非構造知識検索を用いた自己適応型固有表現認識

西田 光甫[1,2,a)]　吉永 直樹[3]　西田 京介[1]

概要：固有表現認識（NER）は，人名や組織名のようなドメインに依存しないエンティティだけでなく，政治ドメインにおける選挙や音楽ドメインにおけるアルバムなど，目標ドメイン固有のエンティティを抽出・分類するために活用できる．しかしながら，個々のユーザが自身の興味のある目的ドメインにおいて，高精度の NER を行うために必要な大規模な訓練データや構造化された知識ベースを構築し，高精度な NER を実現することは難しい．そこで本稿では，生のテキスト集合である非構造知識から，個々の入力文に対して必要な知識テキストを都度検索する，自己適応型固有表現認識を提案する．提案モデルは，まず入力のみから固有表現抽出を行ったのち，確信度の低いラベルを含むエンティティをクエリとした知識検索を行い，知識を用いた予測によって元の予測を改善する 2 段階のモデルである．CrossNER データを用いた評価実験により，提案モデルがベースラインを F_1 で 2.35 ポイント上回ることを確認した．

1. はじめに

固有表現抽出（NER）は，テキスト中の情報をエンティティに注目して整理する言語処理技術であり，情報抽出で用いられる基本的技術の一つである．NER は医療 [1]，化学 [2]，COVID-19 [3] など様々なドメインにおけるエンティティを抽出・分類できる．しかし，個々のドメインで高精度な NER モデルを学習するためには，そのドメインにおける大量の訓練データが必要である [4], [5], [6]．さらに，ドメイン特有のエンティティに関する訓練データを作成するためには多くの場合ドメインの専門家が必要であり，訓練データの作成コストは高い．情報抽出の対象となるドメインは多岐に渡るため，訓練データのコストは情報抽出を多様なドメインで行う際の障壁となっている．

NER における訓練データの不足を補う手段としては，外部知識の利用が一般的である．人手で設計した素性に基づく NER モデルでは，gazetteer や name list などの整備された知識に基づいた素性を利用して分類を行う [7], [8], [9]．これらの外部知識は，ニューラルモデルにおいても有効であることが知られている [10], [11], [12] が，知識を整備することには相応のコストを要する．そこで，生のテキスト集合に自動でラベルを付ける弱教師あり学習手法が提案されている [12], [13], [14] が，あらゆる世界知識を弱教師ラベルから学習してモデルに保存することは現実的ではない．

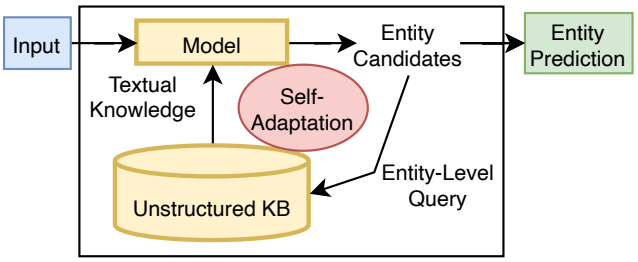

図 1: 自己適応型固有表現認識の概念図．モデルはエンティティ候補を予測してからエンティティレベルの検索を非構造知識ベースから行う．次に，モデルは知識を参照しながら予測を改善する．

本論文では，生のテキスト集合を構造化されていない知識（以降，**非構造知識**）として推論時に利用する NER モデルを提案する．我々のアイディアは，近年の retrieval-augmented 言語モデル [15] に触発されている．これらのモデルは主にオープンドメイン質問応答を下流タスクとして想定し，質問をクエリとして用いることでテキスト集合から知識テキストを検索し，知識テキストと元の入力を繋げて言語モデルに入力することで回答を出力する．しかし，評価実験に示すように，オープンドメイン質問応答のために設計されたモデルは NER における性能の改善幅が小さい．これは，NER では入力が多くのエンティティを含むため，入力中のどのエンティティに注目して知識を検索すればよいかが非自明であることが原因であると考えられる．

この問題に対処するため，我々は入力中の注目すべきエンティティを動的に決定しながら知識検索を行う retrieval-

[1] NTT 人間情報研究所
[2] 東京大学
[3] 東京大学生産技術研究所
[a)] kosuke.nishida.ap@hco.ntt.co.jp

augmented 言語モデルとして，自己適応型固有表現認識モデルを提案する．表 1 に示す提案モデルは 2 段階の予測を行う．つまり，まず元々の入力のみで NER を解き，エンティティの候補とエンティティラベルの予測確率を確信度として得る．次に一定の確信度以下のエンティティ候補をクエリとした知識検索を行い，非構造知識ベースから知識テキストを得る．最後に，知識テキストを元の入力に繋げて言語モデルに入力し，新たな予測結果を得る．2 段階の予測を用いることで，モデルが理解できないエンティティに焦点を当てて効率よく知識を参照できる．

NER における知識検索の有効性を検証するため，7 ドメインの NER データセットでの実験を行った [16], [17], [18]. 5 ドメインは訓練データ数が 200 以下の低資源データセットやドメイン固有のエンティティラベルを含んでいる．

本研究の貢献を以下に示す．

- 本研究は NER に知識テキストの検索を初めて導入した．自己適応型固有表現認識は，エンティティレベルの知識検索を個々の入力に対して動的に行う．
- 評価実験において，提案手法は知識検索を行わないモデルと既存の retrieval-augmented 言語モデルの NER 性能を上回った．NER におけるエンティティレベルの検索の重要性を示した．
- 提案手法が特に事前学習コーパスに含まれないエンティティに有効であることを示した．そのため，提案手法はドメイン特有のエンティティの分類に資する．

2. タスク設定

NER は系列タグ付けタスクである．語彙を V，入力長を L としたとき，入力は $X \in V^L$，出力は BIO ラベルの長さ L の系列である．分類対象のクラス数を C として，BIO 方式でタグ付けを行う場合，ラベルの種類は $2C+1$ である．

自己適応型固有表現認識では, 非構造知識ベースとして生のコーパスを用いる．コーパスは長さ L のトークン系列に分割されるものとする．これは, 既存の retrieval-augmented 言語モデル [19] に準じた分割方法であり，知識を効率的に保持できる．知識検索によって m 個の知識テキスト K_i を検索した後，それらの知識を元の入力 X に繋げてモデルに入力する．よって，モデルへの入力長は $(m+1)L$ である．

3. 関連研究

関連研究として，非構造知識を用いる NER と retrieval-augmented 言語モデルについて述べる．

3.1 非構造知識を用いた NER

これまで，NER のために様々な形で生のテキストが非構造知識として用いられてきた．古典的な NER モデルでは入力の周辺文脈 [20], [21], [22] やリンク先文書 [23] を非局所的な知識として活用している．一方，近年のニューラル NER モデルではより広い文脈を考慮した単語埋め込みを計算するため近傍の文を動的単語埋め込みに利用した [24], [25]．文献 [26], [27] は，UMLS Meta-thesaurus などのドメインの一致する構造知識から質問，定義，事例といったエンティティに関するテキストを抽出して活用した．これらの研究が入力と明示的な手がかりで紐づくテキストを非構造知識として活用しているのに対し，本研究では，NER において，テキスト集合を非構造知識として検索して活用する手法の確立を目指す．

前述の手法がテキストを非構造知識として用いるのに対して，疑似ラベルを付与したテキストを利用して訓練を行うアプローチが試みられている [12], [13], [14], [28], [29]. これらの手法は，目標ドメインのテキストを弱教師としてモデル学習に利用でき，モデルはドメイン固有のエンティティやクラスを学習できる [14] が，疑似ラベルを付与するための構造知識の存在を仮定している．また，これらの手法は訓練を通じて目的ドメインの知識を獲得するが，提案手法は推論時に目標ドメインのテキストを与えることでモデルに保存しきれない世界知識を補完できる．評価実験で確認するように，提案手法は弱教師あり事前学習と併用することでさらに NER 性能を改善する．

3.2 Retrieval-Augmented 言語モデル

モデルにない知識を補うため，外部知識として入力と関連する生のテキストを検索して用いる言語モデルが提案されている [15], [19], [30], [31], [32]．しかし，これらの手法は言語モデルとオープンドメイン質問応答で評価されており，NER における結果は報告されていない．これらの手法は，タスクの入力全体または入力を固定長に分割したチャンクから知識検索のためのクエリを構築している．しかし，入力が複数のエンティティを含みうる NER タスクでは，モデルが追加知識を要するエンティティがどのエンティティであるかは自明ではない．そのため，これらの手法を素朴に NER に適用した場合，追加知識を要するエンティティに焦点を当てたクエリを作成できない．

文献 [33], [34] は，関連知識を検索するときに訓練データ自身を検索対象とすることで，モデルに保存しきれなかった知識を推論時に改めてモデルに与えている．これらの研究を参考に，提案手法では生のテキストから作成する教師なし非構造知識ベースだけでなく，訓練データを教師あり非構造知識ベースとして用いる．

文献 [35] は要素のキーとバリューを埋め込み表現とした知識ベースである仮想知識ベースを NER で用いることを提案した．仮想知識ベースの埋め込み表現は，Wikipedia ハイパーリンクの周辺テキストから学習されている．しかし，仮想的な知識ベースを用いる手法は効率的である一方，知識をテキストのままエンコーダ [31] に入力する手法の方が高い精度を示すことが報告されている．

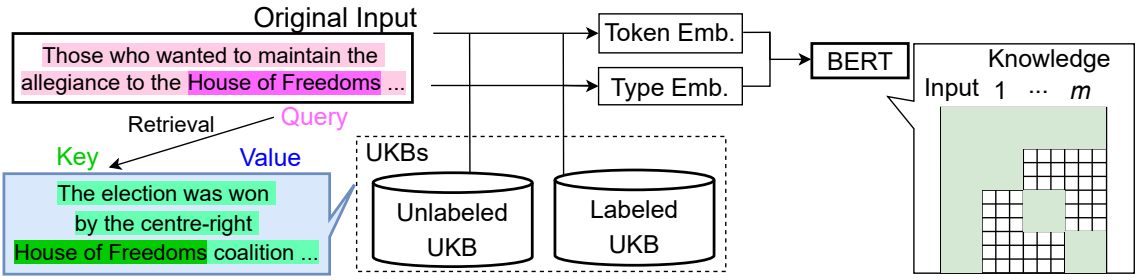

図 2: 自己適応型固有表現認識の構成．非構造知識ベースは n-gram 埋め込みと文埋め込みをキー，テキストをバリューとして持つ．入力と知識と知識ラベルはトークンタイプ埋め込みで区別する．クエリは入力の文埋め込みと確信度の低いエンティティの埋め込みである．BERT 内部でスパースな attention を用いて，複数の知識を高速にエンコードする．

4. 提案手法

本節では，図 2 に示す自己適応型固有表現認識モデルを提案する．以降で，非構造知識ベースの構成（§4.1），エンコーダの構成（§4.2），確信度の低いエンティティに関する知識検索のための 2 段階 NER タグ付けの手法 (§4.3)，訓練手法（§4.4），事前学習手法（§4.5）の順に説明する．

4.1 非構造知識ベース

本研究における非構造知識ベースは 2 種類の知識ベースからなる．第一の知識ベースは目的ドメインのテキストを収集した生のコーパスである．第二の知識ベースは訓練データ自身である．訓練データを知識源としても用いることは文献 [33] の知見に基づいており，モデルに保存しきれなかった知識を補う効果がある．両知識ベースは，入力長上限 L に分割した知識テキストをバリューとする．

1 つの知識テキストに，n-gram 埋め込みと文埋め込みの複数のキーを用意する．文献 [36] の知見に基づき，n-gram 埋め込みと文埋め込みは言語モデルの入力と出力を繋げたベクトルのトークン平均とする．全ての n-gram をキーとして保存することにはコストがかかるので，ストップワードを含まず大文字を含む n-gram のみをキーとして保存する．

4.2 エンコーダ

我々は BERT [37] に線形変換層を加えたものを入力 X と検索して得られた非構造知識 K_i のエンコーダとして用いる．元の入力 X・教師なし知識・教師あり知識とそのラベルを区別するために，トークンタイプ埋め込み t_i を用いる．本来 BERT は 2 種のトークンタイプしか持たないが，提案手法では $2C+3$ 種の埋め込みを学習する

$$t_i = \begin{cases} 0 & \text{if } x_i \text{ is the original input} \\ 1 & \text{if } x_i^+ \text{ is unlabeled knowledge} \\ l_i + 2 & \text{if } x_i^+ \text{ is labeled knowledge} \end{cases}.$$

ここで，l_i は教師あり非構造知識ベースのラベル，X^+ は入力と検索された知識を繋げたテキストである．

self-attention 操作には系列長の 2 乗の計算量が必要なため，知識を挿入した際の計算量は $\mathcal{O}(m^2L^2)$ と知識数の 2 乗オーダになる．そこで，スパースな attention 行列を用いることで計算量を $\mathcal{O}(mL^2)$ に削減し時空間コストを抑え，多くの知識を挿入可能にする．

具体的な実装としては，知識間の相互作用を計算する非対角ブロックをマスクすることでスパースな attention 行列 A を実現する．形式的には，k をその位置のトークンが元々の入力なら 0 を，知識なら $1, ..., m$ を返す関数として

$$A_{ij} = \begin{cases} \frac{\boldsymbol{Q}_i^\top \boldsymbol{K}_j}{\sqrt{d_k}} & \text{if } k(i) = k(j) \text{ or } k(i)k(j) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

である．ここで，i, j はトークン位置，d_k は attention ヘッドの次元数，$\boldsymbol{Q}$ と $\boldsymbol{K} \in \mathbb{R}^{(m+1)L \times d_k}$ はクエリとキーの隠れ状態である．

4.3 自己適応型固有表現認識における 2 段階タグ付け

本稿で提案する自己適応型固有表現認識モデルは，$\boldsymbol{P} = f(X)$ と $\boldsymbol{P}^+ = f(X^+)$ と 2 段階のタグ付けを行う．第一段階の目的は，知識を追加することが必要なエンティティ候補を発見して検索クエリとすることである．第二段階の目的は，検索された知識に基づいて予測を改善することである．提案モデルは，(1) NER のための検索を実現するため，入力全体を検索クエリとするのではなく，入力中の個々のエンティティに着目する検索を行うこと，(2) 検索結果から不要な知識を省くため，知識の必要なエンティティのみを検索クエリとすること，に特徴を持つ．モデルの擬似コードを Algorithm 1 に示す．$\boldsymbol{P}^+$ の位置 L 以降は無視して，$\boldsymbol{P}, \boldsymbol{P}^+$ は共に長さ L のベクトル系列とする．f のパラメータは 2 段階で共有する．

4.3.1 Step1: 知識を検索するクエリを構成するための NER

まず，入力 X から計算した $\boldsymbol{P}$ に基づき確信度の低いエンティティの集合 $\mathcal{U}$ を得る．まず，$\mathcal{E}$ を予測ラベル $\hat{\boldsymbol{y}} = \text{argmax}_c \boldsymbol{P}_{\cdot c} \in \mathbb{N}^L$ に基づいて得るエンティティ e の集合とする．次に，$\mathcal{U} \subset \mathcal{E}$ を，確信度スコア $c_e = \min_{i \in I_e} P_{i,\hat{y}_i}$ が閾値 λ_{conf} より小さいエンティティの集合とする．ただ

Algorithm 1 自己適応型固有表現認識

Require: 入力 X, 知識ベース, ハイパーパラメータ m, λ_{conf}

1: Predict probability $\boldsymbol{P} = f(X)$
2: Compute confidence score $c_e = \min_{i \in I_e} P_{i,\hat{y}_i}$ for each predicted entity $e \in \mathcal{E}$ with span I_e
3: Obtain unconfident entities $\mathcal{U} = \{e | e \in \mathcal{E}, c_e < \lambda_{\text{conf}}\}$
4: Add the sentence and unconfident-entity embeddings to the queries, Q
5: Initialize the retrieval results $R = \Phi$
6: **for** query q_i in the queries, Q **do**
7: Retrieve m nearest-neighbor keys for q_i from the KBs
8: Store their values with the distance in R
9: **end for**
10: Deduplicate R to obtain top-m knowledge K_1^m from R
11: Output probabilities $\boldsymbol{P}$ and $\boldsymbol{P}^+ = f(X^+ = [X; K_1^m])$

し，I_e はエンティティ e のスパンである．また，[B-LOC, I-PER] のようにエンティティ内でエンティティクラスに関する一貫性がなかった場合，$c_e = 0$ とする．

次に，検索クエリを作成する．検索クエリには，文埋め込みとエンティティ埋め込みの 2 種類がある．文埋め込みは文中のトークン埋め込みの平均である．エンティティ埋め込みは，確信度の低いエンティティ $u \in \mathcal{U}$ とトークンを共有する n-gram の埋め込みである．この n-gram は，非構造知識ベース同様に大文字とストップワードによるフィルタリングを行う．この n-gram の総数を E とする．トークン埋め込みは BERT の入出力の結合である．ここで，文埋め込みに由来するクエリは，非構造知識ベースのキーのうち文埋め込みに由来するキーの検索のみ，エンティティ埋め込みに由来するクエリは n-gram 埋め込みに由来するキーの検索のみに用いる．クエリごとに m 個の最近傍の知識を検索した後，マージした $2(E+1)m$ 個の知識から重複を除き，距離の小さい順に m 個の知識を出力する．

4.3.2 Step2: 検索で得た知識を用いた NER

第二段階の予測を $\boldsymbol{P}^+ = f(X^+)$ に基づいて行う．モデルが出力する BIO ラベルは，第一段階で確信度が高かったエンティティに関しては $\boldsymbol{P}$ を用い，それ以外の全てのトークンに関しては $\boldsymbol{P}^+$ を用いる．

4.4 訓練手法

訓練時は，確信度の低いエンティティの集合 $\mathcal{U}$ に，第一段階の予測クラスが誤っているエンティティも加える．損失関数にはクロスエントロピー損失を用いる．第一，第二の予測に基づく損失関数を $\mathcal{L}_1$，$\mathcal{L}_2$，ハイパーパラメータを λ_1 として，モデル全体の損失関数は $\mathcal{L}_2 + \lambda_1 \mathcal{L}_1$ である．

4.5 事前学習手法

retrieval-augmented 言語モデルの一般的な学習手順 [15], [19] に則り，fine-tuning の前に知識検索を用いた事前学習を行った．事前学習の手法として，大規模 NER データの CoNLL03 [16] を用いた学習と，Wikipedia ハイパーリンクと DBPedia オントロジー[*1]によって作成する弱教師 NER データの NERBERT [14] を用いた学習の 2 種を試した．NERBERT の作成方法は原著論文に譲る．

事前学習中の提案手法による知識検索では，非構造知識ベースを事前学習コーパス自身とした．ラベル情報を 95% の確率で削除して教師なし知識ベースとした．効率化のため，2 段階の予測は用いずに，入力とエンティティを共有する m 文をランダムにサンプリングして検索結果とした．

5. 評価実験

F_1 を評価指標として，自己適応型固有表現認識の有効性を調査した．

5.1 データセット

CrossNER [18] は AI，music，literature，science，politics の 5 個のドメインから構成された小規模データセットである．CrossNER は Wikipedia から作成しており，目標ドメインへの適応のために同ドメインの Wikipedia コーパスを公開している．我々はこのコーパスを教師なし知識ベースとした．CrossNER はドメインごとにエンティティクラスの定義が異なる．

Finance [17] はアメリカの SECfillings から収集した中規模の金融ドメインのデータセットである．教師なし知識として Wikipedia から金融ドメインのデータを収集した．エンティティクラスは person, organization, location, miscellaneous の 4 種である．

CoNLL03 [16] は最も広く用いられる NER データセットの一つである．このデータセットは Reuters ニュースの 1996 年 8 月から 1 年の記事を元に作成されている．教師なしテキストとして，1987 年までの Reuters ニュースが収集された Reuters-21578 データセット [38] を用いた．エンティティクラスは Finance と同一である．

5.2 比較手法

比較手法は，知識を用いないモデルと，代表的な retrieval-augmented 言語モデルである REALM を NER データで学習したモデル（REALM-NER）である．ここで，REALM は入力文の文埋め込みをクエリとして検索した 1 つの知識を入力に繋げる言語モデルである．従って，REALM-NER は提案手法からエンティティレベルの検索，教師あり非構造知識ベースを除いて $m = 1$ に設定したモデルである．REALM は通常マスク化言語モデルで学習されるが，NER タスクで知識検索を学習する提案手法と公平に評価するた

[*1] https://databus.dbpedia.org/dbpedia/mappings/instance-types/

表 1: テストセットにおける主結果．モデルは CoNLL03 または NERBERT で事前学習した BERT-base-cased である．実験は 5 種の異なるランダムシードで行い，標準偏差を括弧内に示す．既存文献の数値を文献 [18]† と文献 [14]‡ から引用した．BERT on NERBERT は NERBERT の我々のコーパスにおける追実験に相当する．

	AI.	Mus.	Lit.	Sci.	Pol.	Avg.	Fin.	CoNLL03
# Train (# NE types)	100 (14)	100 (13)	100 (12)	200 (17)	200 (9)	—	1169 (4)	14987 (4)
BERT†	50.37	66.59	59.95	63.73	66.56	61.44	—	—
DAPT†	56.36	73.39	64.96	67.59	70.45	66.55	—	—
NERBERT‡	60.39	76.23	67.85	71.90	73.69	70.01	—	—
BERT on CoNLL03	56.97 (1.05)	69.10 (1.08)	64.37 (0.73)	65.76 (0.58)	70.16 (0.56)	65.27 (0.80)	72.35 (5.32)	—
REALM-NER on CoNLL03	58.05 (1.15)	71.17 (0.63)	64.58 (0.69)	66.33 (0.66)	69.38 (0.36)	66.56 (0.80)	70.03 (1.35)	—
Ours on CoNLL03	**60.31** (1.03)	**72.20** (0.79)	**66.23** (1.30)	**68.22** (0.57)	**71.18** (0.57)	**67.62** (0.85)	**74.02** (2.29)	—
BERT on NERBERT	62.05 (0.66)	76.45 (0.90)	69.68 (0.26)	72.10 (0.67)	74.38 (0.40)	70.93 (0.58)	75.05 (7.47)	90.25 (0.11)
REALM-NER on NERBERT	64.32 (0.31)	77.55 (0.69)	70.42 (0.60)	72.52 (0.42)	74.45 (0.38)	71.85 (0.43)	73.34 (1.74)	89.94 (0.42)
Ours on NERBERT	**65.27** (0.95)	**78.71** (0.47)	**71.79** (0.57)	**74.38** (0.19)	**74.63** (0.36)	**72.96** (0.51)	**75.77** (1.01)	**90.49** (0.49)

表 2: テストセットにおける主結果．モデルは DistilBERT-base-cased である．

	AI.	Mus.	Lit.	Sci.	Pol.	Avg.	Fin.	CoNLL03
DistilBERT on CoNLL03	54.16 (1.21)	66.64 (0.54)	60.53 (1.26)	64.14 (0.49)	67.61 (0.70)	62.61 (0.84)	68.78 (6.35)	—
REALM-NER on CoNLL03	53.85 (1.38)	67.03 (0.41)	61.83 (1.38)	64.19 (0.17)	69.09 (0.52)	63.20 (0.54)	70.35 (5.04)	—
Ours on CoNLL03	**55.31** (1.03)	**67.25** (1.14)	**61.53** (1.18)	**65.71** (1.03)	**69.36** (0.55)	**63.83** (0.99)	**72.89** (2.71)	—
DistilBERT on NERBERT	59.52 (0.89)	71.60 (1.05)	63.52 (0.47)	69.26 (0.97)	68.88 (0.64)	66.56 (0.80)	73.36 (4.17)	89.23 (0.19)
REALM-NER on NERBERT	60.39 (0.53)	71.39 (0.33)	62.89 (0.19)	68.18 (0.83)	69.79 (0.82)	66.53 (0.54)	74.35 (5.06)	88.54 (0.62)
Ours on NERBERT	**61.90** (0.38)	**73.61** (0.45)	**65.48** (0.31)	**70.44** (0.69)	**69.95** (0.90)	**68.27** (0.55))	**75.40** (1.46)	**89.50** (0.30)

め，REALM の訓練も NER タスクで行った．全てのモデルの事前学習は，CoNLL03 または NERBERT を用いた．また，既存研究から BERT, NERBERT, DAPT の数値を引用した．DAPT は教師なし非構造知識をコーパスとした事前学習を行ったドメイン適応済み BERT である．全ての実験は CrossNER の politics ドメインで決定した表 3 に示すハイパーパラメータを用いた．

NERBERT 事前学習は NVIDIA Quadro RTX 8000 (48GB) GPU 8 枚で 17 時間を要した．fine-tuning の訓練時間は，最も大きい CoNLL データセットでも 1 枚の GPU で 6 時間であった．データ数の多い CoNLL データセットでのみ early stop の patience を 5 に設定した．最適化手法は Adam [39]，実装は PyTorch (ver. 1.10.1)[*2] [40] と transformers (ver. 4.15.0)[*3] [41] を用いた．ストップワードは NLTK (ver. 3.7)[*4] [42] のものを用いた．最近傍ベクトルの検索は faiss (ver. 1.7.2)[*5] [43] を用いた．全てのデータでスライディングウインドウを 16 に設定した上で最大系列長 $L = 64$ とした．

[*2] https://pytorch.org/
[*3] https://github.com/huggingface/transformers
[*4] https://www.nltk.org/
[*5] https://github.com/facebookresearch/faiss

表 3: ハイパーパラメータ．

	Pre-Training	Fine-Tuning
Batch size	1024	16
# Epochs	1	300
# Steps	10000	—
# Early stop	—	5/8
m	2	10
n	3	3
λ_{conf}	—	0.9
λ_1	—	0.1
Learning rate	5e-5	5e-5

5.3 主結果

表 1，表 2 に主結果を示す．

提案手法は NER 性能を向上するか？提案手法は全ての目標ドメイン，モデル，事前学習データセットでベースラインモデルの性能を上回った．訓練データ数が少なくエンティティタイプ数の多いドメインほど性能向上が顕著であった．よって，訓練データからの学習が困難なデータセットであるほど，知識検索を用いることが有効と言える．

提案手法は retrieval-augmented 言語モデルの NER 性能を上回るか？提案手法は REALM-NER の性能を上回った．

表 4: politics ドメインの評価データにおける ablation study. Δ は提案手法からの下落を示す．それぞれの構成要素を取り除いてから fine-tuning と評価を行った．

Method	Acc	Δ
Ours	77.33 (0.19)	
w/o Entity-level Retrieval	76.21 (0.23)	1.12
w/o Sentence-level Retrieval	76.54 (0.48)	0.79
w/o Confident Entities (*i.e.*, $\lambda_{\text{conf}} > 1$)	76.91 (0.24)	0.42
w/o using First-Step Prediction on $\mathcal{E} \setminus \mathcal{U}$	76.97 (0.33)	0.36
w/o Unlabeled Knowledge	76.23 (0.44)	1.10
w/o Labeled Knowledge	76.82 (0.45)	0.51
NERBERT	75.90 (0.22)	1.43

提案手法はエンティティレベルの検索を行う，教師あり知識ベースを利用する，複数の知識をエンコードする，といった NER のためにデザインされた構造をもつため，汎用的な retrieval-augmented 言語モデルよりも NER に適している．個々の構成要素の貢献は後述の ablation study で検証する．また，訓練データ数が 1000 個以上のデータセットでは REALM-NER が BERT の性能を下回ることがある．これは，十分に NER 能力を訓練データから獲得した BERT に対して，文レベルのクエリで検索した知識を 1 つだけ追加する REALM-NER が有効ではないことを示す．

5.4 Ablation Study

表 4 に ablation study の結果を示す．提案手法の全ての要素が性能向上に寄与していることがわかる．

エンティティレベルの検索は有効か？エンティティレベルの検索は，通常の retrieval-augmented 言語モデルで用いられる文レベルの検索と比較して，性能向上に寄与している（Δ1.12 vs. Δ0.79）．また，提案手法のように併用することで性能はさらに向上する．

確信度に基づくエンティティのフィルタリングは有効か？提案手法は確信度の低いエンティティ $\mathcal{U} = \{e | c_e < \lambda_{\text{conf}}, e \in \mathcal{E}\}$ のみをクエリとして知識検索を行う．ここで，$\lambda_{\text{conf}} > 1$ と設定することで，全てのエンティティを検索クエリとして利用できる．結果，確信度の高いエンティティを除くことが有効であった (Δ0.42)．これは，m 個の知識の中に不要な知識が入ることを防ぐ効果があるためと言える．次に，第二段階の予測 $\boldsymbol{P}^+$ を確信度の高いエンティティを含む全てのトークンに対して用いた．結果，確信度の高いエンティティに対しては，無関係な知識がエンコードされない第一段階の予測 $\boldsymbol{P}$ を用いることが有効であった（Δ0.36）．確信度に基づくフィルタリングは，m の小さい設定でより有効であると考えられる．

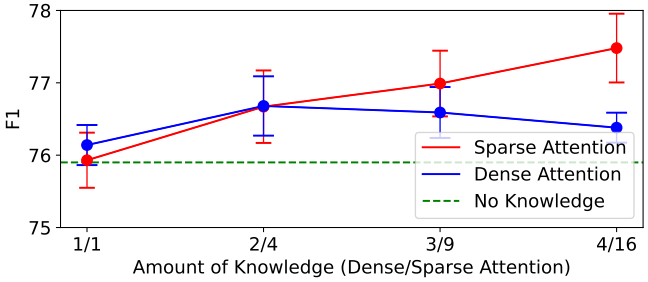

図 3: 知識数 m に対する F_1 スコア．エラーバーは標準偏差を示す．計算量について公平に評価するため，密な attention の知識数 m に対してスパースな attention の知識数は m^2 とした．

表 5: 評価セットにおけるエンティティタイプごとの結果．

	# Entities	NERBERT	Proposed
All	3472	75.90 (0.22)	77.33 (0.19)
Seen in Training	661	84.05 (1.43)	85.20 (0.21)
Unseen in Training	2811	71.39 (0.29)	73.03 (0.36)
Seen in Pre-Training	3083	77.58 (0.17)	78.83 (0.29)
Unseen in Pre-Training	389	50.90 (1.63)	54.18 (1.85)

教師あり・なし非構造知識ベースは有効か？ 2 種の知識ベースを用いることは有効であった (Δ1.10 と Δ0.51)．教師なし非構造知識ベースは広範な知識を含み，教師あり知識ベースはラベル情報を含む．それぞれの知識ベースは NER モデルを異なる側面から補助できるため，組み合わせることで性能が向上したと考えられる．

5.5 議論

提案手法の性能は知識の数に依存するか？図 3 は知識数 m ごとの F_1 スコアをプロットしている．スパースな attention を用いたときは，知識数が多いほど性能が向上している．一方で，密な通常の attention を用いた場合は知識数が性能向上に寄与するとは限らない．m が小さい値のうちは知識数による性能向上が限定的であるが，大きな m を設定することが性能向上に重要であると言える．スパースな attention は知識数を増やすことが容易であるため，自己適応型固有表現認識に適している．

外部知識が有効なエンティティは何か？エンティティを，訓練・事前学習データに表記が同じエンティティが含まれたかによって分類した．表 5 にエンティティタイプごとの結果を示す．提案手法は全てのタイプでベースラインの性能を上回った．ablation study で 2 種の知識が有効であることを確認した通り，エンティティが訓練データに含まれるかどうかに関わらず提案手法は性能向上に寄与する（1.15, 1.64 ポイント）．

特に性能向上が大きいタイプは，事前学習コーパスに含まれないエンティティであった（3.28 ポイント）．NERBERT

表 6: 質的評価，検索された m 個の知識のうち 1 つを示す．

Input	the Association for the Rose in the Fist of Lanfranco Turci and those who wanted to maintain the allegiance to the House of Freedoms coalition.
Knowledge	The election was won in Sardinia by the centre-right House of Freedoms coalition ... voted party with 30.2% .
Prediction	organization → political party
Input	Director Michael Moore partnered with producers Harvey Weinstein and Bob Weinstein in May 2017 to produce and distribute Fahrenheit 11/9 .
Knowledge	... Bob Weinstein, the founders of Miramax Films.
Prediction	politician → politician

の事前学習コーパスに含まれない低頻度のエンティティでは，ベースラインモデルは性能を大きく落とす．提案手法は低頻度のエンティティに対しても，関連する知識を検索することで精度良く分類できる．

5.6 質的評価

表 6 にモデルの出力例を示す．第一の例は自己適応によって予測が改善した例である．元の入力からは the House of Freedoms が政党であることを判断できないが，知識が選挙の文脈で the House of Freedoms に言及することで，提案手法は政党として認識している．第二の例は politics ドメインで最も多い間違いである，人と政治家の誤分類である．入力からも知識からも Bob Weinstein が政治家でないことが判断できるが，モデルは訓練データ中の分布に影響を受けて政治家と分類している．訓練データにおけるバイアスの存在は，知識の活用だけでは解決できない課題である．

6. おわりに

本稿では，NER においてエンティティレベルの知識検索を用いる自己適応型固有表現認識を提案した．提案手法は，2 段階のタグ付けを行うことでエンティティレベルの知識検索を実現する．提案手法は DistilBERT と BERT モデルの性能を 1.22 から 2.35 ポイント向上した．

ドメイン固有のエンティティとエンティティクラスが存在することは，NER 技術を用いて多様なドメインのテキストを対象とした情報抽出を行う際に大きな障壁となっている．提案手法は非構造知識ベースを生のテキストから作成するため，様々なドメイン，さらには様々なユーザに特化した NER モデルを作成でき，情報抽出の適用領域を大きく拡大する．さらに，提案手法は学習後に非構造知識ベースを更新することが容易であるため，経時的な知識やユーザの興味の変化への適応も効率的に行うことができる．

謝辞 本研究の一部（第二著者）は JSPS 科研費 JP21H03494 の助成を受けたものです．

参考文献

[1] Kim, J.-D., Ohta, T., Tateisi, Y., Jun'ichiTsujii: GENIA corpus—a semantically annotated corpus for bio-textmining, *Bioinformatics*, Vol. 19, No. suppl_1, pp. i180–i182（オンライン），入手先 ⟨https://doi.org/10.1093/bioinformatics/btg1023⟩ (2003).

[2] Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D. M. et al.: The CHEMDNER corpus of chemicals and drugs and its annotation principles, *Journal of cheminformatics*, Vol. 7, No. 1, pp. 1–17 (2015).

[3] Wang, X., Song, X., Li, B., Guan, Y. and Han, J.: Comprehensive Named Entity Recognition on CORD-19 with Distant or Weak Supervision, *arXiv preprint arXiv:2003.12218*, (online), DOI: https://doi.org/10.48550/arXiv.2003.12218 (2020).

[4] Chiu, J. P. and Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs, *TACL*, Vol. 4, pp. 357–370 (online), DOI: 10.1162/tacl_a_00104 (2016).

[5] Ma, X. and Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, *ACL*, pp. 1064–1074 (online), DOI: 10.18653/v1/P16-1101 (2016).

[6] Yadav, V. and Bethard, S.: A Survey on Recent Advances in Named Entity Recognition from Deep Learning models, *COLING*, pp. 2145–2158 (online), available from ⟨https://aclanthology.org/C18-1182⟩ (2018).

[7] Florian, R., Ittycheriah, A., Jing, H. and Zhang, T.: Named entity recognition through classifier combination, *CoNLL@HLT-NAACL*, pp. 168–171 (online), available from ⟨https://aclanthology.org/W03-0425⟩ (2003).

[8] Cohen, W. W. and Sarawagi, S.: Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods, *KDD*, pp. 89–98 (online), DOI: https://doi.org/10.1145/1014052.1014065 (2004).

[9] Luo, G., Huang, X., Lin, C.-Y. and Nie, Z.: Joint Entity Recognition and Disambiguation, *EMNLP*, pp. 879–888 (online), DOI: 10.18653/v1/D15-1104 (2015).

[10] Seyler, D., Dembelova, T., Del Corro, L., Hoffart, J. and Weikum, G.: A Study of the Importance of External Knowledge in the Named Entity Recognition Task, *ACL*, pp. 241–246 (online), DOI: 10.18653/v1/P18-2039 (2018).

[11] Liu, T., Yao, J.-G. and Lin, C.-Y.: Towards Improving Neural Named Entity Recognition with Gazetteers, *ACL*, pp. 5301–5307 (online), DOI: 10.18653/v1/P19-1524 (2019).

[12] Mengge, X., Yu, B., Zhang, Z., Liu, T., Zhang, Y. and Wang, B.: Coarse-to-Fine Pre-training for Named Entity Recognition, *EMNLP*, pp. 6345–6354 (online), DOI: 10.18653/v1/2020.emnlp-main.514 (2020).

[13] Cao, Y., Hu, Z., Chua, T.-s., Liu, Z. and Ji, H.: Low-Resource Name Tagging Learned with Weakly Labeled Data, *EMNLP-IJCNLP*, pp. 261–270 (online), DOI: 10.18653/v1/D19-1025 (2019).

[14] Liu, Z., Jiang, F., Hu, Y., Shi, C. and Fung, P.: NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging, *arXiv preprint arXiv:2112.00405*, (online), DOI: https://doi.org/10.48550/arXiv.2112.00405 (2021).

[15] Guu, K., Lee, K., Tung, Z., Pasupat, P. and Chang, M.: Retrieval Augmented Language Model Pre-Training, *ICML*, pp. 3929–3938 (online), available from ⟨https://proceedings.mlr.press/v119/guu20a.html⟩ (2020).

[16] Tjong Kim Sang, E. F. and De Meulder, F.: Introduction

to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *CoNLL*, pp. 142–147 (online), available from ⟨https://aclanthology.org/W03-0419⟩ (2003).

[17] Salinas Alvarado, J. C., Verspoor, K. and Baldwin, T.: Domain Adaption of Named Entity Recognition to Support Credit Risk Assessment, *ALTA*, pp. 84–90 (online), available from ⟨https://aclanthology.org/U15-1010⟩ (2015).

[18] Liu, Z., Xu, Y., Yu, T., Dai, W., Ji, Z., Cahyawijaya, S., Madotto, A. and Fung, P.: CrossNER: Evaluating cross-domain named entity recognition, *AAAI*, pp. 13452–13460 (online), available from ⟨https://ojs.aaai.org/index.php/AAAI/article/view/17587⟩ (2021).

[19] Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Driessche, G. v. d., Lespiau, J.-B., Damoc, B., Clark, A. et al.: Improving language models by retrieving from trillions of tokens, *arXiv preprint arXiv:2112.04426*, (online), DOI: https://doi.org/10.48550/arXiv.2112.04426 (2021).

[20] Sutton, C. and McCallum, A.: Collective Segmentation and Labeling of Distant Entities in Information Extraction, *SRL@ICML* (2004).

[21] Finkel, J. R., Grenager, T. and Manning, C.: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling, *ACL*, pp. 363–370 (online), DOI: 10.3115/1219840.1219885 (2005).

[22] Krishnan, V. and Manning, C. D.: An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition, *COLING*, pp. 1121–1128 (online), DOI: 10.3115/1220175.1220316 (2006).

[23] Plank, B., Hovy, D., McDonald, R. and Søgaard, A.: Adapting taggers to Twitter with not-so-distant supervision, *COLING*, pp. 1783–1792 (online), available from ⟨https://aclanthology.org/C14-1168⟩ (2014).

[24] Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F. and Pyysalo, S.: Multilingual is not enough: BERT for Finnish, *arXiv preprint arXiv:1912.07076*, (online), DOI: https://doi.org/10.48550/arXiv.1912.07076 (2019).

[25] Luoma, J. and Pyysalo, S.: Exploring Cross-sentence Contexts for Named Entity Recognition with BERT, *COLING*, pp. 904–914 (online), DOI: 10.18653/v1/2020.coling-main.78 (2020).

[26] Banerjee, P., Pal, K. K., Devarakonda, M. and Baral, C.: Knowledge guided named entity recognition for biomedical text, *arXiv preprint arXiv:1911.03869*, (online), DOI: https://doi.org/10.48550/arXiv.1911.03869 (2019).

[27] Li, X., Feng, J., Meng, Y., Han, Q., Wu, F. and Li, J.: A Unified MRC Framework for Named Entity Recognition, *ACL*, pp. 5849–5859 (online), DOI: 10.18653/v1/2020.acl-main.519 (2020).

[28] Mintz, M., Bills, S., Snow, R. and Jurafsky, D.: Distant supervision for relation extraction without labeled data, *ACL-IJCNLP*, pp. 1003–1011 (online), available from ⟨https://aclanthology.org/P09-1113⟩ (2009).

[29] Liang, C., Yu, Y., Jiang, H., Er, S., Wang, R., Zhao, T. and Zhang, C.: BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision, *KDD*, p. 1054–1064 (online), DOI: 10.1145/3394486.3403149 (2020).

[30] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S. and Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, *NeurIPS* (Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. and Lin, H., eds.), pp. 9459–9474 (online), available from ⟨https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf⟩ (2020).

[31] Izacard, G. and Grave, E.: Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering, *EACL*, pp. 874–880 (online), DOI: 10.18653/v1/2021.eacl-main.74 (2021).

[32] Singh, D., Reddy, S., Hamilton, W., Dyer, C. and Yogatama, D.: End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering, *NeurIPS*, pp. 25968–25981 (online), available from ⟨https://proceedings.neurips.cc/paper/2021/file/da3fde159d754a2555eaa198d2d105b2-Paper.pdf⟩ (2021).

[33] Wang, S., Xu, Y., Fang, Y., Liu, Y., Sun, S., Xu, R., Zhu, C. and Zeng, M.: Training Data is More Valuable than You Think: A Simple and Effective Method by Retrieving from Training Data, *ACL*, pp. 3170–3179 (online), DOI: 10.18653/v1/2022.acl-long.226 (2022).

[34] Shinzato, K., Yoshinaga, N., Xia, Y. and Chen, W.-T.: Simple and Effective Knowledge-Driven Query Expansion for QA-Based Product Attribute Extraction, *ACL*, pp. 227–234 (online), DOI: 10.18653/v1/2022.acl-short.25 (2022).

[35] de Jong, M., Zemlyanskiy, Y., FitzGerald, N., Sha, F. and Cohen, W.: Mention Memory: incorporating textual knowledge into Transformers through entity mention attention, *ICLR*, (online), available from ⟨https://arxiv.org/abs/2110.06176⟩ (2022).

[36] Huang, J., Tang, D., Zhong, W., Lu, S., Shou, L., Gong, M., Jiang, D. and Duan, N.: WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach, *Findings of EMNLP*, pp. 238–244 (online), DOI: 10.18653/v1/2021.findings-emnlp.23 (2021).

[37] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *NAACL*, pp. 4171–4186 (online), DOI: 10.18653/v1/N19-1423 (2019).

[38] Lewis, D. D.: Reuters-21578 Text Categorization Test Collection, Distribution 1.0 (1997).

[39] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *ICLR (Poster)*, (online), available from ⟨http://arxiv.org/abs/1412.6980⟩ (2015).

[40] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A.: Automatic differentiation in PyTorch, *Autodiff@NIPS*, (online), available from ⟨https://openreview.net/forum?id=BJJsrmfCZ⟩ (2017).

[41] Wolf, T. et al.: Transformers: State-of-the-Art Natural Language Processing, *ACL: System Demonstrations*, pp. 38–45 (online), available from ⟨https://www.aclweb.org/anthology/2020.emnlp-demos.6⟩ (2020).

[42] Bird, S., Klein, E. and Loper, E.: *Natural Language Processing with Python*, O'Reilly Media, Inc. (2009).

[43] Johnson, J., Douze, M. and Jégou, H.: Billion-Scale Similarity Search with GPUs, *IEEE Transactions on Big Data*, Vol. 7, No. 3, pp. 535–547 (online), DOI: 10.1109/TBDATA.2019.2921572 (2021).