系列編集モデルに基づく単語ベクトルからの定義文生成

石渡祥之佑 † 林佑明 ‡ Graham Neubig‡ 吉永直樹 § 豊田正史 § 喜連川優 ¶§
† 東京大学 ‡Carnegie Mellon University § 東京大学生産技術研究所 ¶ 国立情報学研究所
{ishiwatari, ynaga, toyoda, kitsure}@tkl.iis.u-tokyo.ac.jp
{hiroakih, gneubig}@cs.cmu.edu

1 はじめに

単語ベクトルは単言語コーパスから教師なし学習によって獲得可能であるため，未知語に対して意味表現を付与することのできる有力な手法である．これを活用し，単語ベクトルから人間が解釈可能な定義文への変換を行うタスクが Noraset ら [1] により提案されている．このタスクの性能改善は，任意の未知語に対する定義文の自動付与を実現し得る点で意義が大きい．

本タスクを解くにあたり，Noraset らは LSTM 言語モデルを用いた定義文の生成手法を提案した．この手法は教師あり学習によって単語ベクトルから単語系列（定義文）への写像関数を訓練する．この際，定義文特有の構造やパターンは考慮されず，あくまで単語ベクトルに条件付けられた言語モデルとして訓練される．

本研究では，単語の定義文には “the act of ...ing” や “of or pertaining to ...”，“one who ...” 等頻出のパターンが存在しており，かつ意味類似度の高い単語の定義文は共通のパターンで記述されることが多い点に着目する．たとえば，単語 “ruble” の定義文は “the basic unit of money in russia” であり，この語とベクトル空間における距離の近い “lira” の定義文は “the basic unit of money in turkey” である．こうした性質をふまえると，単語系列を生成する問題である定義文生成タスクは，既知の単語系列を適切に書き換える問題として解く方法が有効と考えられる．

一方，単語系列の生成を伴う様々なタスクにおいて，系列を編集するアプローチの有効性は既に報告されている．定義文生成とは異なるタスクであるが，Guu ら [2] は言語モデリングにおいて，訓練データから文をランダムに抽出し，その文を Encoder-decoder モデルにより編集する手法を提案している．また，Gu ら [3] は機械翻訳において，検索エンジンを用いて原言語文と類似度の高い対訳ペアを学習データから抽出し，翻訳の手がかりとして活用するモデルを提案している．

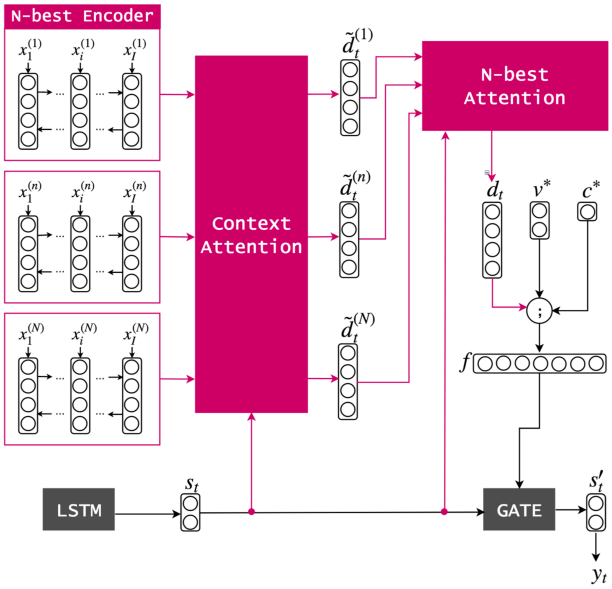

図1 提案手法：系列編集モデルに基づく定義文生成．赤色は提案手法で追加した機構を表す．

これらの研究はいずれも学習データに含まれる系列を編集する手法の有用性を示唆しているが，この手法を定義文生成タスクに適用する方法は自明ではない．そこで，本研究では単語ベクトルからの定義文生成タスクに特化した系列編集モデルを提案する．提案手法の概要を図 1に示す．提案モデルは，まず入力された単語との関連性が強い語の集合をベクトル空間における近傍探索により抽出する．抽出した関連語の定義文を LSTM エンコーダ（図 1左）を用いてモデルに入力し，LSTM 言語モデル（図 1下）がこれを適宜編集しつつ定義文生成を行うように学習を誘導する．

Wordnet / GCIDE 辞書コーパスを用いた実験において，提案手法は従来手法と比較して Perplexity 0.36 ポイント，BLEU 0.62 ポイントの改善が確認された．

2 単語ベクトルからの定義文生成

本研究で取り組む定義文生成は，事前訓練された単語ベクトルを入力とし，その単語の定義文を出力とするタスクである．Noraset ら [1] により提案された Seed + Gated + Character（以降，S+G+CH）は単語ベクトルから定義文への写像を学習するモデルであり，(1) LSTM 言語モデル，(2) CNN 文字情報エンコーダ，(3) ゲート関数で構成される．このうち，(2) は「似た文字列の単語は，似た意味を持ちやすい」という言語的性質をモデルに反映し，(3) は定義文生成に際して単語ベクトルの情報をどの程度重視するかを動的に制御する．

入力単語 w^* の定義文 $Y = \{y_1, ..., y_T\}$ の生成確率は，条件付き確率

$$p(Y|w^*) = \prod_{t=1}^{T} p(y_t|w^*, y_1, ..., y_{t-1}) \tag{1}$$

で表現される．この条件付き確率は LSTM 言語モデル [4] により近似され，時刻 t における単語 y_t の生成確率を以下のように算出する．

$$\boldsymbol{v}_0 = \boldsymbol{v}^* \tag{2}$$
$$\boldsymbol{s}_t = \mathrm{LSTM}(\boldsymbol{v}_{t-1}, \boldsymbol{s'}_{t-1}) \tag{3}$$
$$\boldsymbol{s}'_t = \mathrm{GATE}(\boldsymbol{s}_t, \boldsymbol{v}^*, \boldsymbol{c}^*) \tag{4}$$
$$p(y_t|\boldsymbol{y}_{<t}, w^*) = \mathrm{softmax}(\boldsymbol{W}_{s'}\boldsymbol{s}'_t + \boldsymbol{b}_{s'}). \tag{5}$$

ここで，$\boldsymbol{v}^*$ は単語 w^* の単語ベクトル，$\boldsymbol{s}_t$ は時刻 t における LSTM の隠れ層ベクトル，$\boldsymbol{v}_{t-1}$ は時刻 $t-1$ に出力された単語 y_{t-1} の埋め込み表現，$\boldsymbol{c}^*$ は CNN 文字情報エンコーダにより得られた単語 w^* の文字情報ベクトルである．また $\boldsymbol{W}_{s'}$，$\boldsymbol{b}_{s'}$ はそれぞれ GATE(·) 関数の出力ベクトル $\boldsymbol{s'}_t$ を語彙サイズ次元の空間に写像する重み行列とバイアス項である．

式(4)のゲート関数は以下のように定義される．

$$\boldsymbol{f} = [\boldsymbol{v}^*; \boldsymbol{c}^*] \tag{6}$$
$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_z[\boldsymbol{f}; \boldsymbol{s}_t] + \boldsymbol{b}_z) \tag{7}$$
$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r[\boldsymbol{f}; \boldsymbol{s}_t] + \boldsymbol{b}_r) \tag{8}$$
$$\tilde{s}_t = \tanh(\boldsymbol{W}_s[(\boldsymbol{r}_t \odot \boldsymbol{f}); \boldsymbol{s}_t] + \boldsymbol{b}_s) \tag{9}$$
$$\boldsymbol{s}'_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{s}_t + z_t \odot \tilde{\boldsymbol{s}}_t \tag{10}$$

ここで，$\sigma(\cdot)$ はシグモイド関数，$\odot$ はベクトルの要素積，; はベクトルの連結をそれぞれ表す．また各 $\boldsymbol{W}$ は重み行列，各 $\boldsymbol{b}$ はバイアス項である．時刻 t において，更新ゲート $\boldsymbol{z}_t$ はLSTMの出力 $\boldsymbol{s}_t$ をどの程度更新するかを制御し，リセットゲート $\boldsymbol{r}_t$ は $\boldsymbol{v}^*$ および $\boldsymbol{c}^*$ に含まれる情報が言語モデルに与える影響の強さを制御する．

3 提案手法

提案手法は，学習データ内に存在する単語 w^* の関連語集合の定義文を書き換えることで，単語 w^* の定義文を得る．図 1に提案手法の概要を示す．まず，単語ベクトルのコサイン類似度に基づき $\boldsymbol{v}^*$ の近傍探索を行い，上位 N 位までの単語 $\{w^{(1)}, ..., w^{(N)}\}$ を w^* の関連語とする．

次に，各関連語 $w^{(n)}$ について，その定義文 $X^{(n)} = \{x_1^{(n)}, ..., x_T^{(n)}\}$ を双方向 LSTM エンコーダ

$$\boldsymbol{h}_i^{(n)} = \text{Bi-LSTM}(\boldsymbol{h}_{i-i}^{(n)}, x_i^{(n)}) \tag{11}$$

により隠れ層ベクトル $\{\boldsymbol{h}_1^{(n)}, ..., \boldsymbol{h}_T^{(n)}\}$ にエンコードする（図 1左）．続いて，得られた N 個の隠れ層ベクトルの集合それぞれに対し，アテンション機構

$$\tilde{\boldsymbol{d}}_t^{(n)} = \text{C-Attention}(\boldsymbol{s}_t, \boldsymbol{h}^{(n)}) \tag{12}$$

による重み付け和を計算し，N 個の関連語それぞれに対する文脈ベクトル $\{\tilde{\boldsymbol{d}}_t^{(1)}, ..., \tilde{\boldsymbol{d}}_t^{(N)}\}$ を得る（図 1中央）．ここで，C-Attention(·) は各隠れ層ベクトル $\boldsymbol{h}_i^{(n)}$ に対する重み付け和を下式に基づき算出する．

$$\tilde{\boldsymbol{d}}_t^{(n)} = \sum_{i=1}^{T} \alpha_i^{(n)} \boldsymbol{h}_i^{(n)} \tag{13}$$
$$\alpha_i^{(n)} = \mathrm{softmax}(\boldsymbol{U}_h \boldsymbol{h}_i^{(n)\mathrm{T}} \boldsymbol{U}_s \boldsymbol{s}_t) \tag{14}$$

ここで，$\boldsymbol{U}_h$ と $\boldsymbol{U}_s$ はそれぞれエンコーダと言語モデルの隠れ層ベクトルを同一空間に写像する行列である．

さらに，N 語の関連語がそれぞれ w^* の定義文生成に対してどの程度支配的であるかを動的に制御考慮するため，式(12)とは異なる別のアテンション機構

$$\boldsymbol{d}_t = \text{N-Attention}(\boldsymbol{s}_t, \tilde{\boldsymbol{d}}_t) \tag{15}$$

によって N 個の文脈ベクトルの重み付け和を計算し，$\boldsymbol{d}_t$ を得る（図 1右上）．ここで，式(11)-(14)でエンコーダは N 文の定義文に対し，それぞれ独立に文脈ベクトルを計算している点に注意が必要である．これにより，$\tilde{\boldsymbol{d}}_t^{(n)}$ は n ごとに異なる空間に存在している．この差異を吸収するため，複数のエンコーダの出力を階層的に組み合わせる手法 [5] に倣い，N-Attention(·) ではエンコーダごとに独立した写像行列 $\boldsymbol{U}_d^{(n)}$ を導入し，下式のように d_t を算出する．

$$\boldsymbol{d}_t = \sum_{n=1}^{N} \beta_n \boldsymbol{U}_\beta^{(n)} \tilde{\boldsymbol{d}}_t^{(n)} \tag{16}$$
$$\beta_n = \mathrm{softmax}(\boldsymbol{U}_d^{(n)} \tilde{\boldsymbol{d}}_t^{(n)\mathrm{T}} \boldsymbol{U}_s \boldsymbol{s}_t) \tag{17}$$

コーパス	単語数	定義文数	平均文長
Train	21,998	146,789	7.63
Valid	2,574	18,689	7.71
Test	2,748	18,676	7.73

表1　定義文生成実験に用いたデータセット

ここで，式(14)の $\boldsymbol{U}_h$ が単一の行列であるのに対し，式(17)の $\boldsymbol{U}_d^{(n)}$ および式(16)の $\boldsymbol{U}_\beta^{(n)}$ はいずれも n ごとに異なる行列である．

最後に，得られた文脈ベクトル $\boldsymbol{d}_t$ を $\boldsymbol{v}^*$, $\boldsymbol{c}^*$ とともにゲート関数への入力とし，S+G+CH モデルの式(6)を

$$\boldsymbol{f} = [\boldsymbol{d}_t; \boldsymbol{v}^*; \boldsymbol{c}^*] \tag{18}$$

で置き換えることで，関連語の定義文の情報を言語モデルへと伝達する（図 1右下）．

4　実験

4.1　データセット

定義文生成タスクにおける提案手法の有用性を評価するため，先行研究 [1] に倣い `dict-definition`[*1]ツールキットを用いて WordNet[6][*2]および GCIDE[*3]から抽出された英語辞書データを実験に用いる．各エントリは単語と定義文のペアからなり，一つの単語に複数の定義文があるものも存在する．このデータセットをランダムに 3 分割し，それぞれ Train，Valid，Test データとした．分割の際，3 つのデータセット間で同じ単語が重複して出現しないよう制約を設けた．データセットの統計量は表 1に示す．

4.2　モデル

従来手法 S+G+CH モデルと提案手法はそれぞれ深層学習フレームワーク Pytorch[*4]によって実装した．提案手法が S+G+CH モデルと異なるのは，3節で述べた関連語のエンコーダ，およびそれを単一のベクトルに合成する 2 つのアテンション機構の存在のみであり，その他のハイパーパラメタは両手法で共通である．モデルの詳細は表 2に示す．

式(6)に示すように，$t = 0$ のとき LSTM への入力は単語ベクトル $\boldsymbol{v}^*$ となる．先行研究に倣い，CBOW モデル [7] によりニュースコーパスから学習されたベクトル[*5]を $\boldsymbol{v}^*$ として使用する．また，この単語ベクトルは LSTM 言語モデルの単語埋め込み層初期化にも使用する．

	S+G+CH	提案手法
Bi-LSTM の層数	-	2
Bi-LSTM の隠れ層 $\boldsymbol{s}$	-	300 * 2 次元
アテンション機構の隠れ層	-	300 次元
単語ベクトル $\boldsymbol{v}^*$	300 次元	
文字情報ベクトル $\boldsymbol{c}^*$	160 次元	
LSTM の層数	2	
LSTM の隠れ層 $\boldsymbol{h}$	300 次元	
語彙サイズ	20,000	
Dropout rate	0.2	
最適化手法	ADAM (推奨設定)	

表2　各モデルのハイパーパラメタ

4.3　評価手法

提案手法の有用性を評価するため，定義文生成タスクにおいて自動評価と人手評価を行う．自動評価には，先行研究に倣い `sentence-bleu.cpp`[*6]による文レベル BLEU の平均を使用する．

人手評価では，下記 5 段階のスコアにより定義文の評価を行う．

5　入力単語の定義文として完全に正しい．
4　定義文としておおよそ正しいが，情報の抜けや余計な情報，文法ミスが存在する．
3　完全に正しくはないが，入力単語と関連する語の定義文としては成立する．
2　ほぼ誤りだが，正解の定義文を一部含んでいる．
1　正解の定義文を全く含まず，誤りである．一単語を多数回繰り返している．定義文が入力単語を含む．

評価者は Test データからランダムに抽出した 300 例について，各手法の出力した定義文に 5 段階のスコアをつける．なお，この評価は英語母語話者 1 名が行った．

4.4　実験結果

ベースラインと提案モデルの出力例を表 3に示す．入力単語 “unpack” に対し，S+G+CH モデルは “to” から定義文生成をはじめており，入力単語が動詞であることは認識できていることがわかる．しかし，出力された定義文 “to put into a place or place” は誤りである．

[*1] https://github.com/NorThanapon/dict-definition
[*2] https://wordnet.princeton.edu/
[*3] http://gcide.gnu.org.ua/
[*4] http://pytorch.org/
[*5] https://code.google.com/archive/p/word2vec/
[*6] https://github.com/moses-smt/mosesdecoder/blob/RELEASE-4.0/mert

入力単語	unpack
参照文	to remove from its packing
	to separate and remove, as things packed
	to open and remove the contents of
提案手法	to remove the contents of
S+G+CH	to put into a place or place
関連語	定義文
stow	to pack
unwrap	to remove the outer cover or wrapping of
rearrange	to arrange again
alphabetize	to provide with an alphabet
assemble	to convene

表3　各手法の出力例，および関連語集合（$N = 5$）

一方，提案手法は関連語 “unwrap” の定義文に含まれる “to remove the”，“of” の二箇所をコピーし，“unpack” の定義文としてふさわしくない “the outer cover or wrapping” という箇所は別の単語列 “the contents” に編集していることが確認できる．

自動評価の結果を表 4に示す．提案手法（$N = 5$）では，ベースラインに対して BLEU 0.62 ポイント，Perplexity 0.36 ポイントの性能向上が確認された．また，モデルが使用する関連語の数 N を大きくしていくにつれ，BLEU，Perplexity がともに改善されていく傾向が見られる．ただし，$N = 2$ のときは提案手法の BLEU がベースラインを下回っており，提案手法の優位性を確認するためには，（1）学習時の揺らぎを考慮し複数回の試行を行う，（2）より大規模な辞書データセットで実験を行う，等の対応が必要となる．

人手評価によって得られた各手法の累積得点分布および平均得点を表 5に示す．平均得点ではベースラインが提案手法を僅かに上回っているが，ブートストラップ・リサンプリング法 [8] による検定では統計的優位差は確認されなかった．また，表より提案手法では得点が最低点の 1 であった割合がベースラインと比較して 5% 程度高いことが読み取れる．主な原因として，提案モデルの出力における未知語（学習データにおける出現頻度が上位 20,000 位以内に入らなかった語）の出現回数が，ベースラインよりも 10.7% 高かった点が挙げられる．提案モデルがベースラインよりも未知語を生成しやすい原因は未だ明らかでないが，この現象の詳細な調査，および語彙サイズを拡大した場合の実験は今後の課題である．

	BLEU	**Perplexity**
提案手法 ($N = 5$)	41.27	37.09
提案手法 ($N = 2$)	40.21	37.42
提案手法 ($N = 1$)	40.96	38.13
S+G+CH	40.65	37.45

表4　出力された定義文の平均 BLEU

	提案手法 ($N = 5$)	S+G+CH
2+	57.14	62.13
3+	40.53	42.19
4+	19.60	16.94
5	8.97	10.63
平均得点	2.26	2.32

表5　人手評価の累積得点分布および平均

5　おわりに

本研究では，定義文に頻出のパターンがある点に着目し，入力単語の関連語の定義文を原型とし，それを編集することで定義文生成を行うモデルを提案した．実験により提案手法が関連語の定義文を参照しつつ定義文生成を行うことが確認されたが，詳細な分析と有用性の評価は今後の課題である．

謝辞

本研究の一部は JSPS 科研費 17J06394，16K16109，16H02905 の助成を受けたものです．

参考文献

[1] T. Noraset, C. Liang, L. Birnbaum, and D. Downey. Definition modeling: Learning to define word embeddings in natural language. In *AAAI*, 2017.

[2] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang. Generating sentences by editing prototypes. *arXiv*, 2017.

[3] J. Gu, Y. Wang, K. Cho, and V. O.K. Li. Search engine guided non-parametric neural machine translation. In *AAAI*, 2018.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, 2010.

[5] J. Libovický and J. Helcl. Attention strategies for multi-source sequence-to-sequence learning. In *ACL*, 2017.

[6] G. A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[8] P. Koehn. Statistical significance tests for machine translation evaluation. In *EMNLP*, 2004.