人工知能学会 インタラクティブ
情報アクセスと可視化マイニング研究会(第16回)
SIG-AM-16-01

カテゴリ情報を付与した文の分散表現による逆引き辞書の精度向上

Inprovement in Accuracy of Reverse Dictionary by Predicting Target Word's Category

森永 雄也[1*]　山口 和紀[1]
Yuya MORINAGA[1]　Kazunori YAMAGUCHI[1]

[1] 東京大学
[1] The University of Tokyo

Abstract: 逆引き辞書は、ある概念を説明した文を入力するとその文が表す概念に対応する単語を出力するシステムである。Hill et al. (2016) は辞書データからの学習により、入力文に含まれる単語列のベクトル群を、行列、あるいは Recurrent Neural Network により変換して、入力文を単語の word2vec 分散表現空間のベクトルとして表現する関数を作成し、この関数によって入力の "解釈" を行い単語を検索する分散表現逆引き辞書モデルを提案した。本研究は、WordNet で定義されている lexname をカテゴリとして用いて、入力文が表す単語の（人が見れば推定可能な）カテゴリ情報を Hill のモデルが利用できていない可能性を指摘し、仮にこのモデルが正しいカテゴリ内の単語に絞って検索を行うことができれば精度が大きく向上することを示した。更に、Kim (2014) の Convolutional Neural Network によるテキスト分類を応用してカテゴリ情報を推定する手法、及びそれを用いたカテゴリ推定分散表現逆引き辞書モデルを提案し、検索精度を向上させることに成功した。

1 はじめに

一般的な辞書は、単語を定義にマッピングする。これとは逆に、ある概念の定義あるいは説明をその概念を表す単語にマッピングするシステムが逆引き辞書である。逆引き辞書により、舌先現象（Tip of the Tongue Problem[8]）のような、意味はわかるが名前が思い出せない（あるいは知らない）という状況を解決することができる。クロスパードパズルのソルバ [2] としても応用例がある。逆引き辞書の困難な点は、入力候補である文が無数に存在することである。入力文はある単語の説明になっているという前提を考えると入力候補は限られるが、それでも同じ単語を説明する文を網羅することは難しい。例えば、WordNet[5] における 'brother' の定義は 'a male with the same parents as someone else' となっているが、逆引き辞書はこの定義文だけでなく 'son of my parents' という説明文についても 'brother' へのマッピングを行わなければならない。このため、逆引き辞書の実装には、未知の入力文と出力候補となる各単語との「近さ」を計算できる機構が必要となる。

Node-Graph Architecture モデル [9] は、辞書データを用いて「ある単語が別の単語の定義中に現れる」という関係から単語の隣接行列を作り、得られたグラフ構造上で入力文中の単語を表すノード群から探索を行うことで逆引き辞書を実現するモデルである。このモデルにおいては、グラフのパスの長さが単語間の距離、辞書データ中での頻度の低さが単語の重要度とされ、入力文中でより重要な単語と距離が近い候補が優先的に出力される仕組みになっている。このモデルによる逆引き辞書は、扱う語彙を 3,000 語に絞った評価において、既存の商用逆引き辞書である OneLook Reverse Dictionary[1] を上回る性能を示している。

より大規模な語彙について実験を行ったものとして、分散表現逆引き辞書モデル [2] がある。このモデルは、word2vec 単語ベクトルと辞書データからの学習により、単語ベクトル空間に入力文を射影する、すなわち単語の意味空間上で入力文の意味を分散表現することによって「解釈」を行い単語を検索するモデルである。これは定義中の各単語の分散表現ベクトル群を入力として、定義される語に近いベクトルを出力する関数を学習するというものである。 関数としては線形変換と Recurrent Neural Network (RNN) [4] による変換の 2 種類が用いられている。この関数によって入力文を「解釈」して検索を行う逆引き辞書は、人が記述した説明文を用いた評価において、OneLook Reverse Dictionary

*連絡先： 東京大学 総合文化研究科
〒 153-8902 東京都目黒区駒場 3 丁目 8 − 1
E-mail: morinaga@graco.c.u-tokyo.ac.jp

[1] http://onelook.com/reverse-dictionary.shtml

と比べ遜色のない性能を示した。本研究はこのモデルを用いて実験を行った。

逆引き辞書は、単語の説明による検索に特化した質問応答システムの一種と見ることもでき、従って一般の高精度な質問応答システムの手法を応用することで性能の改善が期待できる。Watson [7] は、アメリカのクイズ番組 *Jeopardy!* において人間に勝利した質問応答システムである。Watson は質問文の分析によって答えのカテゴリを推定することで回答候補を絞り込み、候補内での順位付けに基づいて回答を行うという手法を採用している。Watson の詳細な仕様は公開されていないが、同様のアプローチを取ることは可能である。そこで著者らは、分散表現逆引き辞書モデルに入力文の分散表現とは独立なカテゴリ推定による絞り込み機能を加えることで精度が向上するのではないかと考えた。

まず、カテゴリ情報が分散表現逆引き辞書モデルの検索結果に与える影響を調べるため、カテゴリと word2vec 単語空間の関係を分析したところ、word2vec 空間においては様々なカテゴリに属する単語のベクトルが重なり合って分布していることがわかった。このことから、word2vec 空間上で文を分散表現すると、その文が表す概念の（本来は推定可能であったかもしれない）カテゴリ情報が抜け落ちてしまうことが予想できる。そこで、仮にカテゴリ情報を適切に推定できたとすると、分散表現逆引き辞書モデルによる単語検索のスコアが大きく上がることも確認できた。

この結果を受けて、本研究はカテゴリ推定と分散表現逆引き辞書モデルを併せたカテゴリ推定逆引き辞書モデルを提案する。

カテゴリ推定には word2vec 単語ベクトルを用いた Convolutional Neural Network（CNN）テキスト分類モデル [3] を利用する。このモデルを採用したのは、入力文を 1 つの分散表現に落とし込む前の情報、すなわち入力文に含まれる全単語ベクトルの情報を用いればカテゴリ情報を推定できるのではないかという考えに基づく。具体的には、WordNet から取った synset（WordNet における情報のユニットで、一つの「概念」に対応する）の定義と lexname（lexicographer file name, 45 種類のカテゴリタグ）の訓練用ペアデータから、word2vec 単語ベクトルを用いて、CNN によるテキスト分類モデルの学習を行い、カテゴリ推定器とした。

このカテゴリ推定器と分散表現逆引き辞書モデルを併せたのが、提案モデルであるカテゴリ推定分散表現逆引き辞書モデルである。Hill が配布しているユーザディスクリプションを用いた評価により、提案モデルのスコアが分散表現逆引き辞書モデルのスコアを上回ることが確認できた。この結果から、文章からカテゴリを推定し、推定したカテゴリ内の単語を対象に word2vec 空間上での分散表現で検索を行うことで、より精度の高い単語検索が可能であることが示された。

次章以降の構成は以下のようになっている。第 2 章では先行研究である分散表現逆引き辞書モデルと CNN テキスト分類モデル [3] についての説明を行う。第 3 章では word2vec 単語ベクトルのカテゴリ分析を行い、分散表現逆引き辞書モデルの問題点と提案モデルの指針を明確にする。第 4 章では本研究で提案するモデルの詳細について述べる。第 5 章では実験の詳細と結果、考察を記述し、第 6 章で本研究のまとめと課題、今後の展望を述べて結びとする。

2 先行研究

本章では、本研究がベースとした分散表現逆引き辞書モデルとテキスト分類モデルを概説する。2.1 節では分散表現逆引き辞書モデルについて、2.2 節ではテキスト分類モデルについて述べる。以下、単語 w の（縦）ベクトルを v_w と書き、モデルへの入力を $s = (w_1, w_2, ..., w_n)$ とする。また、以下の$\phi(x)$ は活性化関数で、任意の非線形な微分可能関数を指すが、本研究では各先行研究に倣い$\phi(x) = \tanh(x)$ を採用している。

2.1 文の分散表現モデル

本節では入力文の分散表現モデルについて説明する。これにより分散表現した入力文のベクトルとの cosine 距離を用いて候補となる単語のランク付けを行い出力するのが分散表現逆引き辞書モデルである。

2.1.1 Bag of Words (BOW)

BOW 分散表現モデルは、入力文中の単語ベクトルの総和を線形変換したものを入力文の分散表現として出力するモデルであり、以下のように表される。ここで、v の次元を D として、W は $D \times D$ の行列で、A_j は D 次元のベクトルである。

$$\begin{aligned} A_1 &= W v_{w_1} \\ A_{i(>1)} &= A_{i-1} + W v_{w_i} (= W \sum_{j=1}^{i} v_j) \end{aligned}$$

W は線形変換を行うので、長さ n の入力文に対してこのモデルで最終的に出力される（$=v_{w_n}$ まで読みこんだときの）ベクトル A_nは文中の単語ベクトルの総和に W をかけたものと同じベクトルである。従って、このモデルは語順を考慮しない（Bag of Words）。

モデル学習の際は、これによって計算した文の分散表現と教師例の単語ベクトルからコストを計算し、確率的勾配降下法によって W を更新する。コスト関数は cosine 距離と rank loss 関数の 2 種類である。rank loss

関数とは、以下のような関数である。ある文 s と目的の単語 t のrank loss は、

$$\max(0,\ 0.1 - \cos(v_s, v_t) + \cos(v_p, v_{\text{rand}}))$$

と定義される。ただし v_{rand}は v_t 以外からランダムに選ばれた単語ベクトル、$\cos(a, b)$ は cosine 類似度関数である。

2.1.2 Recurrent Neural Network (RNN)

Recurrent Neural Network 分散表現モデル（以下 RNN 分散表現モデルと記述）は、入力文中の各単語ベクトルを順に読み込んで状態を更新し、最終的な状態を入力文の分散表現として出力するモデルである。このモデルでは、入力 s の $j(=1 \text{ or } i>1)$ 番目までの単語のベクトルを読みこんだ時の状態を A_j とすると、

$$\begin{aligned} A_1 &= \phi(W v_{w_1} + b) \\ A_{i(>1)} &= \phi(U A_{i-1} + W v_{w_i} + b) \end{aligned}$$

となる。ただし、W と U は $D \times D$ の行列で、A_j は D 次元のベクトルである。最後の v_{w_n} を読みこんだ A_nが最終的な文の分散表現となる。これは過去の状態を考慮しつつ新しい状態を計算するモデルであり、最終的な出力は語順が考慮された分散表現になることが期待できる。学習方法、コスト関数は前項と同じだが、更新する対象は W, U, b の 3 つである。

また、過去の状態に U を繰り返しかける影響で、RNN は古い入力の影響が非常に小さくなる（あるいは非常に大きくなる）という欠点がある。そこで Hill は、実装の際に Long Short Term Memory [10] を取り入れてこの問題に対処している。

2.2 テキスト分類モデル

2.2.1 Convolutional Neural Networks (CNN)

Convolutional Neural Networks テキスト分類モデル（以下 CNN 分類モデルと記述）は入力文中の各単語ベクトルから文全体の特徴ベクトルを一つ作り、softmax 関数によって分類先の確率を出すモデルである。

CNN 分類モデルにおいて、入力 s の特徴ベクトル C とカテゴリ i の確率 $p(i)$ は以下のように計算される。$v_{w_j:w_{j+h-1}}$は w_jから w_{j+h-1}までの h 個の単語の D 次元ベクトルを順に連結した $h \times D$ 次元の（縦）ベクトルで、W_iも同様に $h \times D$ 次元の（横）ベクトルである。これは任意長の入力を固定長（＝カテゴリ数）の特徴に変換し、softmax 関数によって各次元の和を 1 にすることで確率とするモデルになっている。

$$\begin{aligned} c_{i,j} &= \phi(W_i v_{w_j:w_{j+h-1}} + b_i) \\ c_i &= \max_j(c_{i.j}) \\ C &= (c_1, c_2, ..., c_k) \\ p(i) &= \frac{\exp(c_i - \max_l c_l)}{\sum_m \exp(c_m - \max_l c_l)} \end{aligned}$$

このモデルにより算出した確率と訓練データを照らし合わせ、確率的勾配降下法によって学習を行う。

また、過学習を防ぐため、学習の際は W をランダムにマスキングしている（dropout）。

3 単語ベクトルの分析

この章では、カテゴリの絞り込みが分散表現逆引き辞書モデルでの検索結果に与える影響を調査するため、word2vec によって学習した単語ベクトルとカテゴリの関係について分析する。なお、逆引き辞書という用途上、用いるカテゴリ情報としては辞書による分類データが適切と考え、本研究では WordNet から取った synset ごとの lexname[2] をカテゴリ情報として使用している。なお、1 つの synset は複数の単語からなり、各単語は複数の synset に属することがあるため、1 つの単語が複数の lexname に属する場合もある。

3.1 単語のカテゴリ分析

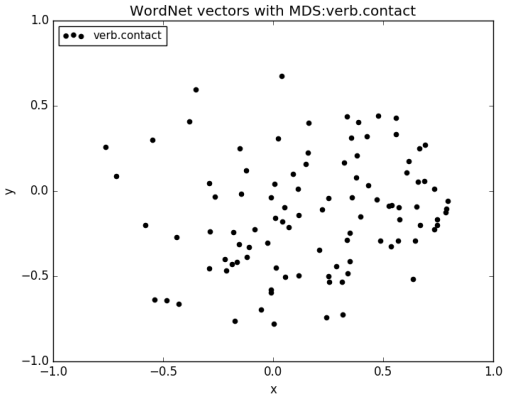

図 1: verb.contact に属する単語ベクトルの分布

本節では、word2vec 単語空間におけるカテゴリ間の関係を分析することにより、分散表現逆引き辞書モデルが推定カテゴリ情報を利用することで性能が改善で

[2] https://wordnet.princeton.edu/man/lexnames.5WN.html

きるのかを調査する。仮にカテゴリが入り組んでいたり、ある単語の周辺に異なるカテゴリの単語が多く存在していたならば、カテゴリ情報による改良の余地があると言える。

また、もしカテゴリが完全にランダムに割り当てられていれば、もちろんこれらの性質は満たされるが、ランダムなカテゴリを推定することは難しい。本節の調査が目的とするのは、少なくとも人が定義を見れば予想可能なカテゴリについて絞り込みの意味があるかを調べることである。

本分析では、まず各カテゴリに属する単語群の word2vec ベクトルの分布を Multi Dimensional Scaling (MDS) によって可視化し、カテゴリ毎の大まかな分布を視覚的に調べた。45 種の各カテゴリに属する単語を 100 個ずつ（そのカテゴリに属する単語数が 100 以下の場合は全て）サンプリングし、プールする。サンプリングを行うのは計算量を減らすためである。プールした単語群の各単語の word2vec ベクトルの cosine 距離に基づいて 2 次元 MDS を作成した。

カテゴリ毎の MDS のうち、視覚的にまとまりが見られたのは noun.animal, noun.food, noun.plant の 3 カテゴリだけで、ほとんどのカテゴリでは図 1 のようにまとまりが見られなかった。

この結果からは、word2vec 単語空間においては多くのカテゴリが互いに重なり合っており、このことからある 1 つの単語を表す word2vec 空間のベクトルからカテゴリ情報を復元するのは難しいことが予想される。従って、文を word2vec 空間で分散表現する先行研究の分散表現逆引き辞書モデルがカテゴリ情報を考慮できていない可能性がある。

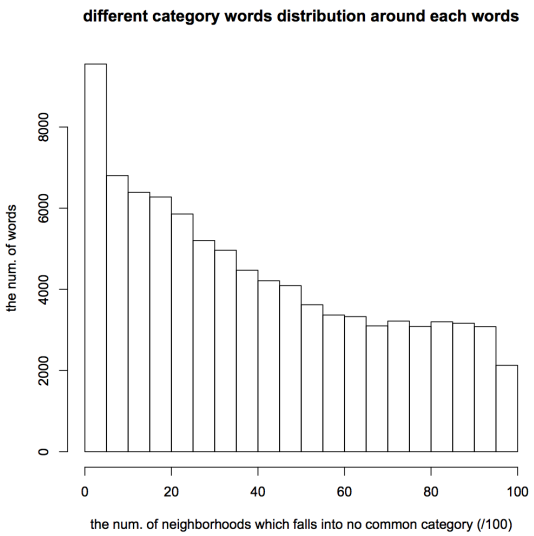

図 2: word2vec 空間において、各単語の近傍 100 単語中で中心単語と同じカテゴリに属さない単語数の分布

次に、全単語の近傍（cosine 距離が小さい）100 単語を計算し、中心の単語と同じカテゴリに属さない単語の数を数えた。図 2 がこの結果のヒストグラムである。横軸が近傍 100 単語のうち中心とカテゴリを共有していない単語の数で、そのような中心単語の数が縦軸である。これを見ると、近傍に中心と同じカテゴリに属す単語が固まっている場合も多くあるが、近傍の半分以上が全く異なるカテゴリの単語であるような単語も無視できない数ある。従って、カテゴリ情報による検索空間の絞り込みで単語検索の精度が上がることが期待できる。

4 提案モデル

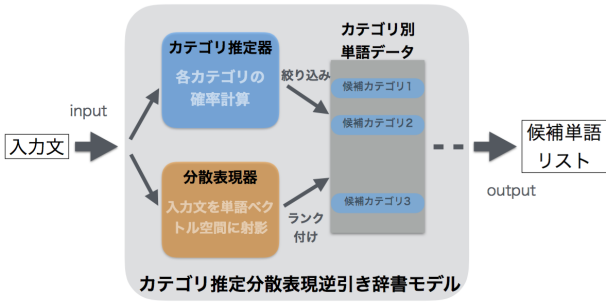

図 3: 本研究の提案するカテゴリ推定分散表現逆引き辞書モデル

本研究が提案するのは、分散表現逆引き辞書モデル [2] と CNN テキスト分類モデル [3] によるカテゴリ推定器を組み合わせたカテゴリ推定分散表現逆引き辞書モデルである。このモデルを図示したのが図 3 である。提案モデルは入力文を受け取るとその文が表す単語のカテゴリを推定し、推定したカテゴリ内の単語から入力文の分散表現を使って近いものを探すという検索システムになっている。

ここでは推定によって検索先のカテゴリをいくつに絞り込むかが問題となる。少数のカテゴリに絞り込んだ場合、絞り込み先に正解カテゴリが含まれている（＝推定に成功する）可能性は低くなるが、推定に成功した場合には目的の単語の順位が大きく向上することが期待できる。逆に多数のカテゴリに絞り込んだ場合、絞り込み先に正解カテゴリが含まれている可能性は高くなるが、推定に成功した場合の順位の向上は少数のカテゴリに絞り込んだときより劣ると考えられる。

カテゴリ推定に使用する CNN テキスト分類モデルでは、与えられた文に対してカテゴリごとにそのカテゴリに属する確率が求められるため、本研究ではこの確率情報を利用して提案モデルの推定カテゴリ数を動的に決定することとした。

ある入力文 q の target word t_q がカテゴリ $c_1, c_2, ..., c_n$ に属す確率が $p_{c_1} \geq p_{c_2} \geq ... \geq p_{c_n}$ (実験では $n = 45$) であったとする。このとき、t_q が推定確率の上位 m カテゴリのどれにも属さない確率は $\prod_{i=1}^{m}(1 - p_{c_i})$ となる。提案モデルでは $\prod_{i=1}^{m}(1 - p_{c_i}) < k$ を満たす最小の m 個の上位カテゴリを採用する。この閾値 k をモデルのパラメータとしてfailure-rate という。

モデルの詳細なプロセスは以下の通りである。

入力： 入力文（単語列）q

1. q が表す単語のカテゴリ別確率 $p_{c_1^q} \geq p_{c_2^q} \geq ... \geq p_{c_n^q}$ を計算
2. failure-rate に従って確率が上位の m カテゴリ $c_1^q, ..., c_m^q$ を決定
3. q の分散表現 v_q を計算
4. $c_1^q, ..., c_m^q$ のどれかに属す単語をその単語のベクトルと v_q との cosine 類似度順に並べ替える

出力： cosine 類似度について上位 100 単語

5 実験

本章では、分散表現逆引き辞書モデルにカテゴリの情報を与えることによる精度の変化について調べた。5.1 節では 2.1 節に基づいた文の分散表現モデルの学習と精度の評価を行い、5.2 節で仮に入力文が表す target word の属すカテゴリ内で分散表現逆引き辞書が検索を行った場合大きく精度が向上することを確認した。しかし実際は正しいカテゴリ情報は不明であり、従ってカテゴリ情報を推定する必要がある。5.3 節では 2.2 節に基づいた文のカテゴリ分類モデルの学習を行い、5.4 節では分散表現と分類についての 2 つの学習結果を併せた本研究の提案するカテゴリ推定分散表現逆引き辞書モデルの実験結果を示す。

5.1 分散表現逆引き辞書モデルの学習

この節では 2.1 節に基づいた分散表現逆引き辞書モデルの学習と評価（[2] の再現実験）について述べる。ここで用いた訓練用データの一部（word2vec ベクトルデータ）と評価用データは Hill が配布しているもの [3] を使用した。訓練データには約 370,000 語の 500 次元 word2vec 単語ベクトルと、教師例辞書データとして約 540,000 個の（単語：定義）のペアを用いた。ベクトルなどと同様、辞書データも Hill によって配布されているが、タグなどの処理が不十分であったため、配布データ中で定義されている全単語について、Wordnik API[4] からアクセス可能な全フリー辞書から定義を取り直し、本節の教師例辞書データとした。また、配布データには（クロスワードソルバとしての用途のため）より一般的な単語に対応できるよう Wikipedia のデータも含まれていたが、本データにはこれらのデータは含まれていない。(逆引き辞書の用途としては、学習に用いるのが Wikipedia を除いた辞書データのみでも大きく結果に変動はないと判断した。) なお、辞書データ中に現れる単語のうち、単語ベクトルがないものは無視した。また、学習には Hill が公開している実装 [5] を利用した。

学習したモデルの評価指標としては、実際に検索を行った場合に目的の単語が上位 1 個/10 個/100 個に入っていた割合（accuracy@1/10/100）と順位の中央値をスコアに用いた。検索空間は WordNet に定義のある全単語（約 90,000 語）である。Hill は 66,000 語の語彙リストを用いて検索空間を設定していたが、本研究は WordNet のカテゴリ情報を利用するためこのような設定とした。また、評価用のデータとしては、配布されている WordNet 辞書データ（seen/unseen）と人による記述（以下、ユーザディスクリプションと書く）の 3 種類を用いたが、そのうち検索対象の語の単語ベクトルがわかっていないものは除いた。seen は分散表現逆引き辞書モデルの訓練データに含まれるもの、unseen は含まれないものである。ベースラインとしては、入力文中のストップワード [6]（以下、sw と略すことがある）を除いた各単語のベクトルの和をその文の分散表現として採用する word2vec add モデルを用いた。

表 2 の上 3 つのブロックが評価結果である。1 つ目のブロックが学習に用いた一部の辞書データ（seen）、2 つ目のブロックが学習に用いなかった一部の辞書データ（unseen）、3 つ目がユーザディスクリプションを使った評価スコアである。[2] の論文に掲載されている結果との差として、辞書データによる評価スコアがユーザディスクリプションによる評価スコアよりも顕著に低い（Hill の研究結果によれば辞書データによる評価スコアがユーザディスクリプションによる評価スコアを上回っていた）。辞書の定義中には答えとなる単語が括弧の中に書かれている場合があるが、本研究は学習・評価の際、定義中の括弧で囲まれた部分を削除しているため、このような結果になったのではないかと考えられる。辞書の定義よりもユーザディスクリプションの検索スコアが良いという結果は、単語ベクトルは人間の書いた文書コーパスから学習されたものであるため、単語ベクトルを用いた検索は辞書の定義よりも人

[3] http://www.cl.cam.ac.uk/~fh295/dicteval.html で配布されている。

[4] http://developer.wordnik.com

[5] https://github.com/fh295/DefGen2

[6] http://www.ranks.nl/stopwords の DefaultEnglishstopwordslist を使用

間による記述であるユーザデ ィスクリプションに適しているのだと解釈できる。また、ユーザディスクリプションによる評価についても全体的なスコアが先行研究よりもやや低くなっているが、これはより検索空間が広い（およそ 1.5 倍になっている）ことと学習データが少ないことが原因の可能性がある。

モデルごとの評価としては、BOW モデルは seen と unseen のスコアに大きな差がなくユーザディスクリプションについても比較的良いスコアを記録していることから汎化性能があり、RNN モデルは seen から unseen に移ると（特に accuracy@100 と median について）大きくスコアが悪化することからやや過剰適合の傾向が見られることが言える。また、BOW モデル、RNN モデルどちらも、ユーザディスクリプションによる評価スコアは rank loss をコスト関数とした方が良い値を示している。

5.2 各単語が属す単一カテゴリ内での検索スコア（ユーザディスクリプション）

モデル	accuracy@1/10/100	median
w2v add (sw, ideal)	0.05 / 0.50 / 0.84	10
BOW cosine (ideal)	0.16 / 0.56 / 0.88	7
BOW rank (ideal)	0.17 / 0.57 / 0.89	6
RNN cosine (ideal)	0.16 / 0.46 / 0.71	13
RNN rank (ideal)	0.15 / 0.48 / 0.76	11

表 1: カテゴリ推定分散表現逆引き辞書モデルの近似理想スコア

本研究の目的は、ユーザディスクリプションを使った検索の精度がより良くなるような逆引き辞書システムを構築することである。本研究はこのための手段としてカテゴリ推定をモデルに組み込むが、本節ではユーザディスクリプションについてこのカテゴリ推定が理想的に動作した場合のスコアを近似的に調べる予備実験を行い、カテゴリ推定を組み込むという方針でどれくらいスコアが上がる見込みがあるのかを調べて提案モデルの目標値とした。

「カテゴリ推定が理想的に動作する」とは、どの入力文に対してもその文が表す単語の唯一のカテゴリを正しく推定し、そのカテゴリのみに検索空間を絞り込むということである。ここでの注意として、検索対象となる単語は複数の概念の意味を内包するため複数のカテゴリに属し得るが、検索のための記述はその中の特定のカテゴリに属する概念としての単語を探すことを試みている（と本研究では仮定している）。従って、正解となるカテゴリはどんな入力に対しても一つだけしか存在しない。

表 1 は、前節で学習した分散表現逆引き辞書モデルが、検索対象となる語が属する単一カテゴリ内で検索を行った場合のスコアを調べたものである。なお、1 つの単語が複数のカテゴリに属する場合は、どのカテゴリがその記述に適切な真の正解かを判別するのが難しいため、それぞれのカテゴリで検索した場合の順位の中央値をその単語の検索順位とした。これはカテゴリ推定が理想的に動作する場合の近似になっている。

この近似理想スコアを表 2 の 3 つ目のブロックと比較すると、いずれのモデルも単一カテゴリ推定に成功すれば大きく精度が向上することがわかる。これが提案モデルの目標スコアとなる。

5.3 カテゴリ推定器の学習

この節では 2.1 節に基づいた CNN テキスト分類モデルによる CNN カテゴリ推定器の学習について述べる。訓練データには約 370,000 語の 500 次元 word2vec 単語ベクトル（前節と同じもの）と、教師例として約 120,000 個の（カテゴリ：定義）のペアを用いた。カテゴリデータは WordNet の各 synset についての lexname と定義をペアにしたもので、前節で用いた教師例辞書データとは異なるデータである。このカテゴリは noun.animal, noun.body, verb.cognition など、45 種類ある。モデルの学習には Harvard NLP による実装 [7] を利用した。

10 分割交差検定によるモデルの評価は、訓練データにおける分類成功率（最も分類確率が高い 1 カテゴリの正解率）が平均 93%、テストデータにおける分類成功率が平均 83%という結果となった。このテストデータは訓練データと同様 WordNet から取った定義とカテゴリのペアだが、逆引き辞書に適用する際は定義ではなくユーザディスクリプションを分類する必要があるため、このテストスコアよりもやや精度が下がることが予想される。しかし現状ではユーザディスクリプションとカテゴリのペアデータを用意できていない [8] ため、ユーザディスクリプションの分類成功率は今後検証する予定である。

5.4 カテゴリ推定分散表現逆引き辞書モデルの評価と考察

この節では 5.1 節と 5.3 節で学習した 2 つのモデルを併せて、第 4 章で提案したカテゴリ推定分散表現逆引き辞書システムを作り、評価を行った。パラメータである failure-rate は、{1, 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.01,

[7] https://github.com/harvardnlp/sent-conv-torch

[8] Hill が配布しているユーザディスクリプションには検索対象語とその説明しかないので、複数のカテゴリに属する検索対象語については被験者がどのカテゴリを想定して記述したのかを確定できない。

モデル	訓練データ	評価データ	accuracy@1	accuracy@10	accuracy@100	median
w2v add (stopword)	-	辞書 (seen)	0.02	0.16	0.40	194
BOW cosine	（単語：定義）	辞書 (seen)	0.07	0.28	0.50	99
BOW rank	（単語：定義）	辞書 (seen)	0.07	0.27	0.49	105
RNN cosine	（単語：定義）	辞書 (seen)	0.10	0.28	0.50	98
RNN rank	（単語：定義）	辞書 (seen)	0.10	0.29	0.48	97
w2v add (stopword)	-	辞書 (unseen)	0.01	0.16	0.43	206
BOW cosine	（単語：定義）	辞書 (unseen)	0.04	0.25	0.48	113
BOW rank	（単語：定義）	辞書 (unseen)	0.05	0.25	0.49	104
RNN cosine	（単語：定義）	辞書 (unseen)	0.04	0.16	0.38	276
RNN rank	（単語：定義）	辞書 (unseen)	0.05	0.21	0.43	191
w2v add (stopword)	-	ユーザ	0.01	0.23	0.61	56
BOW cosine	（単語：定義）	ユーザ	0.04	0.29	0.62	44
BOW rank	（単語：定義）	ユーザ	0.07	0.32	0.61	40
RNN cosine	（単語：定義）	ユーザ	0.08	0.29	0.48	128
RNN rank	（単語：定義）	ユーザ	0.07	0.30	0.51	80
w2v add (sw) CNN	（ カテゴリ:定義）	ユーザ	0.02 (↗ 0.01)	0.32 (↑ 0.09)	**0.68** (↑ 0.07)	**30** (↓ 26)
BOW cosine CNN	（単語：定義）＋（カテゴリ：定義）	ユーザ	0.06 (↑ 0.02)	0.35 (↑ 0.06)	0.67 (↑ 0.05)	32 (↓ 12)
BOW rank CNN	（単語：定義）＋（カテゴリ：定義）	ユーザ	0.09 (↑ 0.02)	**0.36** (↑ 0.04)	0.65 (↑ 0.04)	**30** (↓ 10)
RNN cosine CNN	（単語：定義）＋（カテゴリ：定義）	ユーザ	**0.10** (↑ 0.02)	0.29 (→ 0)	0.49 (↗ 0.01)	108 (↓ 20)
RNN rank CNN	（単語：定義）＋（カテゴリ：定義）	ユーザ	0.08 (↗ 0.01)	0.33 (↑ 0.03)	0.52 (↗ 0.01)	86 (↑ 6)

表 2: 学習モデル・データ別評価スコアまとめ（ユーザ=ユーザディスクリプション。提案モデルのスコアには CNN カテゴリ推定を加えたことにより元のモデルのスコアから上昇/下降した値を矢印付きで示した。）

0.005, 0.001} を候補値として、分類器の学習に用いなかった WordNet のデータで実際に検索した際に最も accuracy@1/10/100 の平均値が高かったものを採用した。具体的な値は、全モデルについて failure-rate= 0.01 である。

表 2 の最下ブロック（...CNN というモデル）がユーザディスクリプションを用いた提案モデルの評価結果である。表の 3 つ目のブロックにある CNN カテゴリ推定なしの分散表現逆引き辞書モデルのスコアと比較して、いずれのモデルもほとんどのスコアが向上している（スコアの後ろの括弧内に上昇値を記載している）。分散表現逆引き辞書モデルのベースラインであった word2vec add モデルが CNN カテゴリ推定と最も親和性が高く (=スコアの上昇幅が大きい)、特に accuracy@100 については最も高い値を記録しているのが興味深い。しかし、実際に逆引き辞書を使用する際には高順位の候補しか見ない場合も多くあると考えられ、その場合に重要なのは accuracy@1/10 である。この観点では、RNN cosine CNN と BOW rank CNN の逆引き辞書モデルが良い性能を示した。特に、BOW rank CNN は accuracy@1/10 を含めて全体的にスコアが高くなっている。

この結果から、辞書データから学習したカテゴリ分類器により推測したカテゴリ情報を加えることで、分散表現逆引き辞書モデルの精度を改良できることが示された。このカテゴリ推定機能は Node-Graph Architecture モデル [9] のような異なる検索機構にも応用可能であり、精度改善が期待できる。

本結果の問題点として、表 1 の理想スコアに大きく迫れていないことがある。これはユーザディスクリプションのカテゴリ推定精度があまり高くないことに起因する。failure-rate=0.01 の際の絞り込み先カテゴリ数の平均は 8.3 個で、絞り込み先のどのカテゴリにも目的の単語がなかった場合は全 200 個記述中 9 個であった。カテゴリ推定の精度を上げ、絞り込みカテゴリ数を減らしつつ絞り込み失敗数を抑えることでより理想スコアに近づくことが今後の課題である。本研究が採

用した CNN によるカテゴリ推定手法は文構造を無視しているため、文法情報を組み込んだ分類手法を適用できればより良い推定ができる可能性がある。

6 結論

本研究は、先行研究による word2vec 空間での文の分散表現が、その文が表す単語のカテゴリの情報を利用できていない可能性を指摘し、仮に正しいカテゴリの情報がわかっていれば、分散表現逆引き辞書モデルの検索精度が大きく向上することを確認した。そこで著者らは、CNN によって文が表す単語のカテゴリを推定する手法と、それを用いたカテゴリ推定分散表現逆引き辞書モデルを提案し、検索精度が向上することを確認した。提案手法はカテゴリの絞り込みを行わない任意の逆引き辞書モデルに応用可能であり、汎用性が高いと言える。

前章の考察で挙げたように、カテゴリ推定の性能が十分でないため、逆引き辞書全体としてのスコア向上が若干にとどまっていることが本研究の問題点の 1 つである。さらに、本研究は WordNet の語彙とカテゴリを前提としているため、WordNet にない単語は検索できない。また、WordNet の lexname 以外により適切な (=推定が簡単かつ word2vec 空間中に適度な広がりを持つ) 単語のカテゴリがある可能性もある。

本研究の展望として、主に 2 通りの方針が考えられる。1 つ目は、文構造を利用したカテゴリ推定、インタラクティブな検索モデルの実装などによって検索空間絞り込みの精度を上げることで逆引き辞書としての検索性能を向上させる方針である。インタラクティブな検索モデルとは、カテゴリ推定がうまくいきそうにない場合 (絞り込み先カテゴリ数が一定値以上の場合など) にはユーザに単語のカテゴリを尋ねるようなもので、絞り込みの精度を補うことができる。この場合はユーザの入力文と正解カテゴリのセットが情報として得られるため、それを用いた再学習によって更にユーザディスクリプションに適した分類器が構成できると期待されるが、大規模な被験者実験が必要となる。2 つ目は、提案モデルの逆引き辞書以外の用途への応用である。例えば、Wikipedia のデータを学習データに取り入れ、Wikipedia 内のカテゴリと WordNet のカテゴリを統合することで新たなカテゴリを作ることができれば、[2] で応用されていたクロスワードパズルのソルバとしての活用が可能になる。

謝辞

本研究の一部は、学際大規模情報基盤共同利用・共同研究拠点、および、革新的ハイパフォーマンス・コンピューティング・インフラの支援による。

参考文献

[1] Mikolov, Tomas., et al.: Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pp. 3111–3119 (2013)

[2] Hill, Felix., et al.: Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 17–30 (2016)

[3] Kim, Yoon.: Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751 (2014)

[4] Mikolov, Tomas., et al.: Recurrent neural network based language model. *Interspeech*, Vol. 2, p. 3 (2010)

[5] George, A., Miller.: WordNet: A Lexical Database for English. *Communications of the ACM* Vol. 38, No. 11, pp. 39–41 (1995)

[6] Arora, Sanjeev., et al. : A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, Vol. 4 pp. 385–399 (2016)

[7] Ferrucci, David., et al.: Building Watson: An overview of the DeepQA project. *AI magazine*, Vol. 31(3), pp. 59–79 (2010)

[8] Schwartz, Bennett L., and Janet Metcalfe. : Tip-of-the-tongue (TOT) states: Retrieval, behavior, and experience. *Memory & Cognition*, Vol. 39(5) pp. 737–749 (2011)

[9] Thorat, Sushrut., Choudhari, Varad.: Implementing a Reverse Dictionary, based on word definitions, using a Node-Graph Architecture. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*: Technical Papers, pp. 2797–2806 (2016)

[10] Hochreiter, Sepp., Jürgen, Schmidhuber.: Long short-term memory. *Neural computation*, 9(8), pp. 1735–1780. (1997)