燃料電池に関する情報抽出への専門知識導入に向けた検討

林大夢 [1] 旭良司 [2] 佐々木裕 [1]
[1] 豊田工業大学 知能数理研究室　[2] 名古屋大学
{sd22073,yutaka.sasaki}@toyota-ti.ac.jp, asahi.ryoji.d9@f.mail.nagoya-u.ac.jp

概要

本稿では，燃料電池に関する文献からの情報抽出において，専門知識を導入する方法について述べる．近年，LLM の性能が向上し，情報抽出も LLM によって実現できるようになってきているが，関係抽出に関するコーパスに関して，用語と関係を抽出する性能は従来の BERT 等をベースとした特化モデルの方が高い性能が得られる．そこで本稿では，Retriever と Reader からなる情報抽出モデル Relik の Retriever 部分を LLM により実現することで，両者の利点を利用することを検討する．現在，複雑な Relik モデルに対するチューニングが不十分なため十分なスコアは出ていないが，実験によりデータを制限したケースでは従来手法よりも高い性能が得られることを示す．

1　はじめに

近年，マテリアルズ・インフォマティクス (Materials Informatics; MI) の研究が盛んに行われている．MI は，機械学習などの情報科学を用いて材料開発の効率を高める取り組みと定義される

MI の学習データは主に論文から抽出され，専門家の手によってラベル付きデータが構築されるが，人手でラベル付けを行うのは非常に時間がかかってしまう．そのため，学習データの収集にも機械学習モデルを使用し，論文から自動的にデータを収集する研究が行われている．

しかし，論文から物性情報などのデータを自動抽出する精度は未だ低く，実用的な段階には達していない [1]．物性情報の抽出の精度が低くなる原因として，機械学習モデルがドメイン知識を十分に考慮できていないことが挙げられる．論文は専門分野特有の用語が多く含まれるため，テキストの埋め込み表現が不十分であり，情報を十分に抽出できないと考えられる．

学習済み知識の制約と文脈理解の限界を克服する方法として，本研究では大規模言語モデル（LLM）の活用に着目する．LLM は，その広範な事前学習データに基づく知識と高度な文脈理解・推論能力を有している．LLM を活用することで，複雑な文章に対する文脈の理解力を向上させ，専門知識に基づいた情報抽出を実行できると期待される．

本研究では，文献からの燃料電池における情報抽出において従来の手法では抽出できていない情報を抽出することを目的とする．

2　関連研究

2.1　DyGIE++

DyGIE++[2] は NER, RE およびイベント抽出を行う情報抽出モデルである．BERT[3] をエンコーダに利用して文脈化された単語埋め込みを得るとともに，グラフ構造によって文章全体の語句の関係をとらえることを目的としたモデルである．文章中のスパンをノードとしてスパングラフを構築し，イテレーションごとにエッジをはるノードを動的に変更し，タスクにおけるノード間の関係をもとに情報を伝播させノードの埋め込み表現を更新する．この手法により，DyGIE++は異なる情報抽出タスクで利用可能な動的スパングラフを構築し，従来手法を超える性能を実現した．

2.2　ReLiK

ReLiK[4] はエンティティリンキング (Entity Linking; EL) と関係抽出 (Relation Extraction;RE) を行う情報抽出モデルである．ReLiK の提案手法の概要図を図 1 に示す．ReLiK は Retriever と Reader という 2 つの独立したモデルにより構成される．

Retriever では事前に設定された取り出し対象であるエンティティや関係のうち，入力文に含まれる候補の取り出しを行っている．ReLiK の Retriever は Dense Passage Retriever(DPR)[5] という手法を用いている．DPR では入力されたテキストと設定された

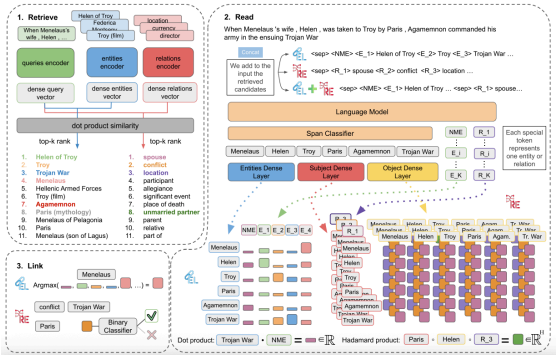

図 1　ReLiK の提案手法の概要図 ([4] より引用)

エンティティ，関係に対して，同じ埋め込み空間上で密な数値ベクトルでの表現を行い，その埋め込み表現を元に候補の取り出しを行う．入力されたテキストを q，設定されたエンティティ，関係の集合を D_p，その集合に含まれる設定されたエンティティ，関係を $p \in D_p$ とすると，Retriever は以下の式で埋め込み表現への変換を行う．

$$E_Q(q) = Retriever(q), E_P(p) = Retriever(p) \quad (1)$$

$E_Q(q)$ と $E_P(p)$ を得るための Retriever のエンコーダは同一のモデルが使用される．また，Retriever ではエンコーダとして E5[?] を用いている．得られた埋め込み表現に対して，内積 $sim(q,p) = E_Q(q)^T E_P(p)$ を計算することにより入力テキストと設定されたエンティティ，関係との類似度を求める．この類似度 $sim(q,p)$ をもとに，テキスト中に含まれるエンティティ，関係の候補の取り出しを行う．

Retriever を学習する際には，目的関数として multi-label noise contrastive estimation(NCE) を用いている．目的関数 $\mathcal{L}_{Retriever}$ は以下の式で表される．

$$\mathcal{L}_{Retriever} = -\sum_{p^+ \in \bar{D}_p(q)} log \frac{e^{sim(q,p^+)}}{e^{sim(q,p^+)} + \sum_{p^- \in P_q^-} e^{sim(q,p^-)}} \quad (2)$$

ここで，$\bar{D}_p(q)$ は入力テキスト中に含まれる正解のエンティティ，関係の集合，P_q^- は in-batch negatives と hard negative mining によって得られたエンティティ，関係の集合である．in-batch negatives では，学習中に同じミニバッチに含まれる他の入力テキスト中における正解のエンティティ，関係を対象テキストの負例として扱う手法である．hard negative mining では，入力テキスト中に含まれない負例のエンティティ，関係のうち，類似度 $sim(q,p^-)$ が最も高いものを負例として学習する．

Reader では，Retriever によって取り出されたエンティティ，関係の候補がどのスパンに対応しているかの紐付けを行う．ReLiK の Reader は従来の手法とは異なり，一度の順方向の処理で全てのエンティティ，関係の候補の計算を行うことで計算速度を向上させている．まず，Retriever で計算された入力テキストとエンティティ，関係の類似度をもとに，類似度の高い候補を top-K で取り出す．取り出した K 個の候補 $p_{1:K} = (p_,..., p_K), p_i \in D_p$ と入力テキスト q を組み合わせて，一つのシーケンス $q\ [SEP]\ \langle ST_0 \rangle\ \langle ST_1 \rangle\ p_1 ... \langle ST_K \rangle\ p_K$ を構成する．ここで $[SEP]$ は入力テキストとエンティティ，関係の候補列を区切るための特殊トークン，$\langle ST_i \rangle$ は i 番目の候補であることを表現するための特殊トークンである．このシーケンスをエンコーダに通すことで埋め込み表現 X を得る．Reader ではエンコーダに DeBERTa-v3[?] を用いている．

$$X = Tr(q\ [SEP]\ \langle ST_0 \rangle\ ...\ p_K) \in \mathbb{R}^{l \times H} \quad (3)$$

ここで，$l = |q| + 1 + (1 + K) + \sum_k |p_k|$ は入力シーケンスの合計トークン長を表している．エンコーダから得られた入力テキストとエンティティ，関係の候補の埋め込み表現を用いて，候補とスパンの対応付けを行う．まず，入力テキスト中の全スパンのうち，固有表現を表すスパンを識別する．テキスト中のそれぞれのトークンに対して，固有表現を表すスパンの先頭のトークンである確率 $p_S(s|X)$ を計算する．

$$p_S(s|X) = \sigma_0(W_S^T X_s + b_S) \quad \forall s \in \{1, ..., |q|\} \quad (4)$$

ここで，$W_S \in \mathbb{R}^{H \times 2}, b_S \in \mathbb{R}^2$ は学習可能パラメータ，$X_s \in \mathbb{R}^H$ は埋め込み表現 X のうち s 個目のトークンを表す s 番目の行，σ_0 は softmax 関数を表す．続いて，固有表現を表すスパンの終端のトークンである確率 $p_E(t|X,s)$ を計算する．

$$p_E(t|X,s) = \sigma_0(W_E^T X_m + b_E) \quad \forall t \in \{s, ..., |q|\} \quad (5)$$

ここで，$W_E \in \mathbb{R}^{2H \times 2}, b_E \in \mathbb{R}^2$ は学習可能パラメータ，$X_m \in \mathbb{R}^{2H}$ は X_s, X_t を結合したベクトルを表す．これらの式により得られた確率をもとに，閾値での二値分類によって固有表現を表すスパンの識別を行う．スパンの識別を行った後は，タスクごとに異なる処理が行われる．

2.3　ORR 触媒データセット

ORR 触媒データセット [6] は燃料電池の研究における酸化還元反応 (Oxidation-Reduction Reaction) を促進する触媒に関する論文の文章から構成された，

表 1 NER でベースの結果と提案手法を用いたモデルの結果の比較

モデル	データ	F1 スコア
ベース	評価データ	47.49
	テストデータ	33.44
提案手法 1	評価データ	47.04
	テストデータ	33.02
提案手法 2	評価データ	32.14
	テストデータ	25.03

表 2 RE でベースの結果と提案手法を用いたモデルの結果の比較

モデル	データ	F1 スコア
ベース	評価データ	40.04
	テストデータ	27.55
提案手法 1	評価データ	39.20
	テストデータ	27.05
提案手法 2	評価データ	41.15
	テストデータ	28.06

固有表現抽出 (Named Entity Recognition;NER) と RE のデータセットである．ORR 触媒データセットは，12 種類のエンティティタイプと 2 種類の関係のアノテーションにより構築されている．

また，Htet らは ORR 触媒データセットを構築するのに加え，DyGIE++[2] を用いて ORR 触媒データセットでの情報抽出性能の評価を行った．F1 スコアを用いた評価では，NER が開発データで 63.81%，テストデータで 61.66%，RE が開発データで 53.29%，テストデータで 51.27%という実験結果が得られた．

3 提案手法

3.1 Retriever への専門知識の導入

ReLiK の Retriever は取り出したいエンティティや関係を事前に設定する必要がある．その際に，エンティティ，関係のラベルに対する説明文を与えることができる．この構造を活用して，LLM が生成した各エンティティ，関係の説明文を ReLiK に与える．LLM にはインストラクションチューニングを施し，材料科学のなかでも燃料電池の ORR 触媒の文献という専門性を考慮した定義分を生成させる．

3.2 LLM を利用したエンティティおよび関係候補の取り出し

ReLiK の Retriever を LLM に置き換えることで，柔軟なエンティティ，関係候補の取り出しを行う．LLM にはインストラクションチューニングとともに，Few-shot 学習を施す．Few-shot の例題にエンティティが含まれない文を含めることで，LLM が候補の取り出しというタスクを正しく行うことができるように調整を行う．

4 実験と結果

4.1 データ制限なし

LLM を用いて，専門知識を導入する手法 (提案手法 1) とエンティティ，関係の候補の取り出しを行う手法 (提案手法 2) の実装，評価を行った．表 1，表 2 に示すように提案手法 1 では NER，RE ともにベースの性能よりもわずかに劣るという結果となった．また，提案手法 2 では NER は大きく性能が下がるものの，RE では性能の向上が見られた．これらの結果から，Retriever でのエンティティタイプと関係の候補の取り出しの精度が最終的な情報抽出の性能に大きく寄与するものの，エンティティタイプ，関係の定義分は取り出しの精度に大きな影響を及ぼさないと考えられる．

4.2 データ制限あり

ORR 触媒データセットに含まれる 8398 個のセンテンスのうち，5311 個が固有表現を含まない．そこで，固有表現を含まないセンテンスを手動で分別し，残った 3078 個のセンテンスを用いて実験を行った結果を表 3，表 4 に示す．

前節の結果と同様に，NER では提案手法 2 の性能が大きく下がるものの，RE では性能の向上が見られる結果となった．また，データの制限をしていない時の結果と比べると，NER，RE の両方で大きな性能の向上が見られた．このことから，Retriever で候補の取り出しを行う前に，固有表現を持たないセンテンスの識別と除外を行うモデルを導入することで，性能の向上が期待できる．

5 おわりに

本研究では，燃料電池の ORR 触媒に関する文献から情報抽出をする際の性能を向上させることを目的として，LLM を利用して専門知識を導入した ORR

表 3 データを制限した時の，NER でベースの結果と提案手法を用いたモデルの結果の比較

モデル	データ	F1 スコア
ベース	評価データ	72.12
	テストデータ	65.66
提案手法 2	評価データ	48.25
	テストデータ	48.34

表 4 データを制限した時の，RE でベースの結果と提案手法を用いたモデルの結果の比較

モデル	データ	F1 スコア
ベース	評価データ	56.15
	テストデータ	48.40
提案手法 2	評価データ	56.48
	テストデータ	48.00

触媒の情報抽出を行う手法を提案した．今後の課題として，プルーニングを行うモデルの導入，LLM のハルシネーションへの対応，プロンプト改善，メモリ使用量を節約する手法などが挙げられる．

謝辞

この成果は，NEDO（国立研究開発法人新エネルギー・産業技術総合開発機構）の委託業務（JPNP25002）の結果得られたものです．

参考文献

[1] Htet, et al. Extracting orr catalyst information for fuel cell from scientific literature, 2025.

[2] David, et al. Entity, relation, and event extraction with contextualized span representations. In **EMNLP-IJCNLP**, 2019.

[3] Jacob, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In **NAACL**, 2019.

[4] Orlando, et al. Relik: Retrieve and link, fast and accurate entity linking and relation extraction on an academic budget, 2025.

[5] 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.

[6] Htet, et al. Information extraction of orr catalyst for fuel cell from scientific literature. Vol. JSAI, , 2025.

A 付録

A.1 エンティティタイプ，関係の定義分を出力させるプロンプト

```
You are an expert in Fuel Cells,
    specificcaly focusing on
    oxidation-reduction reactions (ORR)
    and catalytic chemistry.
You are creating a entity type definition
    to train the Retriever module of a
    Natural Language Processing model
    (ReLiK).
for the list of "entiy types" provided by
    the user, please generate a definition
    for each.

**Requirements:**
1. Considering the context of academic
    papers on fuel cells, explain the role
    the entity plays in experiments or
    theoretical frameworks.
2. Don't generate sentences that related
    to not fuel cells, but materials
    science.
3. Output the result as a list in the
    folowing jsonl format:
    [
        {"text": "<entity_type_1>",
            "definition":
            "<definition_1>"},
        {"text": "<entity_type_2>",
            "definition":
            "<definition_2>"},
        ...
    ]
```

A.2 LLM にエンティティタイプと関係の候補を取り出させるプロンプト

```
You are an expert in Fuel Cells,
    specificcaly focusing on
    oxidation-reduction reactions (ORR)
    and catalytic chemistry.
You are retrieving entity types from the
    given texts that are part of academic
    papers on fuel cells.
the texts and list of entity types
    provided by the user, please retrieve
    the entity types from texts.

**Target Entity Types**
The only allowed entity types are: 省略{~~}
Do not use any other entity types not
    listed above keys.

**Requirements:**
1. For each train, developmetn, and test
    dataset, the data is provided in the
    following format:
    [
        {"words":
            <list_of_words_in_text_1>},
        {"words":
            <list_of_words_in_text_2>},
        ...
    ]
    Please extract candidate entity types
        from the text within each
        dictionary.
2. Although these are materials science
    terms, please disregard any words
    unrelated to fuel cells.
3. Please answer using the examples
    provided as a reference.
4. Output the result as a list in the
    folowing jsonl format:
    [
        {"words":
            <list_of_words_in_text_1>,
            "entity_types":
            <list_of_candidate_entity_types_in_text_1>
        {"words":
            <list_of_words_in_text_2>,
            "entity_types":
            <list_of_candidate_entity_types_in_text_2>
        ...
    ]
5. Please exercise caution when
    determining whether a sentence
    contains an entity type, as it may not
    always be present.
6. When enclosing strings, always use
    double quotation marks instead of
    single quotation marks.
```