# 令和５年度大阪大学未来基金「学部学生による自主研究奨励事業」研究成果報告書

| | | | | | |
|---|---|---|---|---|---|
| ふりがな<br>氏名 | ゆだ　なつき<br>湯田　夏喜 | 学部<br>学科 | 基礎工学部<br>システム科学科 | 学年 | ３年 |
| ふりがな<br>共同<br>研究者氏名 | | 学部<br>学科 | | 学年 | 年 |
| | | | | | 年 |
| | | | | | 年 |

| | | | |
|---|---|---|---|
| アドバイザー教員<br>氏名 | 松原　崇 | 所属 | 基礎工学研究科 |

| | |
|---|---|
| 研究課題名 | 自己教師あり学習モデルにおける論理学的概念の分析 |
| 研究成果の概要 | 研究目的、研究計画、研究方法、研究経過、研究成果等について記述すること。必要に応じて用紙を追加してもよい。（先行する研究を引用する場合は、「阪大生のためのアカデミックライティング入門」に従い、盗作剽窃にならないように引用部分を明示し文末に参考文献リストをつけること。） |

## 1　はじめに

大規模言語モデル(Large language model, LLM)は数多くの自然言語処理タスクにおいて驚くべき性能を発揮し，情報検索，対話システム，文章生成などにおいて人間に匹敵するかそれ以上の結果を示している．特に推論を行うタスクとして自然言語推論(Natural language inference, NLI)があり，このタスクでは与えられた前提(premise, P)と仮説(hypothesis, H)に対して，前提が仮説を含意しているかどうかを検証するものである．知的な推論は LLM に期待される能力の 1 つであるが，LLM が文章を正確に理解し推論する一般化された能力を獲得できているかは不透明である．

NLI は文レベルでの含意の判定を行うタスクが多いが，単語レベルでの意味の包含関係に注目した研究もある[11][10][2]．実用的には文レベルでの意味関係が重視されるが，知的な推論において正確性の高いパフォーマンスを発揮するためには単語レベルでの意味的関係の理解が重要であると考える．また，意味関係には推移関係が成り立ち，階層性が存在している．単語 A, B, C について意味的に A⊇B, B ⊇C が成り立っているとき，A⊇B と A ⊇C の両方の関係が成立していると判定できることが言語モデルには求められる．この 2 つの関係について同じように判定できるようにすることに興味がある．

そこで単語の意味的な含意関係を言語モデルが正確に把握しているか，またどのような様態で把握しているかを調べるため，WordNet の名詞の階層関係を用いることにした．WordNet の階層性を反映させたデータセットを作成し，データセットの各文の埋め込みを入力とした多変量解析手法によって包含関係を分析した．結果，単語レベルの意味的包含関係は単語レベルの埋め込み空間が意味の類似性の観点で学習されていることが必要でありうること，埋め込みの差のノルムが重要であることが分かった．

## 2　実験準備

単語間の包含関係を調べるために，単語のペア$(a_i, b_i)$のうち意味的に$a_i$ が$b_i$ を含意するペアを正例，含意していないペアを負例に分類させる二値分類タスクを行う。そのための準備を以下に説明す

る。

2.1　WordNet の階層関係の抽出

階層性を反映したデータセットの作成のために WordNet[1]から木構造を抽出した. WordNet の名詞は同義語の集合を表す synset によって抽象的な概念に対応するように分類され，synset は階層構造でリンクされている．ある synset X はそれよりも具体的な意味を持つ 1 つ以上の synset Y, Z, …を X の下位語(hyponym) として保持している．逆に，X は Y や Z の上位語 (hypernym)になる．このとき，ある synset の hypernym は複数持ちうるので，WordNet の階層構造は木構造ではなく，閉路も持つ．簡単のために抽出する階層構造は木構造であることが望ましいので，抽出された木が最小全域木(Minimum Spanning Tree, MST) となるように抽出した．このため，抽出した木構造の一部には一般的ではない包含関係が含まれたり，逆に一般的な包含関係が抽出されていない場合がある．

WordNet からは根を”animal”と”vehicle”とする 2 つの木を抽出した．また，それぞれの木の葉は適当と思われる単語のリストを用いて決定した [2][3]．木構造の頂点の数はそれぞれ 346，78 であり 合計で 424 個の単語が抽出された．

2.2　データセットの作成

データセット$\mathcal{D}$は単語の ペア$(a_i, b_i)$ $i = 1, \ldots, n$ の集合であり，正例は抽出した木から，負例は含意関係の無いペアの中から得られ，1 つのデータセットにおいて正例と負例の数は等しくする．ただし，データセットに含まれるペアにはいくつかの条件を設けた．

まず，正例において偏りが生まれないために”animal” のような上位語になりやすい単語は 5 つまでしか正例に含まれない．次に，負例の単語の分布は正例の単語の分布に従うようになっている．これは下位語に葉の単語が来れば正例の可能性が高くなり，上位語に来れば負例の可能性が高くなることで埋め込みを調べなくてもある程度分類できてしまうことを防ぐためである．また，正例には含意関係の向きがあり，含意関係の方向を区別するタスクではないので，負例において$(a_i, b_i) \in \mathcal{D}$ であれば$(b_i, a_i) \notin \mathcal{D}$ である．最後に，負例は異なるドメインをまたがるペアが含まれない．正例には異なるドメインにまたがるペアは含まれていないためである

2.3　モデルへの入力

データセット$\mathcal{D} = (A, B)$ に含まれる単語ペア$(a_i, b_i)$の単語は文に組み込まれて文のペア$(\hat{a}_i, \hat{b}_i)$になる．この際に作成した文は抽出されたすべての単語で成り立つような文とし，実験ではすべて”This is a picture of a(n) [単語].”とした．言語モデルへの入力は$n$ 個の文のペア$(\hat{a}_i, \hat{b}_i)$であり，出力として埋め込み$(\tilde{a}_i, \tilde{b}_i)$，$\tilde{a}_i \in \mathbb{R}^d$，$\tilde{b}_i \in \mathbb{R}^d$が得られる．ただし，モデルが出力する埋め込みの次元を$d$ としている．二値分類タスクは$p$ 次元ベクトル $\tilde{c}_i$を正則化ロジスティック回帰(Logistic regression, LR)あるいはサポートベクターマシーン(support-vector machine, SVM)に入力することによって行われる．ここで$\tilde{c}_i$は$(\tilde{a}_i, \tilde{b}_i)$ に対して $\tilde{c}_i = P(\tilde{a}_i, \tilde{b}_i)$ と変換したベクトルであり，$p$は変換処理 P に依存する．実験で用いた$P$ は表 1 にまとめられている．

[1] バージョンは 3.1
[2] https://en.wikipedia.org/wiki/List_of_animal_names
[3] https://en.wikipedia.org/wiki/Outline_of_vehicles

**表 1 変換処理の一覧**

| $P$ | 操作 | $\tilde{c}_i$ | dim |
|---|---|---|---|
| $P_{none}$ | ベクトルの結合 | $\tilde{c}_i = [\tilde{a}_i, \tilde{b}_i]$ | $2d$ |
| $P_{diff}$ | 要素ごとの差 | $\tilde{c}_i = \tilde{a}_i - \tilde{b}_i$ | $d$ |
| $P_{multiple}$ | 要素ごとの積 | $\tilde{c}_i = \tilde{a}_i \otimes \tilde{b}_i$ | $d$ |
| $P_{std}$ | ベクトルごとに標準化 | $\tilde{c}_i = [std(\tilde{a}_i), std(\tilde{b}_i)]$ | $2d$ |
| $P_{std+diff}$ | 標準化後に差 | $\tilde{c}_i = std(\tilde{a}_i) - std(\tilde{b}_i)$ | $d$ |

### 2.4 モデル

実験では自然減のモデルとして事前学習済みモデルを用い，モデルの埋め込みを分析する．最初の 2 つのモデルは静的埋め込みを出力する Skip-gram[5]，fastText[4] であり，残りは Transformer をもとにしたモデルである BERT(Bidirectional Encoder Representations from Transformers [1])，GPT2(Generative Pre-trained Transformer 2 [8])，SBERT(Sentence-BERT [9])，babbage-001(cpt-text M，以下 babbage と呼ぶ．[6])である．Skip-gram と fastText は Genism の'word2vec-google-news-300'と'fasttext-wiki-news-subwords-300' を用い，BERT，GPT2，SBERT は Hugging Face の BERT-large，GPT2-medium，GPT2-large，'sentence-transformers/all-mpnet-base- v2' を用いた．babbage は OpenAI 社の API から'text-embedding-babbage-001' のモデルの埋め込み を取得した．babbage は GPT-3 のパラメータで初期化したモデルを対照学習して得られたモデルである．

## 3 実験

### 3.1 処理 P による違い

各モデルが出力した埋め込みにして処理$P$ を行い，ロジスティックス回帰(LR) とサポートベクターマシーン(SVM)で 2 値分類タスクを行った．分類タスクでは訓練セットの割合を 50%，セットの割合を 25%，テストセットを 25%としている．表 2 がその結果である．比較のために$n$ 次単位行列$I$ の各行を埋め込みの代わりとして用いた結果も載せている．

**表 2 処理 P，解析手法とモデルごとの正答率**

| | | $I$ | BERT | SBERT | $\text{GPT2}_{\text{medium}}$ | $\text{GPT2}_{\text{large}}$ | Skip-gram | fastText | babbage |
|---|---|---|---|---|---|---|---|---|---|
| $P_{none}$ | LR | 0.584 | 0.629 | 0.631 | 0.631 | 0.648 | 0.621 | 0.643 | 0.623 |
| | SVM | 0.564 | 0.681 | 0.760 | 0.522 | 0.667 | 0.696 | 0.737 | 0.732 |
| $P_{diff}$ | LR | 0.646 | 0.662 | 0.690 | 0.682 | 0.707 | 0.665 | 0.681 | 0.715 |
| | SVM | 0.575 | 0.724 | 0.833 | 0.629 | 0.720 | 0.805 | 0.849 | **0.879** |
| $P_{multiple}$ | LR | - | 0.631 | 0.741 | 0.484 | 0.581 | 0.636 | 0.732 | 0.732 |
| | SVM | - | 0.559 | 0.769 | 0.486 | 0.533 | 0.678 | 0.749 | 0.757 |
| $P_{std}$ | LR | - | 0.662 | 0.653 | 0.614 | 0.668 | 0.650 | 0.653 | 0.654 |
| | SVM | - | 0.679 | 0.760 | 0.511 | 0.660 | 0.709 | 0.738 | 0.732 |
| $P_{std+diff}$ | LR | - | 0.664 | 0.698 | 0.674 | 0.702 | 0.685 | 0.676 | 0.693 |
| | SVM | - | 0.748 | 0.833 | 0.654 | 0.726 | 0.804 | 0.854 | 0.879 |

最も正答率が高かったのは babbage の埋め込みを$P_{diff}$ で差をとって SVM で分類をした時であり，87.9%の正答率だった．GPT2medium 以外のモデルは$P_{diff}$あるいは$P_{std+diff}$ を 埋め込みに施した後，SVM で分類をする組み合わせが最も良い結果をもたらしていた．LR では正答率が 75%に達する

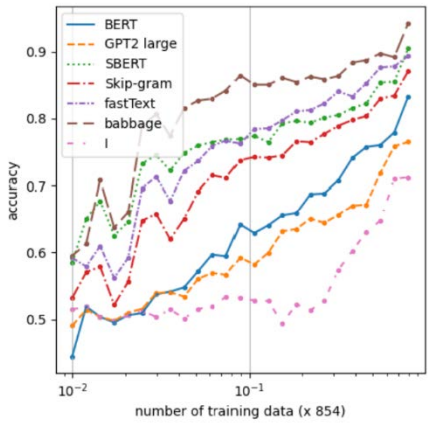


図 1　訓練データの割合による正答率の変化

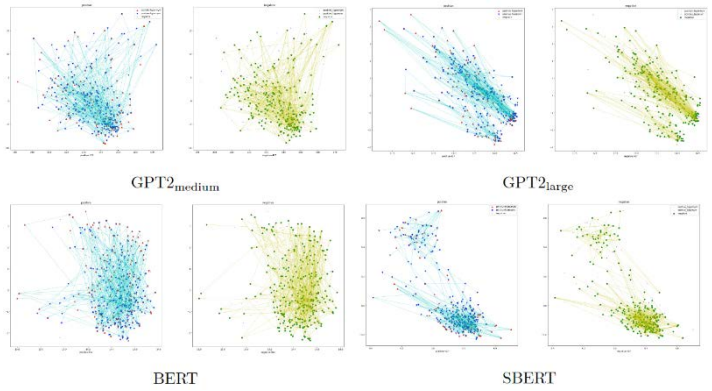


図 2 埋め込みを LSA で可視化した図．各図左は正例に含まれる$(\tilde{a}_i, \tilde{b}_i)$を結ぶ線が示され、各図右は同じく負例に含まれる線が示されている。

ことが無く, GPT2 系以外のモデルでは SVM の方が良い結果だった. $P_{none}$ と $P_{std}, P_{diff}$ と $P_{std+diff}$ を比較すると, 標準化によってわずかによくなるケースも多いが, ほぼ大差はなかった. また, $P_{multiple}$ では正答率が 50%前後になりベクトルが意味を成さなくなっているものもあれば, $P_{none}$ よりも良くなっている組み合わせもあった.

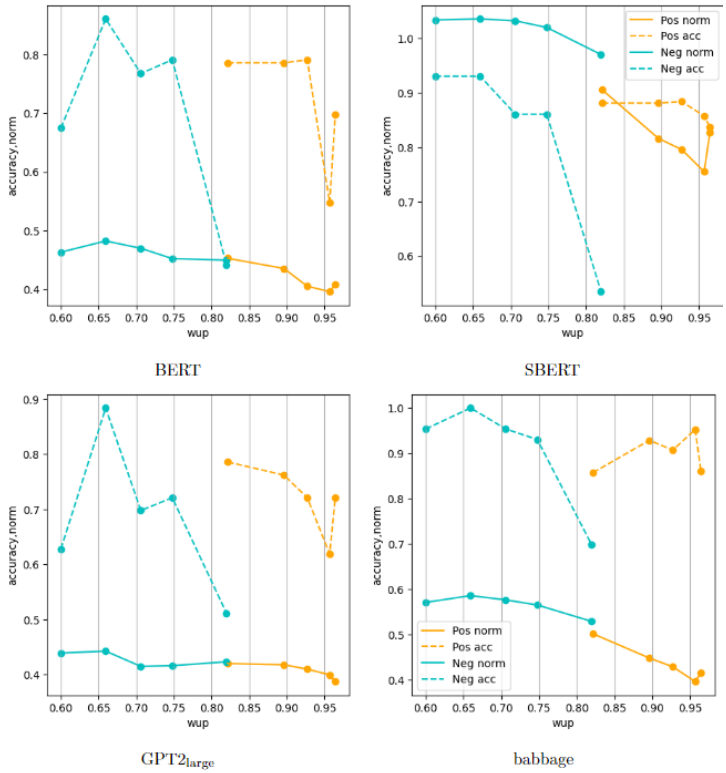


図 2 $d_{wup}$に対する単語ベクトルの差のノルムと正答率の変化．グラフ上の各点はそれぞれ$d_{wup}$の下位 0~20%,20~40%,…,80~100%の正答率とノルムの平均値を示す。

3.2　訓練データを変化させたときの正答率

それぞれのモデルで最も高い正答率を出せる$P$ と分類法の組み合わせに対して, 訓練ペアの割合を変化させたときの正答率の変化を調べた．結果は図 1 である．babbage は訓練データが少なくても正答率が高いが，BERT と GPT2large では訓練データ数の対数に比例して正解率が上昇している．ま

た，$I$を埋め込みの代わりに用いた場合でも訓練データの割合 20%を超えたあたりから正答率が増加していくことが分かる．

### 3.3 埋め込みの分布

モデルから得られる埋め込み$(\tilde{a}_i, \tilde{b}_i)_n$を潜在的意味解析(Latent Semantic Analysis, LSA)によって 2 次元に次元削減して可視化した．図 2 はそれぞれ GPT2medium，GPT2large，BERT，SBERT の埋め込みを図示したものである．以前に述べたように作成したデータセットには”animal” と”vehicle” の 2 つのドメインのいずれかに属する単語で構成されており，SBERT の埋め込みは 2 つのクラスターにはっきりと分かれている．GPT2large の埋め込みもわずかに 2 つのクラスターに分かれているように見えるが，GPT2medium と BERT の埋め込みはそのようには見えない．他の Skip-gram，fastText，babbage の埋め込みの分布は省略しているが SBERT のように 2 つのクラスターに分かれている様子がはっきりと確認できる．

### 3.4 意味上の距離とノルム

$P_{diff}$を埋め込みに施した後 SVM で分類をする組み合わせが 2 値分類に有効なこと，babbage の埋め込み$(\tilde{a}_i, \tilde{b}_i)_n$,に対して$P_{diff}$ で差をとったベクトルを 3 次元空間で可視化すると中心部分に正例が集まり，周辺に負例が分布する様子が見られることから，ベクトルの差分のノルムが分類に影響を与えている可能性を考えた．そこで Wordnet 上で定義される単語ペア$(a_i, b_i)$間の距離$d_{wup}$ に対して埋め込みの差$c_i = P_{diff}(\tilde{a}_i, \tilde{b}_i)$のノルム$d_{norm}$，これと $(a_i, b_i)$に対する正答率がどのように変化するのかを調べた．$d_{wup}$は 2 つの synset の Wu-Palmer 類似度であり，分類における 2 つの単語の語義の深さとそれらの最も具体的な祖先ノードの深さに基づく類似度で$0 \leq d_{wup} \leq 1$となる．ノルムは 2-ノルムである．

図 3 は正例と負例に含まれる埋め込みの差のノルムを比較していたが，LR や SVM の分類結果と埋め込みそれぞれに対しても同じような傾向がある．含意関係があると分類された場合に E，無いと分類された場合に NE と表記することにすると，正例，負例，E，NE の ペアの集合のノルムの平均値は表 3 のようになる．skip-gram，fastText 以外の埋め込みは標準化されている．結果から GPT2medium 以外の E と N E の埋め込みの差のノルムは正例，負例のノルムとそれぞれ近い数字をとっており，これらの埋め込みの分類結果はデータセットの埋め込みの差のノルムと対応があることが分かる．

**表 3 正例，負例，E，NE に含まれる埋め込みの差のノルム**

| | BERT-large | SBERT | $GPT2_{medium}$ | $GPT2_{large}$ | skip-gram | fastText | babbage-001 |
|---|---|---|---|---|---|---|---|
| 正例 | 0.419 | 0.820 | 0.034 | 0.407 | 0.517 | 0.472 | 0.438 |
| 負例 | 0.463 | 1.020 | 0.032 | 0.427 | 0.604 | 0.560 | 0.565 |
| E | 0.419 | 0.818 | 0.031 | 0.395 | 0.514 | 0.473 | 0.431 |
| NE | 0.465 | 1.030 | 0.035 | 0.441 | 0.611 | 0.563 | 0.571 |

## 4 考察

### 4.1 含意関係における埋め込み分布の重要性

少ない訓練データでも高い正答率を出す babbage のようなモデルの埋め込みは単語ペア間の含 意関係の有無について一般化された情報を保持していることが考えられる．一方，訓練データの数の対

数に比例して正答率が上昇して行くような BERT，GPT2 モデルはそのような情報を保持して いることが期待できない．Skip-gram や fastText，SBERT の埋め込みも babbage モデルよりは劣るが少ない訓練データでもある程度のパフォーマンスをしており，babbage モデルを含めた 4 つのモデルの共通点は図で見たように 2 次元上に図示した際にドメインがクラスターになって区別が付けられる点がある．意味の近い単語同士を近づけ，意味の遠い単語を話すような分布である埋め込みが含意関係を識別する上でも適していると考えられる．

BERT の埋め込みが意味的に不十分であることは[9]で述べられており，具体的には BERT の埋め込みを直接用いると Glove[7]の埋め込みよりも文の意味を捉えられていないことが示されている．また，[3]では BERT の埋め込みが持つ欠点を示し，埋め込みを等方的なガウス分布に変換することで解決することを提案している．今回の実験の結果は BERT の意味的な性能の不十分さと SBERT や babbage のような表現学習を行ったモデルが単語の意味を捉える点で優れていることを 再確認することになった．

### 4.2 ノルムと含意関係

図 3 や表 3 の結果から，埋め込みの差$c_i = P_{diff}(\tilde{a}_i, \tilde{b}_i)$のノルムの大きさは含意関係の有無を区別する方法の 1 つであることが考えられる．babbage や SBERT の埋め込みは意味的な単語間の距離を埋め込みの差のノルムとして保持しており，ベクトル空間上で近くなりがちなデータセットの正例と遠くなりがちな負例を区別する上で有効な特徴量となっているだろう．正例と負例のノルムの大きさが近づく$d_{wup}$= 0.80 付近で分類の正答率が下がることや，SVM による分類が有効であることがこのことを支持している．

ただし，ノルムが含意関係の有無の分類に有効である可能性が高いことは今回作成したデータセットにしか言えない．なぜなら今回作成したデータセットは正例と負例の間で意味的距離が平均的に十分離れており，意味的距離(今回は$d_{wup}$)が遠い単語ペアほど差のノルムが大きくなるなどノルムに意味的距離が反映されることが自然だからである．意味的距離あるいは差のノルムのどちらか，あるいは両方を正例と負例の間でそろえたデータセットを作成し実験をしなければ一般的なノルムと含意関係の有無の関係性ははっきりしないだろう．

### 4.3 分類を誤った単語ペアの分析

図 4 は最も高い正答率になる babbage モデルの埋め込みの差をとって SVM に入力し訓練セット の割合を 50%として分類した際，分類を誤ったペアを示している．また，各点は各単語のベクトル$\tilde{a}_i$を主成分分析をして表示したものである．検証セットによるハイパーパラメータのチューニングを行っていないので表 1 の結果とは異なり，正答率は 0.90，混同行列は$\begin{bmatrix} 190 & 20 \\ 21 & 191 \end{bmatrix}$である．図中の紫の矢印は誤って含意関係がないと分類したペア(False Negative, FN)を上位語から下位語の方向にさしており，緑の矢印は誤って含意関係があると分類したペア(Falese True, FT)である．緑のペアの矢印の方向に意味はない．図 3 からは $d_{wup}$ が大きい負例の正答率が低くなっていることが分かる．実際に図 4 の短い緑の矢印を見てみると，例えば(shorebird(渉禽類),pelagic bird(外洋にすむ鳥)) でどちらも aquatic bird(水鳥) だが含意関係はない．$d_{wup}$ (shorebird, pelagic bird) = 0.846 である．他にも $d_{wup}$ (cetacean(鯨類),dugong(ジュゴン)) = 0.889(共通項は aquatic mammal) や $d_{wup}$ (heavier-than-air craft(重航空機，飛行には推力が必要),airship(飛行船)) = 0.870(共通項は air-craft) などがあり，人間でも分類を間違いうるペアが多い．

一方長い紫の線に注目すると，$d_{wup}$（wheeled vehicle, amphibian(両生類，水陸両用車)）= 0.8, $d_{wup}$（lizard, monitor(オオトカゲ)）= 0.962963 など文脈が不足していて間違えているケー スがある．これらのケースは文脈を指定する文があれば正解する可能性が高く なると考えられる．一方紫の短い矢印もいくつもあり，$d_{wup}$（primate, homo）= 0.923 や$d_{wup}$（cavy(ハツカネズミ),guinea pig(モルモット)）= 0.963 は含意関係を捉えられていない．$d_{wup}$が大きいペアを正確に分類することが単語の含意関係を区別することにつながると考えられるが，$d_{wup}$ の大きい含意関係がないペアは少なく実験を行うためには語数を増やす必要がある． また，正確な単語レベルの含意関係の区別には単語の意味を特定するために文脈をつける必要もある．

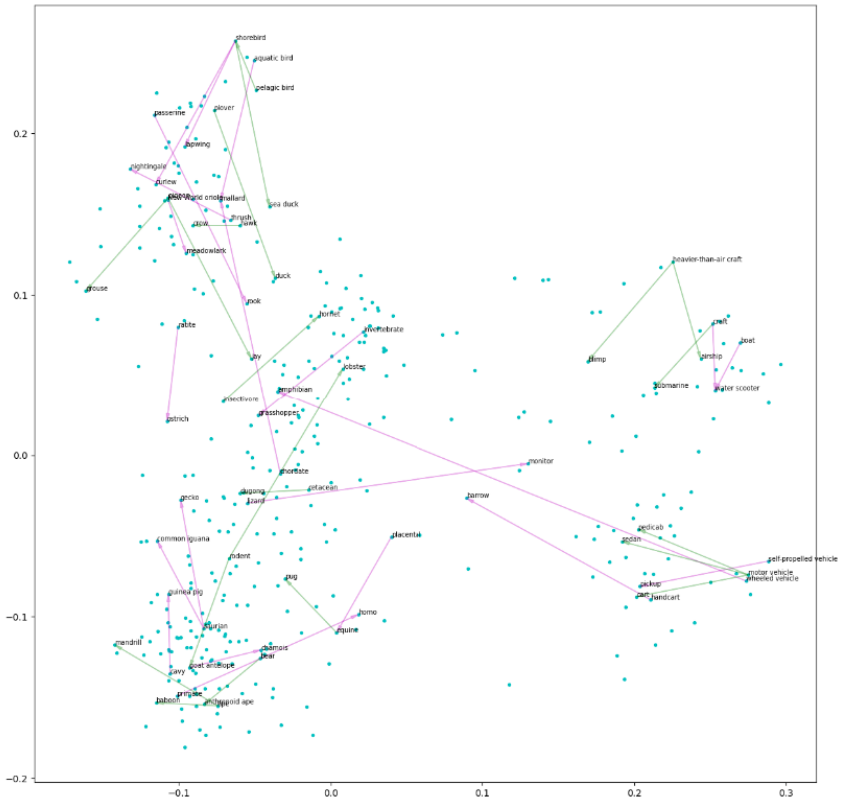


**図 4 babbage の埋め込みを用いて分類を誤ったペア**

## 5　まとめ

今回の研究では単語の意味的な含意関係を言語モデルが正確に把握しているか，またどのような様態で把握しているかを調べた．WordNet から単語の含意関係を階層関係を保持したまま抽出し，各言語モデルに入力して埋め込みを得た後，処理$P$ を加えてロジスティックス回帰とサポートベクターマシーンで分類した．結果，単語の意味的包含関係はモデルの埋め込みの分布が意味の類似性の観点で適切に分布している必要がありうること，埋め込みの差によって 9 割近くのペアの包含関係の有無は分類できることが分かった．しかし，意味的に近い単語ペアの包含関係を分類することは難しく，含意関係は何を基準にすれば正確に分類できるか知るためには実験手法にさらに改良を加える必要があることも分かった．

## 6　謝辞

本研究を進めるにあたり，ご指導をいただいたアドバイザー教員の松原崇先生に心よりお礼申し上げます．

参考文献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, Vol. abs/1810.04805, , 2018. [2] Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 163-173, Online, November 2020. Association for Computational Linguistics.

[3] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In Proceedings of the 2020 Conference on Em- pirical Methods in Natural Language Processing (EMNLP), pp. 9119-9130, Online, November 2020. Association for Computational Linguistics.

[4] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In Proceedings of the Eleventh In- ternational Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed represen- tations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, Advances in Neural Information Processing Sys- tems, Vol. 26. Curran Associates, Inc., 2013.

[6] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, et al. Text and code embeddings by contrastive pre-training. ArXiv, Vol. abs/2201.10005, , 2022.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532-1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, Vol. 1, No. 8, p. 9, 2019. [9] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert- networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 11 2019.

[10] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. Do neural models learn systematicity of monotonicity inference in natural language? In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6105-6117, Online, July 2020. Association for Computational Linguistics.

[11] Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzian-idze, and Johan Bos. Can neural networks understand monotonicity reasoning? In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 31-40, Florence, Italy, August 2019. Association for Computational Linguistics.