n-gram 素性に対する注意機構を利用したニューラルネットによる単語穴埋め

Sentence Completion with Neural Attentions on n-gram Features

森 洸樹　三輪 誠　佐々木 裕
Koki Mori　Makoto Miwa　Yutaka Sasaki
豊田工業大学
Toyota Technological Institute
{sd15435, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

1 はじめに

word2vec [1] や vector Log-Bilinear Language (vLBL) モデル [2] など，大規模コーパス上で単語の共起に関する学習をし，単語の意味情報を埋め込んだ単語ベクトルが言語処理の様々な分野で応用されている．しかし，これらの手法は語順を考慮していないことや単語のみを対象としているため，言語の統語的な特徴や熟語など複数の単語で成り立つ意味を十分に表現できていない．言語を処理する上で，それらの情報は重要である．語順を考慮することで統語的な特徴を学習する研究は盛んに行われている [3, 4]．また，n-gram を単語と同様のベクトルとして利用することで，複数の単語で成り立つ熟語などを考慮に入れ，文章ベクトルを生成した研究も報告されている [5]．一方で，LSTM など RNN を用いることで，統語的な特徴や文章を表現する手法は，言語処理の様々な分野で良い成果を挙げている [6, 7]．しかし RNN は計算量が多いことや，単語間の関係を十分に捉えることができないという問題がある．単語の特徴を十分に捉えた単語ベクトルを生成することは，単語単位でベクトルを扱え，さらに RNN の入力として有効的に使えるとも考えられる．本研究では，単語の意味や文法など言語の多角的な特徴を捉えている単語穴埋め問題の正答率の向上を目的として，n-gram 素性とアテンション機構を利用したモデルを提案する．

2 関連研究

関連研究として，共起に関する学習を行い単語ベクトルを生成する vLBL モデルとアテンション機構を用いることで語順を考慮に入れた CBOW(a) モデルに付いて説明する．

2.1 vector Log-Bilinear Language モデル

vector Log-Bilinear Language (vLBL) モデル [2] では単語間の共起に関する情報をテキストデータから学習することで，単語の意味的な特徴を実数値ベクトルに埋め込む．テキスト上のある一つの単語を対象としその周辺の単語を文脈として，単語と文脈の “類似度” を大きくするように学習を行う．対象の単語を w_t，文脈を $h_t = \{w_{t-l}, ..., w_{t-1}\}$ として，単語と文脈の類似度を表すスコア関数 $s_{vLBL}(w_t, h_t)$ は，以下のように計算する．

$$\mathbf{h}_t = \frac{1}{l}\sum_{i=1}^{l} \mathbf{c}_{w_{t-i}} \tag{1}$$

$$s_{vLBL}(w_t, h_t) = \mathbf{q}_{w_t} \cdot \mathbf{h}_t + b_t \tag{2}$$

ここで $\mathbf{c}_w$，$\mathbf{q}_w$ はそれぞれ文脈，対象の単語を表現するときに用いる単語 w に対するベクトル，b_t はバイアスである．

2.2 Continuous Bag-of-Words with attention モデル

Continuous Bag-of-Words with attention (CBOW(a)) モデルでは，文脈において，各単語が文脈内のどの位置に出現したかによって，その単語に対する注意のかけ方を変える．例えば，“the” という単語が直前に出現した場合は，その直後の単語の品詞は限られてくる．そのため “the” という単語が直前に出現した場合は，それ以外に出現した場合と比べて，“the” に対する注意の度合い (アテンション) を大きくすることを期待する．CBOW(a) モデルでは，文脈 $h_t = \{w_{t-l}, ..., w_{t-1}\}$ において，単語 w_{t-i} が文脈において位置 $-i$ に現れたとき，その単語に対するアテンション $att(w_{t-i}, -i)$ は以下のように表す．

$$att(w_{t-i}, -i) = \frac{\exp(a^{-i}_{w_{t-i}} + b_{-i})}{\sum_{j=1}^{l} \exp(a^{-j}_{w_{t-j}} + b_{-j})} \tag{3}$$

ここで，a^i_w は単語 w が文脈内で位置 i に出現した場合にどれだけ重要となるかを決定するパラメータ，b_i は単語の位置にのみ依存し，文脈内の位置 i にある単語がどれだけ重要かを表すバイアスである．式 (3) を用いて，文脈 h_t における，文脈ベクトル $\mathbf{h}_t$ を以下のように計算する．

$$\mathbf{h}^a_t = \sum_{i=1}^{l} att(w_{t-i}, -i)\mathbf{c}_{t-i} \tag{4}$$

この文脈ベクトル $\mathbf{h}_t$ を用いて，CBOW(a) によるスコア関数を以下のように表す．

$$s_{CBOW(a)}(w_t, h_t) = \mathbf{q}_{w_t} \cdot \mathbf{h}^a_t \tag{5}$$

これにより，文脈 h_t において予測する際に重要と判断した単語ほど，アテンションが大きくなる．

3 提案手法

vLBL モデルは単語のみを対象とし，さらに語順を考慮していないため，熟語など複数の単語で成り立つ意味や言語の統語的な特徴を十分に表現できない．そこで本研究では，隣り合った n 単語 (n-gram) を一つの素性として利用する手法を提案する．さらに CBOW(a) モデルで用いているアテンション機構を n-gram 素性に適用するモデルを 3 つ提案する．

3.1 vLBL-ngram モデル

vLBL-ngram モデルでは，単語だけでなく n-gram 素性もベクトルで表現することで，熟語など複数の単語で成り立つ意味も獲得する．vLBL-ngram では，文脈 $h_t = \{w_{t-l}, ..., w_{t-1}\}$ を表す文脈ベクトルを以下のように表す．

$$\mathbf{h}^{ng}_t = \frac{1}{\sum_{k=1}^{m}(l-(k-1))} \sum_{j=1}^{m} \sum_{i=1}^{l-(j-1)} \mathbf{ng}^j_{t,-i} \tag{6}$$

ここで，m は m-gram までを素性として扱うこと，$\mathbf{ng}^j_{t,i}$ はテキスト上の t の位置にある単語から見て i の位置にある j-gram に対応するベクトルを表す．この文脈ベクトルを用いて vLBL-ngram における単語 w_t と文脈の類似度を表すスコア関数は以下のように表す．

$$s_{vLBL-ngram}(w_t, h_t) = \mathbf{q}_{w_t} \cdot \mathbf{h}^{ng}_t + b_t \tag{7}$$

3.2 vLBL-ngram with attention モデル

vLBL-ngram モデルに対して，CBOW(a) モデルで用いているアテンションを適用した vLBL-ngram with attention (vLBL-ngram(a)) モデルを提案する．vLBL-ngram(a) モデルでは，文脈 $h_t = \{w_{t-l}, ..., w_{t-1}\}$ において i の位置に出現した j-gram の素性 $f^j_{t,i}$ に対するアテンションは以下のように表す．

$$att(f^j_{t,-i}, -i, j) = \frac{\exp(a^i_{f^j_{t,i}} + b^j_i)}{\sum_{n=1}^{m}\sum_{k=1}^{l-(n-1)} \exp(a^{-k}_{f^n_{t,-k}} + b^n_{-k})} \tag{8}$$

ここで，$a^i_{f^j_{t,i}}$ はテキスト上の t の位置ある単語から見て i の位置にある j-gram がどれだけ重要となるかを決定するパラメータ，b^j_i は素性の位置にのみ依存し，j-gram の素性が位置 i に出現した場合，どれだけ重要かを表すバイアス，m は m-gram までの組みあわせを素性として扱うことを表す．これにより，n-gram 素性ごとがどの位置に出現したかによってアテンションを求めることができる．

3.3 vLBL-ngram with attention grouped into n-gram モデル

vLBL-ngram with attention grouped into n-gram (vLBL-ngram(ag)) モデルでは，各 n-gram が文脈内のどの位置に出現したかによって，各 n-gram に対するアテンションを計算し，各 n-gram に対応するベクトルに加重として付加する．さらに n-gram を構成する単語数ごとに重要度が変わってくると考え，文脈を n-gram を構成する単語数ごとに分割し，単語数に対するアテンションも加重として付加する．vLBL-ngram(ag) モデルにおける文脈ベクトル $\mathbf{h}^{ng(ag)}_t$ は以下のように計算する．

$$att(f^j_{t,i}, i, j) = \frac{\exp(a^i_{f^j_{t,i}} + b^j_i)}{\sum_{k=1}^{l-(j-1)} \exp(a^{-k}_{f^j_{t,-k}} + b^j_{-k})} \tag{9}$$

$$\mathbf{h}^j_t = \sum_{i=1}^{l-(j-1)} att(f^j_{t,i}, i, j)\mathbf{ng}^j_{t,-i} \tag{10}$$

$$att_j = \frac{\exp(a_j + b_j)}{\sum_{l=1}^{m} \exp(a_l + b_l)} \tag{11}$$

$$\mathbf{h}^{ng(ag)}_t = \sum_{j=1}^{m} att_j \mathbf{h}^j_t \tag{12}$$

ここで $a^i_{f^j_{t,i}}$，$\mathbf{ng}^j_{t,-i}$ はそれぞれテキスト上の t の位置ある単語から見て i の位置にある j-gram の特徴 $f^j_{t,i}$ の重要度を表すパラメータとベクトル，b^j_i は位置に依存したバイアス，m は m-gram までの組みあわせを扱うこと，a_i と b_i はそれぞれ i 単語で構成される n-gram の重要度を表すパラメータとバイアスである．式 (12) で得た文脈ベクトルを式 (2) に適用することで vLBL-ngram(ag) のスコア関数を得る．

3.4 vLBL-ngram(ag) by using a uni-gram attention モデル

n-gram に対する位置ごとのアテンションを利用するとパラメータ数が多くなるため，n-gram に対するアテンションに自身を構成する単語のアテンションを利用する vLBL-ngram(ag) by using a uni-gram attention (vLBL-ngram(uag)) モデルを提案する．式 (9) で用いている，n-gram の重要度を表すパラメータ $a^i_{f^j_{t,i}}$ を以下のように書き換える．

$$a^{-i}_{f^j_{t,-i}} = \frac{1}{j}\sum_{k=-i}^{-i+j} a^{-k}_{w-k} \tag{13}$$

ここで a^{-k}_{w-k} は文脈において位置 $-k$ に現れた単語 w の重要度を表すパラメータである．

4 実験

以下の 7 つのモデルに対して英文穴埋め問題を用いて，n-gram 素性の効果およびアテンション機構の効果について評価および既存モデルとの比較を行う．

- vLBL モデル
- vLBL-ngram モデル (提案モデル)
- vLBL-ngram(a) モデル (提案モデル)
- vLBL-ngram(ag) モデル (提案モデル)
- vLBL-ngram(uag) モデル (提案モデル)

4.1 実験設定

2 種類の英文穴埋め問題を用いて各モデルの評価を行った．1 つ目は “english-test.net” から取得した 4 択の英文穴埋め問題 (ETN) [8] である．取得した問題数は 1,228 問であり，613 問を開発セット，615 問をテストセットとした．2 つ目はホームズの小説を基に作成された 5 択の英文穴埋め問題 (MSR) [9] である．問題数は 1,040 問あり，それらを 520 問ずつに分割し，開発セット・テストセットとした．モデルの学習に用いるコーパスとして，ETN では British National Corpus，MSR ではプロジェクト・グーテンベルグにより公開されている小説 522 冊を用いた．各モデルパラメータの更新には Adam を用いて，512 回分まとめて行うミニバッチ処理を行った．n-gram 素性は，tri-gram までを対象とした．目的関数は，負例抽出を行う Sampled Softmax Loss を用いて，1 回の更新で抽出する負例数は 10 とした．学習はコーパス内の全ての単語に対して 20 回行い，1 回終えるごとに開発セットにおける正答率を計算し，最も正答率の高かった回のモデルパラメータをそのモデルの最適なパラメータとした．学習の対象とする n-gram 素性はコーパス内で，ETN では 80 回以上，MSR では 40 回以上出現した素性とした．

4.2 結果と考察

ベクトルの次元数を 100，文脈の領域を前後 5,10,15,20 単語とし，開発セット上で各モデルの実験を行った結果を表 1 に示す．結果としてアテンションを用いていないモデルでは，文脈の領域を広げるにつれて正答率が悪くなり，アテンションを利用しているモデルでは正答率を保持していることが分かる．アテンションを用いることで，直近に素性に注意できること，情報量の少ない素性に注意しないことができていることが伺える．また，n-gram 素性を利用することで正答率が高くなることも確認でき，n-gram 素性を利用して文脈を表現することの有効性が分かる．n-gram 素性間の類似度を計算したところ “look for” には “search for”，“look at” には “stared at” や “glance at”，“look after” には “care for” と n-gram 素性の意味も捉えられており，隣り合う単語で意味の変わる単語による曖昧性も解消できることが分かった．

次に MSR において，次元数を 600 に設定しアテンションを用いた提案モデルと既存モデルの比較を行った結果を表 2 に示す．比較手法として，ivLBL モデル [2] と RMN モデル [6] の正答率を示す．ivLBL モデルとは，vLBL モデルとは逆のモデルで単語からその周辺の単語を予測するモデルである．RMN モデルとは，LSTM の出力にアテンションを用いた手法である．結果として，RMN モデルにおける正答率には及ばない結果となった．MN モデルは LSTM を用いて，時系列を記憶し，さらにアテンション機構を用いることで，必要な情報のみに注意できるモデルである．一方で本提案モデルと比較すると，パラメータ数が多く，計算にも多大な時間を有する．

最後に ETN において，次元数を 300,600 に設定しアテンションを用いた提案モデルと既存モデルの比較を行った結果を表 3 に示す．比較手法として，Google 社による n-gram の頻度表 Google n-gram データセット [10] を用いて選択肢と問題文のマッチング数により解答選択を行った手法 [11] と単語が文脈内において出現した位置により扱うベクトルを変えるモデルと vLBL モデルを組合せた vLBL(c) モデルによる正答率を示す．結果として，Google n-gram dataset を用いた手法が最も高い正答率という結果になった．Google n-gram データセットは膨大なデータ量を含んでおり，かつ，この手法では選択肢を含んだ上で 5-gram までの組合せを考慮している．そのため，高い正答率を得られたのではないかと考えられる．本提案モデルでは tri-gram までの組合せを扱い，さらに選択肢を含めた n-gram は考慮できていない．そこで，n-gram の組み

表 1　開発セット上で各評価データに対する文脈の領域ごとの各モデルの正答率 (%)

	MSR				ETN			
モデル	5	10	15	20	5	10	15	20
vLBL	49.3	48.3	46.3	45.8	50.4	46.5	46.2	44.4
vLBL-ngram	51.3	50.8	48.9	47.3	54.3	52.0	52.2	49.1
vLBL-ngram(a)	49.0	51.7	**52.3**	51.2	56.6	57.1	56.8	57.6
vLBL-ngram(ag)	49.0	51.7	51.3	49.6	**60.8**	58.2	59.1	57.6
vLBL-ngram(uag)	48.5	50.6	52.1	50.8	59.1	58.7	60.0	57.4

合わせ数を増やすこと，選択肢を含めた n-gram も考慮に入れることで，正答率の向上につながるのではないかと考えられる．

5 おわりに

単語穴埋め問題における正答率の向上を目的として，n-gram 素性とアテンションを利用したモデルを提案した．結果として 2 種類の英文穴埋め問題を用いて，n-gram 素性を用いてアテンション機構を利用することで正答率の向上する結果が得られた．n-gram 素性を利用することで，隣り合う数単語で成り立つ意味や，隣り合った単語によって意味の変わる曖昧性を含む単語を捉えることができ，さらにアテンションを利用することで語順を考慮に入れることの有効性が分かった．今後の課題として，予測する単語に寄せたアテンションを利用することなどが挙げられる．

参考文献

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*. 2013.

[2] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*. 2013.

[3] 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. pages 760–763, 2015.

[4] Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W Black, Isabel Trancoso, and Chu-Cheng Lin. Not all contexts are created equal: Better word representations with variable attention. In *EMNLP*, pages 1367–1372, 2015.

[5] Bofang Li, Tao Liu, Xiaoyong Du, Deyuan Zhang, and Zhe Zhao. Learning document embeddings by predicting n-grams for sentiment classification of long movie reviews. *CoRR*, abs/1512.08183, 2015.

[6] Ke Tran, Arianna Bisazza, and Christof Monz. Recurrent memory networks for language modeling. In *NAACL*, pages 321–331, San Diego, California, 2016.

[7] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *CoNLL*, 2016.

[8] english-test.net, http://www.english-test.net.

[9] Geoffrey Zweig and Christopher J.C. Burges. The microsoft research sentence completion challenge. Technical Report MSR-TR-2011-129, 2011.

[10] Google ngram dataset, http://storage.googleapis.com/books/ngrams/books/datasetsv2.html.

[11] 佐々木 裕，白石 裕次郎. 英語穴埋め問題の自動解法. In *NLP*, pages 101–104. 2014.

表 2　MSR における実験結果

モデル	閾値 *1	次元数	文脈の領域	開発	テスト	全て
ivLBL [2]	10	600	前後 5 単語	-	-	55.5
RMN [6]	10	512	前 15 単語	-	-	**69.2**
vLBL-ngram(a)	40	600	前後 15 単語	56.9	55.6	56.2
vLBL-ngram(ag)	40	600	前後 15 単語	56.2	55.2	55.7
vLBL-ngram(uag)	40	600	前後 15 単語	57.3	54.6	56.0
vLBL-ngram(ua)	10	600	前後 15 単語	56.3	56.9	56.6
vLBL-ngram(uag)	10	600	前後 15 単語	57.9	57.3	57.6

∗1 単語数削減のための閾値

表 3　ETN における実験結果

モデル	閾値 *1	次元数	文脈の領域	開発	テスト
n-gram モデル [11]	-	-	-	-	**73.1**
vLBL(c) [3]	5	300	前後 5 単語	70.2	72.7
vLBL-ngram(ag)	80	300	前後 5 単語	64.8	67.6
vLBL-ngram(uag)	80	300	前後 5 単語	62.3	68.1
vLBL-ngram(ag)	80	300	前後 15 単語	65.7	67.8
vLBL-ngram(uag)	80	300	前後 15 単語	65.7	64.8
vLBL-ngram(ag)	80	600	前後 5 単語	66.7	70.4
vLBL-ngram(uag)	80	600	前後 5 単語	65.3	68.0
vLBL-ngram(ag)	80	600	前後 15 単語	66.6	72.7
vLBL-ngram(uag)	80	600	前後 15 単語	65.7	67.8

∗1 単語数削減のための閾値