日本語 T5 を用いた Entity 辞書のメンション候補自動獲得手法の提案と評価

上田 直生也 [1]　岡 照晃 [1]　杉山 雅和 [2]　邊土名 朝飛 [2]　小町 守 [1]
[1] 東京都立大学　[2] 株式会社 AI Shift
ueda-naoya@ed.tmu.ac.jp, teruaki-oka@tmu.ac.jp
{sugiyama_masakazu, hentona_asahi}@cyberagent.co.jp
komachi@tmu.ac.jp

概要

本稿では Entity 辞書エントリのメンション候補を自動獲得する手法を提案する．タスク指向型の音声対話システムの Entity Linking は，Entity 辞書を知識ベースとして用いる．Entity 辞書はユーザの発話パターンに対応するために，表記揺れや通称，略称などメンション候補登録が必要となるが，構築にコストがかかる問題がある．提案手法は，日本語 T5 を Wikipedia から獲得した見出し語とその同義語対でファインチューニングし，メンション候補生成モデルを構築した．実験の結果，メンション候補を自動獲得した Entity 辞書を用いると，人手によるメンション候補を登録した Entity 辞書と同等の Entity Linking 性能を得られるとわかった．

1　はじめに

本稿が扱うタスク指向型の音声対話システムはユーザ発話に対して応答を生成する．適切な応答生成の実現には，ユーザ発話に含まれる商品名や店舗名などの Entity を正確に認識する必要がある．Entity を正確に認識する手法として，テキスト中に含まれる Entity を知識ベース上に存在するエントリと紐づける Entity Linking がある．実際の処理の流れは，入力されたユーザ発話を自動音声認識システムによりテキスト化する．テキストに含まれる Entity を Entity Linking システムを通して，知識ベースにある適切なエントリに紐づける（図 1 を参照）．

知識ベースには Entity 辞書を用いる．Entity 辞書は，エントリとそのメンション候補が登録された辞書であり，ユーザの多様な発話パターンに対応するために，表記揺れや通称，略称などの様々なフレーズがメンション候補として登録される．しかしながら，Entity 辞書は Entity Linking を行いたい各ドメインに対して人手で構築する必要があるため，メンション候補の登録には多大なコストがかかるという問題がある．

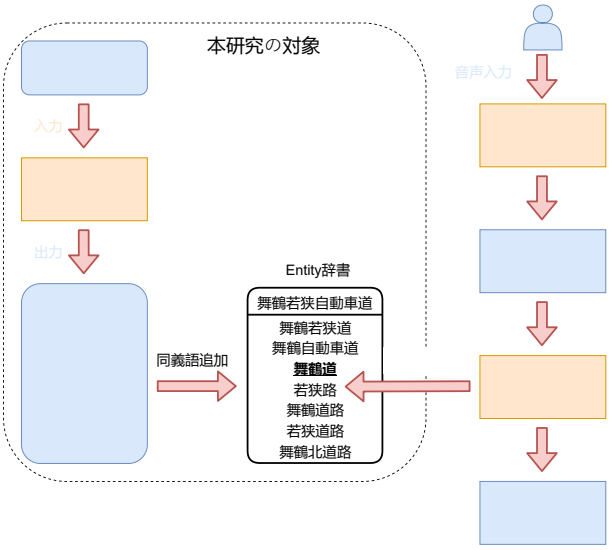

図 1: 想定する音声対話システムのフローチャート

同義語対を自動獲得する手法に関しては，これまで様々な研究が行われてきた [1, 2, 3]．しかしながら，タスク指向型の音声対話システムにおける Entity Linking で用いられる Entity 辞書では，ユーザの多様な発話パターンに対応するために，表記揺れや通称，略称などの様々なフレーズがメンション候補として必要である．そのようなメンション候補は既存手法では獲得することは対象としていない．また前述の手法 [1, 2, 3] では取得したいエントリに関連するドメインにおける大量のコーパスが必要となる．タスク指向型の音声対話システムの場合，対象となるドメインのコーパスが少量しか存在しない場合もあり，メンション候補を獲得できないなどといった問題がある．

そこで本研究では，Entity 辞書におけるエントリのメンション候補を自動獲得する手法を提案する．提案手法は，Wikipedia から獲得した見出し語とその同義語対を T5 [4] で学習することにより，メンション候補生成モデルを構築する．入力されたエントリに対して出力されたメンション候補集合を用いることで，Entity 辞書のエントリに対するメンション候補を登録した．実験では，福井県の道路交通情報に関するシステム主導型の対話ログデータと教師なし Entity Linking システム [5] を使用する．提案手法でメンション候補を自動獲得した Entity 辞書と人手によるメンション候補登録が行われた Entity 辞書で精度を比較し，提案手法の有効性を示した．

2 関連研究

入力単語に対する同義語獲得は，これまで様々な研究が行われてきた．大野ら [6] や Tam ら [7]，柏岡ら [8] は Wikipedia のエントリ–リダイレクトを用いた同義語の獲得を行っている．これらの研究では，Wikipedia のエントリ名とリダイレクト先の見出し語が同義関係にあることに着目し，同義語対として獲得している．しかし，この手法では Wikipedia に存在しないエントリの同義語は獲得できない問題がある．提案手法では獲得した同義語対で言語モデルをファインチューニングし，Wikipedia に存在しない単語に対してもメンション候補を生成できるモデルを構築する．

Transformer [9] 系統の言語モデルを用いて，同義語獲得を行った研究がある．Shen ら [1] は Encoder モデルである BERT [10] を用いることで，Entity とその同義語の集合に対して，新たな同義語を獲得する手法を提案している．Transformer 系統の言語モデルを用いることで同義語獲得を行うといった点では本研究と同じだが，Encoder–Decoder モデルでメンション候補を生成した点と Entity に対する同義語の集合があらかじめ存在しないといった点で異なる．

伴ら [2] は Word2Vec [11] を用いた同義語辞書自動構築手法を提案している．提案手法は，抽出対象である要求仕様書を用いて学習した Word2Vec モデルを用いて，要求仕様書から同義語候補を獲得することで同義語辞書を自動構築している．また，酒井ら [3] は名詞間類似度を考慮することにより原型語に対する略語を獲得する手法を提案している．これらの手法では，特定のエントリに対する同義語を獲得するために，大量のドメインデータが必要となる．一方，本研究ではメンション候補を獲得したいエントリに関連する大量のドメインデータが存在しないという設定の下，Entity 辞書のメンション候補の自動登録に取り組む．

表 1: Wikipedia から獲得した見出し語とその同義語対の例

見出し語	同義語
下井草	下井草一丁目，下井草四丁目，シモイグサ，下井草 15 丁目，下井草村，下井草二丁目，下井草三丁目
宮崎県立延岡商業高等学校	延岡商，延岡商業高，延岡商業高校，延岡商業高等学校

3 手法

本研究では，先行研究 [6, 7, 8] の手法で獲得された Wikipedia の見出し語とその同義語対は，誤表記や表記ゆれなど広義での同義語が含まれていることに着目した．そのデータを用いて Encoder–Decoder モデルである T5 [4] をファインチューニングした．ファインチューニングすることで，モデルによる誤表記や表記揺れのパターンを踏まえたメンション候補の生成を期待する．

3.1 Wikipedia の見出し語と同義語対の獲得

Wikipedia の見出し語とその同義語対は，先行研究 [6, 7, 8] と同様の手法で獲得した．日本語 Wikipedia のダンプデータ [1] としては 2022 年 4 月 25 日時点のもの使用した．獲得した同義語対にはタイプミスなどの再現性が低い誤りが存在した．そのため，各見出し語に対して Wikipedia 全体で 1 回しか出現しなかった同義語対は，再現性のないメンション候補として取り除いた．本研究で獲得した見出し語とその同義語対の例を表 1 に表す．

3.2 メンション候補生成モデル

本研究では，入力されるエントリに対してメンション候補を自動生成するメンション候補生成モデルを構築した．Wikipedia の見出し語を入力とし，その同義語対を出力する形式で日本語 T5 モデルをファインチューニングすることにより構築する．

出力では，一つのエントリに対して複数のメン

1) https://dumps.wikimedia.org/jawiki/

表 2: Wikipedia データセットにおける
エントリ数と同義語対の数

	エントリ数	同義語対
訓練データ	886,854	2,165,428
評価データ	5,000	22,607

ション候補を獲得するため，サンプリングによる生成を行った．これにより，多様なメンション候補の獲得ができる．サンプリング手法としては，Temperature サンプリングや Top-K サンプリング，Top-p サンプリング [12] が存在する．本研究では，Temperature サンプリングを用いて，メンション候補の生成を行う[2)]．

4 実験

4.1 データ

本研究では，3.1 の手法で構築した Wikipedia の見出し語とその同義語対のデータセットを用いた．データセットは訓練データと評価データに分割した．データセットに含まれるエントリ数と同義語対の数を表 2 に示す．Entity Linking を行う対象としては，福井県の道路交通情報に関するシステム主導型の対話ログデータを用いた [5]．対話ログデータは，紐づけ対象となる Entity が正解ラベルとして与えられる．データセットに含まれるユーザ発話数は 2,225 件，紐付け対象となる Entity 数は 214 件である．

4.2 実験設定

T5 は Megagon Labs の公開モデル [3)] をファインチューニングした．ファインチューニングには，Huggingface の公開している Transformers [4)] のスクリプトを用いた．パラメータは学習率を 0.001，バッチサイズを 16，エポック数を 10，最大系列長を 128 と設定し，10 エポック目のモデルを使用した．全ての入力や出力は SentencePiece [13] を用いてサブワード分割を行った．サブワードの語彙サイズは 32,000 に設定した．

2) 多様な出力を行う理由として，多様な出力から適しているものを選択するような用途も考えられるためである．これは，人手による辞書作成では，多様なドメインでメンション候補を作成できるほど知識を持つと限らないからである．

3) https://huggingface.co/megagonlabs/t5-base-japanese-web

4) https://github.com/huggingface/transformers

表 3: Entity 辞書の統計量

辞書	平均メンション候補	カバー率（%）
登録なし	0	0
人手登録	2.0	N/A
自動登録		
T = 0.1	1.6	23.7
T = 1.0	18.1	38.2

メンション候補の生成では，Entity 辞書の各エントリに対し 50 個のメンション候補を出力させた．メンション候補集合から重複を取り除き，Entity 辞書に追加した．また，Temperature（T）は，0.1〜1.0 の間で 0.1 ずつ変動させた．シード値による性能変動が大きいため，複数のシード値で実験を行い，その平均値を結果として示す．シード値は 2022，2032，2042，2052，2062 とした．

メンション候補が登録されていない Entity 辞書（登録なし），人手によるメンション候補登録を行った Entity 辞書（人手登録），メンション候補を自動登録した Entity 辞書（自動登録）を用いた際の Entity Linking 性能について比較を行う．それぞれの Entity 辞書に登録される平均メンション候補数と人手登録されたメンション候補のカバー率を表 3 に示す．

4.3 評価手法

内的評価として，Wikipedia から獲得した見出し語とその同義語対を真とした場合の評価を行った．完全一致を真陽性とみなした Precision，Recall，F1 スコアを算出して評価を行う．外的評価として，各 Entity 辞書を用いた際の Entity Linking 性能で評価を行った．Entity Linking システムとしては，音声類似度を考慮した教師なし Entity Linking システム [5] を用いる．評価指標は Precision at 1（P@1）を用いた．

4.4 実験結果

各 Entity 辞書を用いた実験結果を表 4 に示す．実験の結果，内的評価では Temperature が小さいほど Precision が高くなり，大きいほど Recall が高くなるという結果が得られた．これは，表 6 が示すように Temperature が大きいほど多様なメンション候補を生成したためである．また，外的評価における結果では，T=0.8 のとき最も性能の平均値が高くなり，P@1 で 84.4%であった．この結果は人手によるメンション候補登録した Entity 辞書を用いる場合と

表 4: 実験結果

辞書	内的評価			外的評価
	Precision	Recall	F1	P@1
登録なし	N/A	N/A	N/A	79.9
人手登録	N/A	N/A	N/A	**84.9**
自動登録				
T = 0.1	**30.0**	7.8	12.3	78.5
T = 0.2	23.9	9.8	13.9	79.3
T = 0.3	19.4	11.9	**14.8**	80.8
T = 0.4	15.8	13.7	14.7	81.7
T = 0.5	13.1	15.6	14.2	82.0
T = 0.6	11.1	17.2	13.5	83.0
T = 0.7	9.5	18.7	12.6	83.7
T = 0.8	8.3	20.2	11.8	**84.4**
T = 0.9	7.3	20.9	10.8	84.0
T = 1.0	6.3	**21.5**	9.7	83.8
Best	6.4	**21.8**	9.9	**85.8**

比較しても，−0.5%となっており，本手法によりメンション候補登録した Entity 辞書は人手によるメンション候補登録した Entity 辞書と同等の性能を得られるとわかった．特に T=1.0 のとき，最も内的評価における Recall が高かった Entity 辞書を用いた場合（Best）の Entity Linking の P@1 は 85.8%となっており，人手でメンション候補登録した Entity 辞書の Entity Linking 性能を上回る結果となった．

4.5 考察

内的評価と外的評価の相関 内的評価における Recall と外的評価における P@1 の相関を調べた．図 2 は各シード値ごとの内的評価における Recall と外的評価における P@1 をプロットした結果である．ピアソンの積率相関係数を計測したところ，0.868 となり，強い正の相関があることが確認できた．結果から，内的評価における Recall と外的評価における P@1 には相関があり，メンション候補を自動登録した Entity 辞書の良さは内的評価における Recall を評価することで確認できることを示した．

事例分析 表 5 に提案手法で構築した Entity 辞書を用いた Entity Linking が登録なしや人手登録した辞書と比べて改善・悪化した事例数を示す．Entity Linking が改善される事例が多数存在する一方で，悪化している事例も確認される．Appendix の表 7 と表 8 にメンション候補を自動登録したことにより

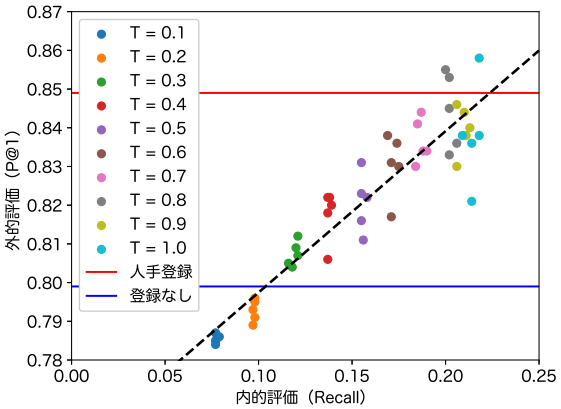

図 2: 内的評価と外的評価における相関
（各シード値ごとの性能をプロットしたときの結果）

Entity Linking が改善・悪化した事例を示す．改善した事例では “(主 6) 福井四ヶ浦線” のメンション候補として “県道 6 号” が登録される．そのため，“県道 6 号 福井 自然が” の入力に対して，正しい Entity を返したことが確認できる．このように提案手法を用いることで Entity 辞書に良いメンション候補が登録され，正しい Entity が出力できる．一方，悪化した事例では “(県 109) 南横地芦原線” のメンション候補として “熊本空港線” が誤って登録される．そのため，“えっと 空港” の入力に対して，誤った Entity Linking がされたことが確認できる．

このように，全体における Entity Linking 性能は向上しているが，個別の事例では誤ったメンション候補の登録による Entity Linking の悪化が確認される．そのため，生成されるメンション候補をフィルタリングする手法などを開発することにより，メンション候補が自動登録された Entity 辞書を用いた Entity Linking の性能が改善すると考えられる．

5 おわりに

本研究では，Entity 辞書のエントリにおけるメンション候補を自動的に獲得する手法を提案した．実験の結果，提案手法でメンション候補を自動獲得した Entity 辞書を用いる Entity Linking は，人手でメンション候補登録した Entity 辞書を用いる場合と同等もしくはそれ以上の性能を得ることができることを示した．また，Entity 辞書の質は Wikipedia データに対する Recall を用いることで評価することが可能であることを示した．今後は Entity 辞書を改善するため，生成されたメンション候補を自動でフィルタリングする手法を検討する．

参考文献

[1] Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. SynSetExpan: An iterative framework for joint entity set expansion and synonym discovery. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 8292–8307, Online, November 2020. Association for Computational Linguistics.

[2] 伴凌太, 高橋宏季, 位野木万里. word2vec を用いた同義語辞書自動作成手法の提案と適用評価. 情報処理学会第 81 回全国大会, pp. 265–266, 2019.

[3] 酒井浩之, 増山繁. 略語とその原型語との対応関係のコーパスからの自動獲得手法の改良. 自然言語処理, Vol. 12, No. 5, pp. 207–231, 2005.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[5] 邊土名朝飛, 戸田隆道, 友松祐太, 杉山雅和, 東佑樹, 下山翔. ユーザ発話と Entity の音声類似度を考慮した Entity Linking 手法の検討. 人工知能学会全国大会論文集, Vol. JSAI2022, pp. 4Yin255–4Yin255, 2022.

[6] 大野潤一, 柴木優美, 山本和英. Wikipedia のエントリ-リダイレクト間を対象にした同義関係抽出. 言語処理学会第 17 回年次大会, pp. 296–299, 2011.

[7] Derek Tam, Nicholas Monath, Ari Kobren, Aaron Traylor, Rajarshi Das, and Andrew McCallum. Optimal transport-based alignment of learned character representations for string similarity. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5907–5917, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] 柏岡秀紀. Wikipedia のリダイレクトから得られる同義語の分析. 言語処理学会第 13 回年次大会, pp. 1094–1096, 2007.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Proceedings of the 31st International Conference on Neural Information Processing Systems**, p. 6000–6010, 2017.

[10] 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, 2019.

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. **CoRR**, Vol. abs/1301.3781, , 2013.

[12] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In **International Conference on Learning Representations**, 2020.

[13] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

A　改善・悪化した事例数

表 5: 提案手法で構築した Entity 辞書を用いることで Entity Linking が改善・悪化した事例数
（最も内的評価における Recall が高かった Entity 辞書（Best）との比較）

辞書	改善	悪化
登録なし	175	44
人手登録	110	90

B　出力例

表 6: 各 Temperature におけるメンション候補生成例
（“(県 142) 松島若葉線” をエントリとして入力したときの出力結果）

Temperature	出力されたメンション候補
T = 0.1	県道 142 号，松島若葉線
T = 0.4	県道 142 号，松島若葉線，若葉線，県道 142 号松島若葉線
T = 0.7	県道 142 号松島若葉線，松島若葉線，若葉線，県道 142 号，県道 142 松島若葉線，県道 142 号線，宮城野比女線
T = 1.0	県道 142 号線，松島若葉線，県道 142 号，若葉線，小松島市広域バス，県道 142 号松島若葉線，松島観光開発線，県道 142 線，県道 142，県道松島若葉線，黒川線，松浦炭礦，産業道路，市バス，県 142，松島線

表 7: メンション候補登録により Entity Linking が改善した例

辞書	入力	期待される Entity	出力された Entity	紐づけられたメンション候補
登録なし	県道 6 号 福井 自然が	(主 6) 福井四ヶ浦線	(県 265) ふくい健康の森線	(県 265) ふくい健康の森線
提案手法	県道 6 号 福井 自然が	(主 6) 福井四ヶ浦線	(主 6) 福井四ヶ浦線	県道 6 号
登録なし	えーと 17 号 丸岡	(主 17) 勝山丸岡線	(主 38) 丸岡インター線	(主 38) 丸岡インター線
提案手法	えーと 17 号 丸岡	(主 17) 勝山丸岡線	(主 17) 勝山丸岡線	17 号
登録なし	県道 31 号線 篠岡 津山線	(主 31) 篠尾勝山線	(県 260) 勝山インター線	(県 260) 勝山インター線
提案手法	県道 31 号線 篠岡 津山線	(主 31) 篠尾勝山線	(主 31) 篠尾勝山線	31 号線

表 8: メンション候補登録により Entity Linking が悪化した例

辞書	入力	期待される Entity	出力された Entity	紐づけられたメンション候補
登録なし	えっと 空港	(県 234) 福井空港線	(県 234) 福井空港線	(県 234) 福井空港線
提案手法	えっと 空港	(県 234) 福井空港線	(県 109) 南横地芦原線	熊本空港線
登録なし	県道 158	国道 158 号	国道 158 号	国道 158 号
提案手法	県道 158	国道 158 号	(主 3) 福井大森河野線	県道 158 号
登録なし	3305	国道 305 号	国道 305 号	国道 305 号
提案手法	3305	国道 305 号	(県 107) 泊小浜停車場線	福井県道 330 号泊小浜停車場線