

英文穴埋め問題における文章ベクトルと 学習データの質の影響

森 洸樹^{1,a)} 三輪 誠^{1,b)} 佐々木 裕^{1,c)}

概要：語順の情報を組み込んだニューラル言語モデルは TOEIC の英文穴埋め問題において高い正答率を出すことが報告されている。この手法では文脈外の情報を利用できないという問題がある。また、学習に用いている Wikipedia のデータは、文法的な誤りがあるなど、質の良いデータではない可能性がある。これらの問題に対応するために、文章ベクトルによる文脈外の情報の利用と BNC のデータによる学習データの質の影響について調査したので、これらの結果について報告する。

1. 序論

英文穴埋め問題は言語モデルを評価する方法として広く利用されている。英文穴埋め問題の代表的なタスクとして、MSR Sentence Completion Challenge (MSR) [7] と呼ばれる Holmes の文章を対象にした穴埋め問題が提供されており、このタスクでは、単語の共起に関する学習を行っているニューラル言語モデル [2] である ivLBL [6] が高い正答率を出している。また、図 1 のような TOEIC やセンター試験などで用いられるの英文穴埋め問題では文法や熟語など語順が影響する問題が多く出題されているが、ivLBL やそれに似たモデルである vLBL では語順を考慮していないモデルなので、このような問題を解くことは難しい。例えば、図 1 では、冠詞”the”と名詞句”weather conditions”の間に入る品詞は形容詞であるため、d) の”predictable”が正解となる。しかし、vLBL や ivLBL は語順を考慮せず、選択肢の中から空欄前後の数単語と共起しやすいものを解答として選ぶ。おそらく”to”や”weather”の共起しやすい c) の”predict”と誤答してしまう。そこで、我々はこのような問題に対して、単語の位置情報を用いるモデルと vLBL と組み合わせることで、語順と共起両方を考慮した vLBL+vLBL(c) を提案し TOEIC の問題において高い正答率を達成したことを報告した [8]。

文献 [6] や [8] などで利用されているように、ニューラル言語モデルの学習には学習データとして Wikipedia のデータがよく利用されている。これらの理由として、ニュー

The BBQ will be held in May due to the [] weather conditions .

- a) predictably
- b) prediction
- c) predict
- d) predictable

図 1 英文穴埋め問題の一例

ラル言語モデルの学習に用いる学習データには語句が十分な頻度で出現するテキストが必要であることがある。Wikipedia によるテキストデータは非常に大きなものであり、ニューラル言語モデルの学習には適しているといえる。一方、誰にでも編集ができるという点から、文法や語彙に誤りがあることがある。文法や語彙の誤りはノイズとなり学習に悪影響を及ぼし、効率のよい学習が行うことができないと考えられる。そのため Wikipedia のデータは質のよいデータとはいえない可能性がある。特に TOEIC やセンター試験などの文法や語彙に関する問題が多く出題される英文穴埋め問題では、このような質の悪さによる影響が大きいと推測できる。

英文穴埋め問題では解くために空欄から離れた場所にある単語に注目しなければならない場合がある。しかし、ivLBL や vLBL+vLBL(c) では定めた文脈の領域内の情報しか利用していないので、このような問題を解くことは難しい。そこで、PV-DM [4] と呼ばれる、可変長の文や段落など単語よりも大きな文章という単位を固定長のベクトルで表現し、その文章ベクトルを各単語ベクトルとともに学習を行う研究に注目した。文章ベクトルは、その文章に書かれている内容を記憶する機能を持っており、ivLBL や vLBL+vLBL(c) では扱っていなかった文章の内容、つまり定めた文脈の外に書かれている内容を学習に利用でき

¹ 豊田工業大学
Toyota Technological Institute
a) sd15435@toyota-ti.ac.jp
b) makoto-miwa@toyota-ti.ac.jp
c) yutaka.sasaki@toyota-ti.ac.jp

ていると考えられる。そのため、英文穴埋め問題において問題文を一つの文章と捉え、文章ベクトルを生成することで、空欄から離れた場所にある単語に関する情報を利用することで、正答率の向上につながるのではないかと考えられる。

本研究では、図1のようなTOEICの英文穴埋め問題の正答率向上を目的とし、英文穴埋め問題における学習データの質による影響の比較調査および、英文穴埋め問題における文章ベクトルの利用について調査した結果について報告する。

2. 関連研究

この章では、まず単語の共起に関する学習を行うことで単語の意味的・統語的な表現をベクトルに割り当てるvLBLモデルについて述べる。次にvLBLでは無視してしまう語順に関する学習を行うvLBL(c)をvLBLを組み合わせることで、語順と共起両方に関する学習を可能としたvLBL+vLBL(c)モデルについて述べる。最後に文や段落などの文章にベクトルを割り当て、その文章内の各単語ベクトルとともに学習を行うことで、定めた文脈の外にある単語の情報の利用を可能とした文章ベクトルについて述べる。

2.1 vLBL

ニューラル言語モデルの一種であるvector Log-bilinear language model (vLBL) [6]では、各単語をベクトルで表現し、文中の1つの単語を対象として、その単語のベクトルと周辺の単語のベクトルの類似度が高くなるように学習を行う。この学習により単語の共起に関する知識を得ることができる。その結果、似たような文脈で扱われる単語は、似た特徴のベクトルを持つことになり、単語の意味的・統語的な特徴をベクトルで表現することができる。例えば「今日/は/雨/が/降る」と「明日/は/雪/が/降る」のように、「今日」と「明日」や、「雨」と「雪」など似た文脈で使われる単語は似た特徴のベクトルを持つことになる。vLBLによる学習の様子を図2に示す。図2では、「send」を対象の単語としてそのベクトルと、文脈の単語「Tomorrow」、「we」、「will」、「the」、「letter」、「by」のベクトルを類似させ、対象の「send」を「room」などの文脈外の単語と置換し、こちらは類似していない単語として相違させている様子を示している。対象の単語 w_t から相対的に i 語離れた位置にある単語 w_{t+i} のベクトルを \mathbf{w}_{t+i} として、 w_t から前後 n 単語を w_t の文脈 c_t と定めたとき、 c_t のベクトル \mathbf{c}_t を次式で表す。

$$\mathbf{c}_t = \frac{1}{2n} \sum_{-n \leq i \leq n, i \neq 0} \mathbf{w}_{t+i} \quad (1)$$

対象の単語 w_t のベクトルを \mathbf{w}_t 、 w_t に対応するバイアス

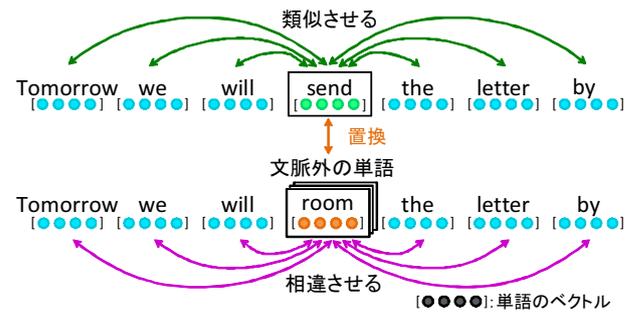


図2 vLBLによる学習

を b_t としたとき、式(1)を用いて、vLBLにおける w_t と c_t の類似度を示すスコア関数 $s_{vLBL}(c_t, w_t)$ を次式で表す。

$$s_{vLBL}(w_t, c_t) = \mathbf{c}_t \cdot \mathbf{w}_t + b_t \quad (2)$$

(2)式で定義したスコア関数を効率よく学習するために、文脈外の単語 $w_{t'}$ を生成し、 c_t とは相違しているとして、逆向きの学習を同時に行うNegative Sampling(NEG) [5]を用いる。対象の単語を w_t として学習に用いる目的関数 g_t を次式で表す。

$$g_t = \log \sigma(s(c_t, w_t)) + \sum_{w_{t'} \sim P_n}^k \log(1 - \sigma(s(c_t, w_{t'}))) \quad (3)$$

ただし $\sigma(x)$ はロジスティック関数、 P_n は単語の頻度分布、 k はNEGで生成する単語数、第2項は P_n から k 単語をサンプルすることを示す。式(2)を式(3)で用いることでvLBLの目的関数を得る。得られる目的関数を最大化するように各ベクトルを学習することで、対象の単語と文脈内の各単語のベクトルは類似し、文脈外の単語と文脈内の各単語のベクトルは相違させる。

2.2 vLBL+vLBL(c)

vLBLでは、文脈内の各単語が対象の単語 w_t に対して相対的にどの位置にあるかを考慮せず、文脈内で共起しやすい単語の類似度が高くなるように学習を行っている。しかし、この学習で得られる単語のベクトルは、語順に関する知識を十分に学習できていない。そこで[8]において、語順に関する学習を行っているvLBL with position-dependent contexts (vLBL(c))とvLBLを組み合わせ、語順と共起に関する学習を行っているvLBL+vLBL(c)を提案した。

vLBL(c)では、文脈内の各単語に w_t との相対的な位置に依存したベクトル \mathbf{w}_{t+i}^p と、そのベクトルが w_t に作用するベクトル \mathbf{w}_t^p の類似度が高くなるように学習を行うことで、語順に関する知識を得る。vLBL(c)による学習の様子を図3に示す。図3では、対象の単語「send」に対応するベクトルと文脈の各単語のそれぞれ位置に対応したベクトルを類似、対象の単語を文脈外の単語に置換し相違させている様子を示している。vLBL(c)における文脈 c_t のベクトル \mathbf{c}_t は次式で表す。

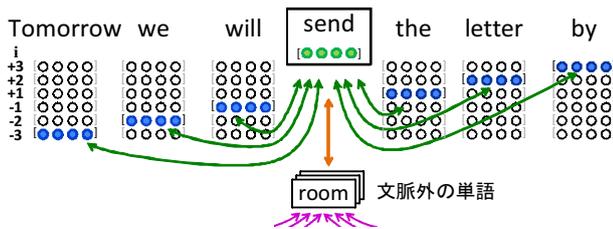


図3 vLBL(c)による学習

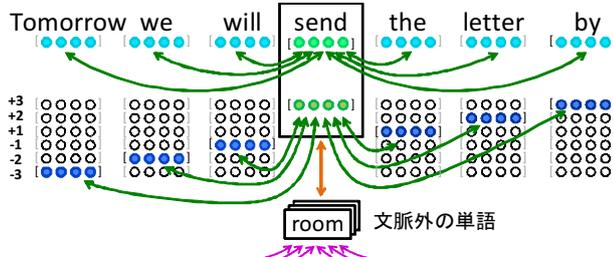


図4 vLBL+vLBL(c)による学習

$$\mathbf{c}_t^p = \frac{1}{2n} \sum_{-n \leq i \leq n, i \neq 0} \mathbf{w}_{t+i}^i \quad (4)$$

式(4)を用いて、vLBL(c)における文脈 c_t と w_t の類似度を示すスコア関数を $s_{vLBL(c)}(c_t, w_t)$ を次式で表す。

$$s_{vLBL(c)}(c_t, w_t) = \mathbf{c}_t^p \cdot \mathbf{w}_t^p + b_t \quad (5)$$

式(5)を式(3)で用いることで、vLBL(c)の目的関数を得る。共起に関する学習を行っているvLBLと語順に関する学習を行っているvLBL(c)を組み合わせたモデルとしてvLBL+vLBL(c)における文脈 c_t と w_t の類似度を示すスコア関数を $s_{vLBL(c)}(c_t, w_t)$ を次式で表す。

$$s_{vLBL+vLBL(c)}(c_t, w_t) = \mathbf{c}_t \cdot \mathbf{w}_t + \mathbf{c}_t^p \cdot \mathbf{w}_t^p + b_t \quad (6)$$

式(6)を式(3)で用いることで、vLBL+vLBL(c)の目的関数を得る。vLBL+vLBL(c)による学習の様子を図4に示す。図4では、上の部分がvLBLによる学習、下の部分がvLBL(c)による学習、さらに対象の単語を文脈外の単語に置換し相違させている様子を示している。

2.3 PV-DM

The Distributed Memory Model of Paragraph Vectors (PV-DM) [2] とは、文や段落、文書など可変長の文章を文章ベクトルとして固定長のベクトルで表現し、それを用いて単語ベクトル学習・予測するモデルである。同一文章内では大抵ある一つ話題に則って書かれている。そこでPV-DMでは文章ベクトルを用いて学習を行うことで、その話題の内容を文章ベクトルに記憶させることができる。そして、その文章ベクトルとともに各単語のベクトルの学習を行うことで、それらの単語がどのような話題、文章に現れやすいのかについて、学習することができるようにな

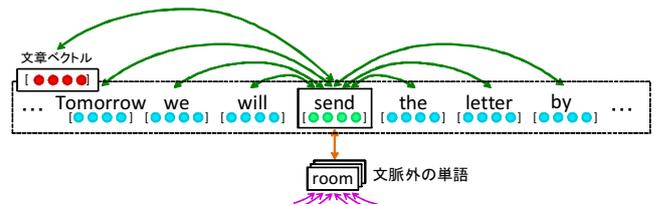


図5 PV-DMによる学習

る。そのため、その文章ベクトルとともに単語ベクトルを学習することで、単語の特徴をより精確に表現できるようになると推測できる。

PV-DMでは2種類の方法が提案されている。それは対象の単語の推定を行う際に単語ベクトルと文章ベクトル D の平均する手法と結合する手法である。平均を用いるPV-DMでは、文章ベクトルを用いて文脈ベクトル \mathbf{c}_t^D を次式で表す。

$$\mathbf{c}_t^D = \frac{1}{2n+1} \sum_{-n \leq i \leq n, i \neq 0} (\mathbf{w}_{t+i} + \mathbf{D}) \quad (7)$$

式(7)を用いて、平均を用いるPV-DMのスコア関数を $s_{PV-DM(ave)}(c_t, w_t, D)$ を次式で表す。

$$s_{PV-DM(ave)}(c_t, w_t, D) = \mathbf{c}_t^D \cdot \mathbf{w}_t + b_t \quad (8)$$

式(8)を式(3)で用いることで、PV-DMの目的関数を得る。なお、このモデルはvLBLに文章ベクトルを持たせたモデルと捉えられる。

結合を用いるPV-DMでは、 \mathbf{w}_{t+i} および \mathbf{D} を結合したベクトルを $\mathbf{v}^{con} = [\mathbf{w}_{t-n}, \dots, \mathbf{w}_{-1}, \mathbf{w}_1, \dots, \mathbf{w}_{t+n}, \mathbf{D}]$ とし、対象の単語のベクトルを $\mathbf{w}_t^{con} = [\mathbf{w}_t^{t-n}, \dots, \mathbf{w}_t^{-1}, \mathbf{w}_t^1, \dots, \mathbf{w}_t^{t+n}, \mathbf{w}_t^D]$ として、スコア関数を $s_{PV-DM(con)}(c_t, w_t, D)$ を次式で表す。

$$s_{PV-DM(con)}(c_t, w_t, D) = \frac{1}{2n+1} (\mathbf{v}^{con} \cdot \mathbf{w}_t^{con}) + b_t \quad (9)$$

ただし、 \mathbf{w}_t^i , \mathbf{w}_t^D は、 \mathbf{v}^{con} における \mathbf{w}_{t+i} , \mathbf{D} に対応するベクトルを表している。結合を用いるPV-DMは、このように文脈の単語ベクトルを結合させて類似度の計算を行っているのでこの手法は語順を考慮しているモデルであるといえる。

このようにPV-DMは2種類の手法が提案されているが、本研究では単語ベクトルと文章ベクトルの平均をとる手法を用いた。平均を用いるPV-DMによる学習の様子を図5に示す。図5では、対象の単語“send”に対応するベクトルとその文脈の各単語に対応するベクトル、対象の単語およびその文脈が含まれる文章に対応するベクトルを類似させ、さらに対象の単語を文脈外の単語に置換し相違させている様子を示している。

2.4 Adam

適切に学習率を調整させ、高速かつ高精度の学習を行う Adaptive Moment Estimation(Adam) [3] がある。Adam では、モデルのパラメータごとの学習率を自動的に調整する。モデルのパラメータベクトル θ の各要素における Adam の更新式を次式で表す。

$$\mathbf{m}_j = \beta_1 \lambda^{j-1} \mathbf{m}_{j-1} + (1 - \beta_1) \nabla \mathbf{g}_j \quad (10)$$

$$\mathbf{v}_j = \beta_2 \mathbf{v}_{j-1} + (1 - \beta_2) (\nabla \mathbf{g}_j \odot \nabla \mathbf{g}_j) \quad (11)$$

$$\alpha_j = \frac{\alpha \sqrt{1 - \beta_2^j}}{1 - \beta_1^j} \quad (12)$$

$$\theta_j = \theta_{j-1} + \frac{\alpha_j \mathbf{m}_j}{\sqrt{\mathbf{v}_j} + \epsilon} \quad (13)$$

ただし、 j は更新回数、 α は学習率、 $\nabla \mathbf{g}_j$ は目的関数 g_t におけるパラメータ θ に対する勾配、 λ は更新ごとに学習率を小さくするためのパラメータ、 ϵ はパラメータの発散を防ぐための小さな正の値である。また、 \mathbf{m}_j および \mathbf{v}_j はモーメントベクトルであり、指数移動平均の計算を行っている。 \mathbf{m}_j および \mathbf{v}_j の各要素の初期値は 0 とする。

3. 提案手法

この章では、まず英文穴埋め問題における学習データの質による影響について調査するための手法を提案し、次に英文穴埋め問題における文章ベクトルの利用する手法および vLBL+vLBL(c) に文章ベクトルを組み込んだモデルを 2 種類提案する。

3.1 学習データによる影響

vLBL などのニューラル言語モデルによる教師なし学習の学習データとして Wikipedia がよく用いられている。Wikipedia は誰でも利用することができ、英語版 Wikipedia ではテキスト量 4 百万記事以上、単語数 7 千万語以上と充実している。ニューラル言語モデルの学習には語句が十分な頻度出てくる必要があるため、Wikipedia を学習データとすることは適しているといえる。しかし、Wikipedia は誰にでも編集することができるため、文法や語彙に誤りがあるなど、質の良いデータではない可能性がある。特に TOEIC などの英文穴埋め問題では、文法や語彙に関する問いが多く出題されている。そのため、Wikipedia は学習データにはふさわしくないのではないかと考える。そこで本研究では、イギリスの学術機関や出版社が多数参加して設立されたコンソーシアムによって管理される世界最大のイギリス英語の均衡コーパス The British National Corpus (BNC) を用いて、Wikipedia のデータとの比較実験を行った。

3.2 文章ベクトルの利用

英文穴埋め問題では空欄から離れた場所にある単語に

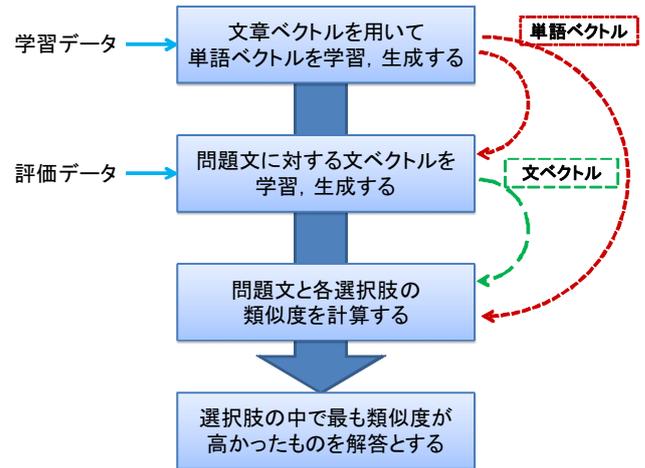


図 6 文章ベクトルを利用した英文穴埋め問題における解答の流れ

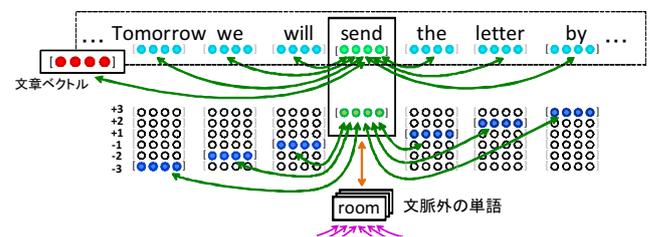


図 7 PV-DM+vLBL(c) による学習

注目しなければ解けない場合がある。しかし、vLBL や vLBL+vLBL(c) では、定めた文脈の領域よりも外の単語は考慮されないため、このような問題は解くことが難しい可能性がある。そこで、文や段落などの文章を文章ベクトルとして固定長のベクトルで表現し、学習を行う手法に注目した。文章ベクトルは、文章の内容を記憶することができるため、vLBL や vLBL+vLBL(c) では考慮できなかった文脈外の情報を利用することができる。これらを踏まえて本研究では、学習を 2 段階に分けて実験を行った。1 段階目は学習データに対する学習であり、学習データの 1 つの記事ごとに文章ベクトルを生成し、各単語の学習を行う。2 段階目は評価データに対する学習であり、一つの間ごとに文ベクトルを生成し、1 段階目で生成した単語ベクトルを用いて問題文の空欄以外の単語に対して文ベクトルの学習を行う。2 段階の学習を終えたら、1 段階目で生成した単語ベクトルおよび 2 段階目で生成した文ベクトルを用いて問題文と各選択肢の類似度の計算し、最も類似度の高い選択肢を解答とする。文章ベクトルを用いた、学習から解答の予測までの一連の流れを図 6 に示す。

vLBL+vLBL(c) に (7) から得られる文章ベクトルを持たせたモデルとして PV-DM+vLBL(c) モデルを提案する。PV-DM+vLBL(c) による学習の様子を図 7 に示す。図 7 では、上の部分が PV-DM による学習、下の部分が vLBL(c) による学習、さらに対象の単語を文脈外の単語に置換し相違させている様子を示している。スコア関数 $s_{PV-DM+vLBL(c)}(c_t, w_t, D)$ を次式で表す。

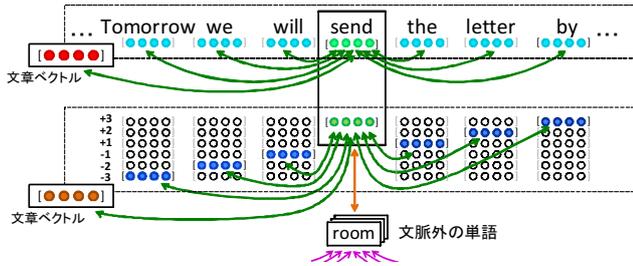


図 8 PV-DM+PV-vLBL(c) による学習

$$s_{PV-DM+vLBL(c)}(c_t, w_t, D) = c_t^D \cdot w_t + c_t^p \cdot w_t^p + b_t \quad (14)$$

式 (14) を式 (3) で用いることで、PV-DM+vLBL(c) の目的関数を得る。

PV-DM+vLBL(c) では、単語の位置に関する項が文章の情報を利用できていないので、さらに単語の位置に関する項には文章ベクトルをもたせ、それをを用いるモデル PV-DM+vLBL(c) with position paragraph vector (PV-DM+PV-vLBL(c)) も提案する。vLBL+vLBL(c)+PV による学習の様子を図 8 に示す。図 8 では、上の部分が PV-DM による学習、下の部分が対象の単語の位置に関するベクトルと文脈の各単語のそれぞれの位置に対応したベクトル、対象の単語とその文脈が含まれる文章に対応するベクトルを類似させ、さらに対象の単語を文脈外の単語に置換し相違させている様子を示している。位置に関する文章ベクトル D^p を用いた位置に関する文脈ベクトル c_t^{pD} を次式で表す。

$$c_t^{pD} = \frac{1}{2n+1} \sum_{-n \leq i \leq n, i \neq 0} (w_{t+i}^i + D^p) \quad (15)$$

式 (14) と式 (15) を用いて、PV-DM+PV-vLBL(c) のスコア関数 $s_{PV-DM+PV-vLBL(c)}$ のスコア関数を次式で表す。

$$s_{PV-DM+PV-vLBL(c)}(c_t, w_t, D) = c_t^D \cdot w_t + c_t^{pD} \cdot w_t^p + b_t \quad (16)$$

式 (16) を式 (3) で用いることで、PV-DM+PV-vLBL(c) の目的関数を得る。

4. 実験設定

本実験では英文穴埋め問題において、学習データの質による影響および文章ベクトルの利用について調査するために、以下の 5 つのモデルを用いて比較実験を行った。

- vLBL
- vLBL+vLBL(c)
- PV-DM
- PV-DM+vLBL(c)
- PV-DM+PV-vLBL(c)

学習データには、BNC のデータ (4,433,757 語)、Wikipedia のデータ (73,767,109 語) および比較実験用に Wikipedia の

データを BNC と同等の単語数に分割したデータ (4,495,470 語) を用いた。また、数回しか出現していない単語は十分な学習が行えないため学習時間短縮のために、BNC および Wikipedia を分割したデータは最低 5 回以上、Wikipedia の全データは最低 50 回以上出てきている単語に絞って学習を行った。評価データには Webanhvan [1] から取得した TOEIC の練習問題を用いた。問題は 4 択の英文穴埋め問題となっており、全 1,228 問を開発セット 613 問、テストセット 615 問に分けて用いた。なお、学習データおよび評価データの前処理として、小文字化し、文に分割し、それぞれの文に対して Natural Language Toolkit (NLTK) *1 の “nltk.tokenize.treebank” パッケージを用いて、個々の単語と句読点を分割する処理を行った。

学習時間の短縮のために計算は並列化し、各ベクトルの更新は 100 回分をまとめてに行うミニバッチを用いた。NEG で生成する単語数 k を 5 個、文脈の領域 n を対象の単語の前後 5 単語とした。また、NEG で生成する単語は全単語のユニグラム頻度分布 P_n から抽出するが、抽出する際よく出現する単語 (a, the など) が必然的に選ばれやすくなるため、すべての単語の出現頻度を $3/4$ 乗してから、抽出する単語を選ぶようにしている [5]。

学習には Adam を用いた。Adam の各パラメータは $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $\lambda = 1 - 10^{-8}$ と [3] で推奨されている値を用いた。

実験は E5-2670 v2, コア数 10, 動作周波数 2.50GHz の CPU を 2 つ、メモリ 384GB の環境で行った。また、実装は C++ で行い、OpenMP で並列化し、コンパイラは g++ 4.8.3 を用いた。

5. 実験結果と考察

5.1 学習データによる影響

vLBL および vLBL+vLBL(c) を BNC, 比較実験用に分割した Wikipedia のデータでそれぞれ学習し、開発データで評価した結果を図 9 に示す。図 9 より vLBL および vLBL+vLBL(c) 両モデルに対して、学習データに BNC のモデルを用いた場合の各イテレーションごとの正答率は、Wikipedia を用いた場合の正答率よりも約 5% ずつ落ちていることが分かる。この結果より英文穴埋め問題において、学習データの質の影響は大きいことが分かる。次に vLBL および vLBL+vLBL(c) を BNC, 比較実験用に分割した Wikipedia のデータ Wikipedia の全データでそれぞれ学習し、開発データで評価した結果および学習に有する時間を表 1 に示す。結果として、Wikipedia の全データで学習した場合の正答率が最も高くなった。これは単純にデータ量が BNC に比べて Wikipedia は 16 倍以上であり十分に学習ができているということや、また Wikipedia の文章

*1 <http://www.nltk.org/>

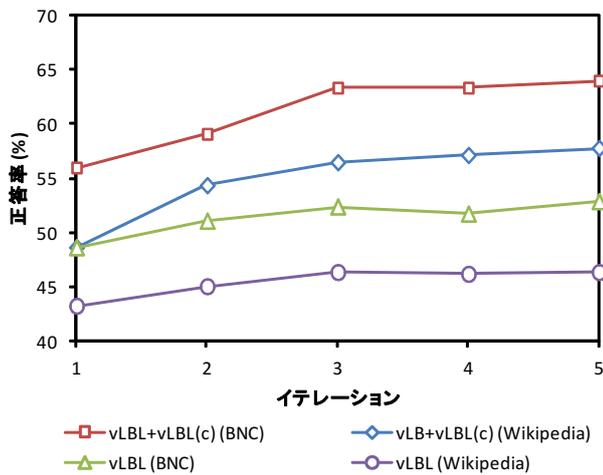


図 9 学習データごとの各モデルの学習曲線

は誰でも編集ができるが、基本的に何かを説明するために書かれている文章なので文法や語彙の誤りはあまり酷くはないという要因も考えられる。しかし、Wikipediaの全てのデータを学習する際、1イテレーションに掛かる時間はBNCの学習に比べて約8~9倍必要である。これらを踏まえて、Wikipediaなどデータ量が大きなデータを用いた方が正答率は高くなることが分かったが、一方でBNCなど質の良い文章が書かれているデータを用意することも重要であることが分かった。

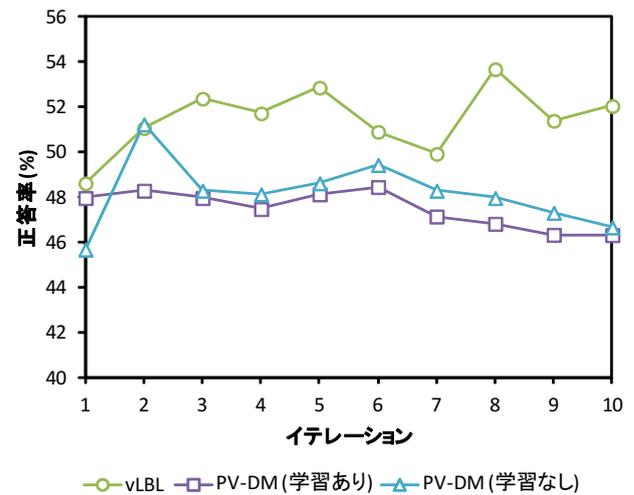
最後にvLBLおよびvLBL+vLBL(c)を評価した結果を表2に示す。結果として、開発データによる結果と同じようにBNCによる学習における正答率は分割したWikipediaによる学習における正答率よりも高い結果となった。

5.2 文章ベクトルの利用

開発データで次元数を100としてvLBLおよびPV-

モデル	正答率 (%)		
	Wikipedia		
	BNC	分割	全て
vLBL	53.67	46.33	59.38
vLBL+vLBL(c)	66.56	57.75	68.19
1イテレーションに有する時間 (h)			
vLBL	0.15	0.14	1.39
vLBL+vLBL(c)	0.33	0.27	2.78

モデル	正答率 (%)	
	BNC	Wikipedia (分割)
vLBL	52.20	51.06
vLBL+vLBL(c)	67.64	59.35



学習あり、学習なしはそれぞれ評価データの文ベクトルを学習した、しなかった場合である。

図 10 vLBL, PV-DM による学習曲線

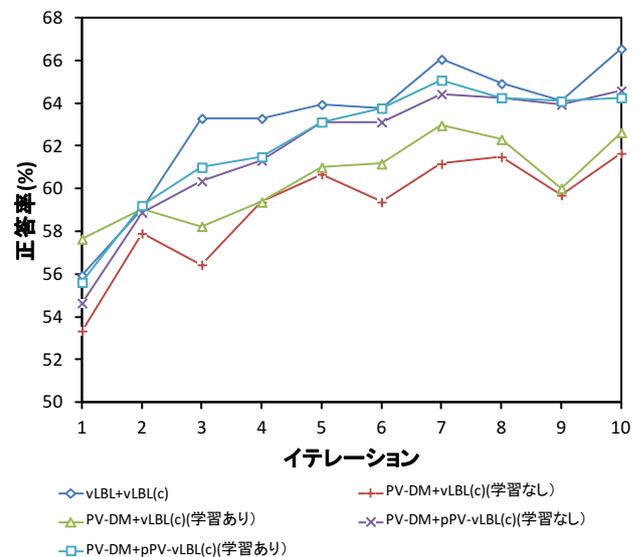


図 11 vLBL + vLBL(c), PV-DM+vLBL(c) による学習曲線

DMを評価した結果を図10, vLBL + vLBL(c), PV-DM+vLBL(c) および PV-DM+PV-vLBL(c) を評価した結果を図11に示す。なお、PV-DM, PV-DM+vLBL(c) および PV-DM+PV-vLBL(c) において (学習あり), (学習なし) と示してあるモデルは、評価データの学習を行った、行っていない場合の正答率である。つまり、評価データの学習を行っていない場合は学習データの学習により生成した単語ベクトルのみを用いて評価した場合の正答率である。また、PV-DM+vLBL(c), PV-DM+PV-vLBL(c) において、文ベクトルの学習するにおけるイテレーションは1~9回行い、正答率の最も高かったものを結果として用いている。図10, 11より、どちらもvLBL+vLBL(c)が最も高い正答率という結果となり、文章ベクトルを用いたモデルはどれも、元のモデルよりも悪い結果となった。

ここで、文章ベクトルを用いることにより、英文穴埋め問題における解答へがどの程度変化しているか調べるために、vLBL+vLBL(c)とPV-DM+PV-vLBL(c)の正答率の高かった出力結果を用いて、まとめた表を表3に示す。表3の結果よりマクネマー検定を行うとP値は0.4118となり、統計的に有意差があるとはいえない結果となった。この結果より、文脈情報のみでは誤答してしまう問題に対して、文章ベクトルを用いることで正答できる問題があれば、その逆も十分に起こりうることを示している。つまり、文章ベクトルを用いることで、英文穴埋め問題における解答には良くも悪くも影響を与えていることが分かった。

PV-DM, PV-DM+vLBL(c) および PV-DM+PV-vLBL(c)において、評価に文章ベクトルを用いた場合(学習あり)と用いなかった場合(学習なし)の比較では、PV-DMでは文章ベクトルを用いなかった方が高く、PV-DM+vLBL(c)では文章ベクトルを用いた方が高く、PV-DM+PV-vLBL(c)ではわずかであるが文章ベクトルを用いた方が高い結果となった。いずれにせよ、正答率に大きな差はなく文章ベクトルを用いることによる影響力が小さいことが分かる。これは文脈の領域を前後5単語としており、短い問題文に関しては文脈情報のみで文全体を網羅できるため、文章ベクトルによる影響が小さくなることが原因であると考えられる。しかし、PV-DM+PV-vLBL(c)では文章ベクトルを用いることでわずかに正答率が大きくなっているため、文章ベクトルによる影響力を大きくすることでさらに正答率を高くできるのではないかと推測できる。一方、vLBL+vLBL(c)とPV-DM+PV-vLBL(c)を比較するとvLBL+vLBL(c)の正答率の方が高いことを考えると、文章ベクトルによる影響は小さくした方がよいのではないとも考えられる。そこで、文章ベクトルによる影響力についての実験を行った。文章ベクトルによる影響力を変化させるために式(15)から得られるPV-DM+PV-vLBL(c)の文章ベクトルを次式のように変化させた。

$$\mathbf{c}_t^{pD} = \frac{1}{2n + \alpha} \sum_{-n \leq i \leq n, i \neq 0} (\mathbf{w}_{t+i}^i + \alpha \mathbf{D}^p) \quad (17)$$

ただし、 α はPV-DM+PV-vLBL(c)における文章ベクトルの影響力を表すパラメータである。式(17)による文章ベクトルを用いて、PV-DM+PV-vLBL(c)における文章ベクトル

表3 vLBL+vLBL(c)とPV-DM+PV-vLBL(c)による解答の正誤関係

		vLBL+vLBL(c)		
		正答	誤答	
PV-DM+PV-vLBL(c)	正答	356	43	399
	誤答	52	162	214
		408	205	613

ルの影響力における調査を行った結果を表4に示す。なお、イテレーション数は図11において、PV-DM+PV-vLBL(c)が最も高い正答率を出した7回で固定して行った。

$$\mathbf{c}_t^{pD} = \frac{1}{2n} \sum_{-n \leq i \leq n, i \neq 0} (\mathbf{w}_{t+i}^i) + \alpha \mathbf{D}^p \quad (18)$$

結果として、 $\alpha = 1.25$ のときの正答率が最も高い結果となり、文章ベクトルの影響力を少し強くすることで正答率が高くなることが分かった。しかし、vLBL+vLBL(c)単体での正答率には及ばなかった。

最後にテストデータでvLBL+vLBL(c)および $\alpha = 1.25$ としたPV-DM+PV-vLBL(c)を評価した結果を表5に示す。結果として、開発データによる評価と同じように文章ベクトルを用いたPV-DM+PV-vLBL(c)の正答率はvLBL+vLBL(c)の正答率に及ばなかった。

6. 結論

本研究では英文穴埋め問題における学習データの質による影響と文章ベクトルの利用に関する調査を行った。まず英文穴埋め問題における学習データの質による影響の比較調査の結果として、BNCで学習した結果は、BNCと同サイズに分割したWikipediaで学習した結果と比較すると、TOEICの問題において正答率は約5%高い結果となった。しかし、BNCよりも16倍以上大きいWikipediaの全データで学習した結果よりは低い結果となった。これらの結果より、英文穴埋め問題においてデータサイズの大きな学習データを用いた方がよいものの、学習データの質による影響も大きいことが分かった。次に英文穴埋め問題における文章ベクトルの利用についての調査については、文章ベクトルによる正答率の向上は得られなかった。この原因とし

表4 文章ベクトルの影響力に関する実験結果

モデル	正答率 (%)		
	α	学習なし	学習あり
vLBL+vLBL(c)	-	66.56	-
PV-DM+PV-vLBL(c)	0.25	63.13	63.30
PV-DM+PV-vLBL(c)	0.50	63.62	64.44
PV-DM+PV-vLBL(c)	0.75	63.62	63.62
PV-DM+PV-vLBL(c)	1.00	64.44	65.09
PV-DM+PV-vLBL(c)	1.25	64.27	65.58
PV-DM+PV-vLBL(c)	1.50	65.09	64.76
PV-DM+PV-vLBL(c)	1.75	64.11	64.76
PV-DM+PV-vLBL(c)	2.00	63.78	63.46
PV-DM+PV-vLBL(c)	5.00	62.97	63.30
PV-DM+PV-vLBL(c)	10.00	57.26	63.46

表5 テストデータにおける正答率

モデル	正答率 (%)
vLBL+vLBL(c)	68.13
PV-DM+PV-vLBL(c)	67.00

て、文脈ベクトルによる情報で解答を得られる問題において文章ベクトルによる情報は逆に悪影響を及ぼすことや、文脈の領域を前後5単語としたことにより短い問題文では文脈情報のみで文全体の情報を十分に得られているため文章ベクトルによる影響は小さいことなどが可能性として考えられる。今後は、各種パラメータの調整と問題の解析を進めるとともに、今回提案したモデルとは違った視点で文脈外の情報を利用できる方法について考えていく予定である。

参考文献

- [1] Webanhvan, <http://www.webanhvan.com>.
- [2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, Vol. 3, pp. 1137–1155, 2003.
- [3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [4] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196, 2014.
- [5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 26*, pp. 3111–3119. 2013.
- [6] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS 26*, pp. 2265–2273. 2013.
- [7] Geoffrey Zweig and Christopher J.C. Burges. The microsoft research sentence completion challenge. Technical Report MSR-TR-2011-129, December 2011.
- [8] 森洸樹, 三輪誠, 佐々木裕. 語順と共起を考慮したニューラル言語モデルによる英文穴埋め. pp. 760–763, 2015.