので，本論文では定義文と呼ぶが，実際は複数の文から成ることもある．

これらの定義文は欠損している可能性がある．そこで，MeSH および Wikipedia 両方に定義文が存在する事例を用いて，2 つの言語生成モデル [6] を学習する．1 つ目は MeSH の定義文を入力し Wikipedia の定義文らしい出力を自動生成するモデル，2 つ目は Wikipedia の定義文から MeSH の定義文らしい出力を生成するモデルである．これら 2 つの言語生成モデルを用い，知識ベースに存在する MeSH および Wikipedia の定義文から疑似的な MeSH および Wikipedia の定義文を獲得する．これにより，MeSH もしくは Wikipedia いずれかに定義文が獲得できれば欠損テキストに対し疑似的な定義文を獲得でき，定義文の欠損の問題が軽減できると考えられる．最終的に 1 つの候補エンティティに対し，MeSH および Wikipedia の定義文とそれらから自動生成した疑似的な定義文という 4 種類のテキストを得る．

複数の種類の定義文を利用するエンコードモデルとして 2 種類の方法を提案する．具体的には 1) 欠損箇所を疑似定義文で補完する手法，2) 疑似定義文を疑似定義文用のエンコーダで読み込む手法を比較する．MedMentions [7] を用いた実験より，2) の学習手法，すなわち MeSH 定義文，Wikipedia 定義文，またそれぞれに対し自動生成した疑似定義文の 4 種類を異なるエンコーダで読み込む手法が良い性能を示すことが分かった．疑似定義文も用いて学習したモデルは既存手法 [3] よりも良い性能を示すことを報告する．この結果は，従来人間が読むテキストを生成することを目的としていた言語生成モデルの新たな応用の可能性を示すものである．

2 関連研究

EL の既存設定は Wikipedia や Freebase [8] など一般分野の知識ベースを対象とする設定 [1, 9, 2] や，UMLS [10] などの専門分野の知識ベースをリンク先とする設定が存在する．これらは，テキストデータの利用という観点では定義文を用いる設定 [1, 2] と定義文を用いずエンティティ名，エイリアス名，グラフ構造 [9] といった定義文以外を用いる設定 [9] に分けられる．定義文を用いる設定はただ 1 つの定義文の存在を仮定しており，1 つの候補エンティティに対し複数種類の定義文を用いる本研究の設定と異なる．

EL の既存手法には入力文のエンティティ抽出と知識ベースエンティティへのリンキングを同時に行う手法 [11]，それぞれ別に行う手法 [3] が存在する．

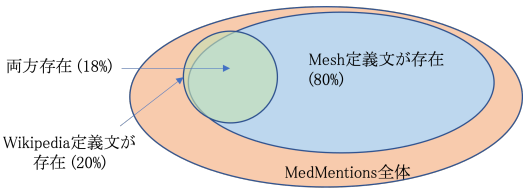

図 2: MeSH および Wikipedia の定義文.

本研究では特に候補エンティティに対するスコアリングに着目し，既存のエンティティ抽出および候補エンティティ生成を用いる．

一方，Wikipedia や Web コーパスから事前学習した言語生成モデルが，要約など多くの生成タスクにおいて良い性能を示している．本研究でも，疑似定義文の生成に Wikipedia から事前学習した BART [6] を用いる．言語生成モデルと EL を組み合わせた手法としては，出力エンティティ名を言語生成モデルで直接生成する手法 [12] が注目されている．これに対し本研究は，既存の分類問題として EL を行う枠組みで学習可能な疑似的な学習データを作成するために言語生成モデルを用いる．

3 提案手法

本研究では，4 種類の定義文を用いる．すなわち，1) UMLS とリンクされた MeSH の定義文，2) UMLS とリンクされた Wikipedia の定義文，3) Wikipedia の定義文から自動生成された疑似的な MeSH 定義文，および 4) MeSH の定義文から自動生成された疑似的な Wikipedia 定義文である．以後，疑似的な定義文を自動生成する手法，ベースラインとして用いる既存モデル MedLinker および複数ソースの定義文を活用するためのモデル拡張について述べる．

3.1 定義文の獲得

本研究では特に MedMentions データセットを用いる設定を扱う．この設定は，PubMed の論文要旨に含まれる言及を，UMLS エンティティに付ける．図 2 に示すように，MedMentions に出現する UMLS エンティティのうち 80%については MeSH の定義文が取得可能である．すなわち，残りの 20%については MeSH の定義文が欠損する．一方，MedMentions に出現する UMLS エンティティのうち 20%については Wikipedia 定義文が取得可能である．さらに MedMentions 全体のうち 18%は MeSH および Wikipedia の両方の定義文が存在する．

本研究では両方から定義文が取得可能な 18%部分を言語生成モデルの学習に用いる．1 つ目は MeSH

の定義文を入力とし，疑似的な Wikipedia 定義文を生成するモデルである．2 つ目は Wikipedia の定義文を入力し，疑似的な Mesh 定義文を生成するモデルである．これらの言語生成モデルは英語版の Wikipedia で事前学習された BART を用い，MeSH および Wikipedia の定義文を生成するよう追加学習を行った．

3.2 ベースモデル: MedLinker

ベースラインとする既存手法 MedLinker [3] について述べる．

MedLinker の学習済み配布モデル [2] を，言及抽出，候補生成に用いる．MedLinker による候補スコアリングは定義文の情報は用いず，1) 入力文中の言及のみを入力として計算するスコアおよび，2) 言及と候補エンティティのコサイン類似度による表層一致スコアを組み合わせる手法である．本研究では 4 つの定義文も考慮するスコアを用いて MedLinker を拡張する．以下に MedLinker の 2 つのスコアについて概説する．

1 つ目のスコア ScoreCLF はニューラルネットワークによるスコアリングであり，以下のように入力文中の言及 ent および候補エンティティ c に対するスコアを計算する:

$$\mathbf{s} = \mathtt{Softmax}(\mathbf{W}_e \mathtt{AverageBERT}(ent)), \quad (1)$$

$$\mathtt{ScoreCLF}(ent, c) = \mathbf{s} \cdot \delta_c. \quad (2)$$

ここで，AverageBERT は入力文全体を読み込みエンティティ部分のサブワードに対する埋め込み表現の平均値を返す．実際には，BERT の最終層だけでなく，その 2 つ前までの層におけるベクトル表現も含め平均ベクトルを計算する．重み行列 $\mathbf{W}_e$ とソフトマックス関数を用いて，学習データに含まれるエンティティ種類数と同じ数の次元を持つ確率分布 $\mathbf{s}$ に変換する．よって，各次元が学習データに含まれる UMLS エンティティとリンクするか否かを表現するスコアとなる．候補エンティティ c に対応する要素のみが 1 をとる one-hot ベクトル δ_c との内積を計算することで，候補エンティティ c に対するスコアを得る．

2 つ目のスコア scoreSTR は ent と c のエンティティ名をそれぞれ文字 1 グラムから 4 グラムを用いて素性ベクトルとし，表層一致度をコサイン類似度を用いて計算した:

$$\mathtt{scoreSTR}(e, c) = cos(e, c), \quad (3)$$

ここで，cos() はコサイン類似度を返す．これら 2 つのスコアの最大値が最終的なスコアとなる．

3.3 定義文を活用するモデル

次に，既存モデルを定義文を用いるよう拡張する．この拡張では，MedLinker での 2 つのスコアに加え，定義文を用いてスコアリングする提案スコア ScoreDef を計算し，3 つのスコアの最大値を最終的なスコアとする: $Score = max(\mathtt{ScoreSTR}, \mathtt{ScoreCLF}, \mathtt{ScoreDef})$. ScoreDef の計算手法として，1) MeSH 定義文を用いる手法，2) MeSH および Wikipedia 定義文を用いる手法，3) 疑似定義文で MeSH および Wikipedia 定義文の欠損を補完する手法，および 4) 別のエンコーダで 4 種類の定義文を読み込む手法を比較する．

1. MeSH 定義文を用いる手法: MeSH の定義文 def_M を用いて候補エンティティ c に対しスコア計算する以下の手法を用いる:

$$\begin{aligned}\mathtt{ScoreDef}(ent, c) = \mathtt{Softmax}(\mathbf{W}_M[&\\ &\mathtt{AverageBERT}(ent);\\ &\mathtt{BERT}_M(def_M)]),\end{aligned}$$

ここで，$\mathtt{BERT}_M$ は候補エンティティの MeSH 定義文を BERT で読み込み，先頭に付与された [CLF] トークンに対する表現ベクトルを返す．また，[;] は結合ベクトルを表し，$\mathbf{W}_M$ は結合ベクトルを 2 次元に圧縮する重み行列である．

2. MeSH および Wikipedia の定義文を用いる手法: さらに，複数種類の定義文を用いる効果を検証するため，Wikipedia の定義文も用いスコアリングするモデルを考える:

$$\begin{aligned}\mathtt{ScoreDef}(ent, c) = \mathtt{Softmax}(\mathbf{W}_{M+W}[&\\ &\mathtt{AverageBERT}(ent);\\ &\mathtt{BERT}_M(def_M); \mathtt{BERT}_W(def_W)]),\end{aligned}$$

ここで，$\mathtt{BERT}_W$ は Wikipedia 定義文先頭の [CLF] トークンに対する埋め込み表現で，$\mathbf{W}_{M+W}$ は結合ベクトルを二次元に圧縮するための重み行列である．

3. 定義文の欠損を疑似定義文で補完する手法: 前述したように MeSH の定義文や Wikipedia 定義文は欠損する．その場合，空のサブトークン列の先頭に [CLF] トークンが付与され読み込まれるため，エンティティの定義に関する情報を活用できない．その

2) https://github.com/danlou/MedLinker

ため，欠損箇所を自動生成した疑似定義文で補完するモデルを提案する:

$$\begin{aligned}\texttt{ScoreDef}(ent, c) \quad = \quad & \texttt{Softmax}(\mathbf{W}_{M+W+Aug}[\\ & \texttt{AverageBERT}(ent); \\ & \texttt{BERT}_M(def_{AugM}); \\ & \texttt{BERT}_W(def_{AugW})]),\end{aligned}$$

ここで，$\texttt{BERT}_{AugM}$ および $\texttt{BERT}_{AugW}$ は MeSH または Wikipedia 定義文が欠損する場合に，疑似定義文で補完したテキストの埋め込み表現を表す．$\mathbf{W}_{M+W+Aug}$ は前述した手法と同様に重み行列である．

4. 4 つの定義文を異なるエンコーダで読み込む手法: 前述した補完による手法は，自動生成した疑似定義文と MeSH もしくは Wikipedia に含まれる定義文を区別せず，それぞれ単一のエンコーダで読み込む．ここで，疑似定義文と本来の定義文は区別し別のエンコーダで読み込む手法を提案する:

$$\begin{aligned}\texttt{ScoreDef}(ent, c) \quad = \quad & \texttt{Softmax}(\mathbf{W}_{M+W+Sep}[\\ & \texttt{AverageBERT}(ent); \texttt{BERT}_M(def_M); \\ & \texttt{BERT}_W(def_W); \\ & \texttt{BERT}_{PseudoM}(def_{PseudoM}); \\ & \texttt{BERT}_{PseudoW}(def_{PseudoW})]),\end{aligned}$$

ここで，$\texttt{BERT}_{PseudoW}$ および $\texttt{BERT}_{PseudoM}$ は疑似定義文をそれぞれ読み込む BERT である．

4 実験

実験には MedMentions st21pv [7] データを用いた．MedMentions は PubMed の論文要旨を UMLS に人手でリンクしたデータを格納している．UMLS エンティティには固有の ID である CUI の他に STR と呼ばれるカテゴリに相当する ID も含むが，本研究ではより難しい問題である CUI アノテーションを正解として用いる．実験には MedMentions データセットの学習，開発および評価用のデータ分割を用いた．ScoreSTR，ScoreCLF の計算には学習済みモデルを用い，提案スコアの ScoreDef は学習セットを用いて学習した．評価指標には MedLinker との比較のため，適合率，再現率，F 値を用いる．

5 結果

表 1 に結果を示す．MedMentions 全体での評価を表の左側に示す．ScoreDef は MeSH および Wikipedia 定義文の両方もしくはいずれかが存在する場合に計算できる．よって，両方が存在しない場合には MedLinker のスコアと変化がない．よって，ScoreDef の計算可能な部分 (図 2 における青と緑部分の和集合) のみでの評価も行う．また，MeSH もしくは Wikipedia の少なくともいずれかの定義文が存在し ScoreDef が計算可能な部分のみでの評価は表 1 の右側に示す．

	MedMentions 全体			ScoreDef が計算可能な部分		
	適合率	再現率	F 値	適合率	再現率	F 値
MedLinker	40.69	59.59	48.36	41.68	63.31	50.27
MeSH のみ	40.92	59.73	48.57	42.10	64.54	50.64
MeSH+Wiki	40.97	59.75	48.61	42.19	63.59	50.72
補完	40.98	59.76	48.62	42.20	63.60	50.74
別エンコーダ	**41.13**	**59.85**	**48.75***	**42.38**	**63.70**	**50.89***

表 1: MedMentions 全体での評価 (左) と MeSH もしくは Wikipedia いずれかもしくは両方の定義文が入手できる部分での評価 (右). * 別エンコーダモデルと UMLS+Wikipedia モデルとの差は統計的に有意 ($p < 0.01$).

MedMentions 全体での評価では，定義文の情報を用いない MedLinker の F 値は 48.36 であるのに対し，定義文を用いる手法では F 値が 48.57 まで向上した．さらに，さらに複数種類の定義文を用いる拡張として Wikipedia の定義文を追加すると F 値が 48.61 まで向上した．疑似的な定義文も用いる 2 つのモデルは補完による手法が 48.62 であり効果が見られなかった．一方，別のエンコーダで読み込むモデルでは 48.75 と性能を向上させ，別エンコーダで読み込む手法が有効であることが分かった．

ScoreDef の計算可能な部分での評価では，MedLinker と疑似定義文を用いる手法の性能差は MedMentions 全体での評価値での性能差より大きい．具体的には MedMentions 全体での評価では MedLinker の F 値からの向上が 0.39 (48.36 から 48.75) なのに対し，ScoreDef が計算可能な部分では 0.62 (50.27 から 50.89) であった．よって，定義文を用いるスコア関数によってスコアが変わる部分に対して，提案した枠組みはより効果的に作用していることがわかる．

6 おわりに

本研究では専門分野向け EL モデルの学習に，言語生成モデルにより作成した疑似的な定義文を活用する新たな枠組みを提案した．実験より，疑似的な定義文であっても EL モデルの性能を向上させることが分かった．この知見は，言語生成モデルの新たな活用法としての新たな可能性を示すものである．

謝辞

この成果は, 国立研究開発法人新エネルギー・産業技術総合開発機構 (NEDO) の助成事業 (JPNP16010) の結果得られたものである．産総研の AI 橋渡しクラウド (ABCI) を利用し実験を行った．

参考文献

[1] Rada Mihalcea and Andras Csomai. Wikify! linking documents to encyclopedic knowledge. In **Proceedings of 29th ACM International Conference on Information and Knowledge Management**, p. 233–242, New York, USA, 2007. Association for Computing Machinery.

[2] Manoj Prabhakar Kannan Ravi, Kuldeep Singh, Isaiah Onando Mulang', Saeedeh Shekarpour, Johannes Hoffart, and Jens Lehmann. CHOLAN: A modular approach for neural entity linking on Wikipedia and Wikidata. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 504–514, Online, April 2021. Association for Computational Linguistics.

[3] Daniel Loureiro and Alípio Mário Jorge. In Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins, editors, **Proceedings of 42nd European Conference on Information Retrieval**.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **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] Afshin Rahimi, Timothy Baldwin, and Karin Verspoor. WikiUMLS: Aligning UMLS to Wikipedia via cross-lingual neural ranking. In **Proceedings of the 28th International Conference on Computational Linguistics**, pp. 5957–5962, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[6] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[7] Sunil Mohan and Donghui Li. Medmentions: A large biomedical corpus annotated with {umls} concepts. In **Proceedings of Automated Knowledge Base Construction (AKBC)**, 2019.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In **SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data**, pp. 1247–1250, New York, NY, USA, 2008. ACM.

[9] Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. Entity disambiguation with freebase. In **Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01**, WI-IAT '12, p. 82–89, USA, 2012. IEEE Computer Society.

[10] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. **Nucleic Acids Res.**, Vol. 32, No. Database-Issue, pp. 267–270, 2004.

[11] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. End-to-end neural entity linking. In **Proceedings of the 22nd Conference on Computational Natural Language Learning**, pp. 519–529, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[12] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In **Proceedings of The Ninth International Conference on Learning Representations (ICLR2021)**, 2021.