

# SC-GRU encoder-decoder 모델을 이용한 자연어생성

김건영<sup>o</sup>, 이창기

강원대학교

uhi7074@gmail.com, leeck@kangwon.ac.kr

## Natural Language Generation Using SC-GRU Encoder-Decoder Model

Geonyeong Kim<sup>o</sup>, Changki Lee

Kangwon National University

### 요약

자연어 생성은 특정한 조건들을 만족하는 문장을 생성하는 연구로, 이러한 조건들은 주로 표와 같은 축약되고 구조화된 의미 표현으로 주어지며 사용자가 자연어로 생성된 문장을 받아야 하는 어떤 분야에서든 응용이 가능하다. 본 논문에서는 SC(Semantically Conditioned)-GRU기반 encoder-decoder 모델을 이용한 자연어 생성 모델을 제안한다. 본 논문에서 제안한 모델이 SF Hotel 데이터에서는 0.8645 BLEU의 성능을, SF Restaurant 데이터에서는 0.7570 BLEU의 성능을 보였다.

주제어: 자연어생성, 딥러닝, 머신러닝, GRU

### 1. 서론

자연어 생성은 특정한 조건들을 만족하는 문장을 생성하는 연구이다. 이러한 조건들은 주로 표와 같은 축약되고 구조화된 의미 표현으로 주어지며 사용자가 자연어로 생성된 문장을 받아야 하는 어떤 분야에서든 응용이 가능하다.

의미 표현	inform( name='colibri mexican bistro'; type=restaurant )
출력	colibri mexican bistro is a nice restaurant

표 1. 자연어 생성 예제

표 1은 자연어 생성 예제를 보여준다. 위 행은 입력으로 들어가는 의미표현으로, inform은 출력의 대화 의도(dialogue act)이고 소괄호 안의 정보들은 출력이 가져야 하는 슬롯들을 의미한다.

전통적인 자연어 생성 방법은 방대한 양의 언어 표현 지식과 템플릿을 손으로 구축하여 규칙 기반에 의존하는 방식이었다. 그러나 기계학습이 발전하면서, 통계 기반 방법[6]을 거쳐 현재에 들어와서는 딥러닝 기반 방법[1,3,4]이 지배하고 있다.

본 논문에서는 SC(Semantically Conditioned)-LSTM[1]의 DA(Dialogue Act) cell을 GRU(Gated Recurrent Unit)[2]에 적용한다. DA cell은 GRU 내부 마지막 히든 레이어 갱신에 one-hot 벡터로 된 슬롯 정보를 추가해주는 구조이다. 대화 의도는 그대로 1로 고정하고 슬롯은 출력이 생성될 때마다 레이어와 시그모이드 함수를 통하여 점차 1에서 0이 되도록 조정한다. 이 조정을 통제하기 위해 DA Cell 내부정보들을 이용한 목적함수를 사용한다.

전체적인 모델로는 encoder-decoder 모델을 활용한다.

의미표현은 단어표현(word embedding)의 형태로 일반 GRU encoder로 들어가 압축(encoding)이 된다. 압축된 의미표현은 SC-GRU decoder로 들어가고, 이 decoder는 자연어로 된 출력을 생성한다. 출력들은 beam search 알고리즘에 의해 생성되며 잘못된 슬롯들이 등장하는걸 방지하기 위해 생성된 문장의 스코어와 슬롯 에러율을 더하여 re-ranking한다.

### 2. 관련 연구

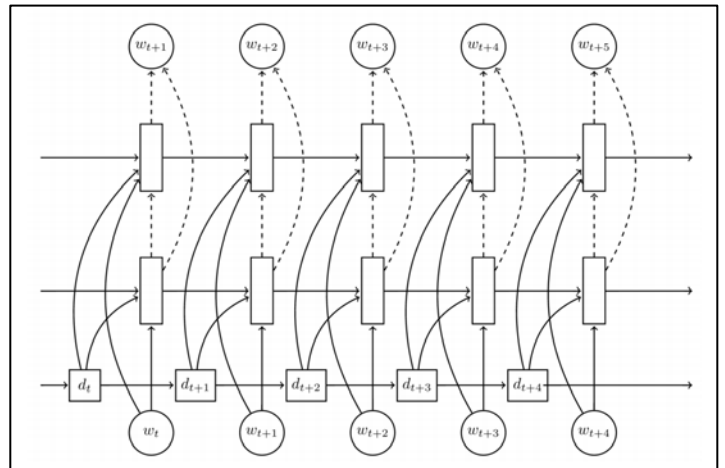


그림 1. Stacked SC-LSTM[1]

본 논문의 모태가 되는 [1]은 인공 신경망을 이용한 대표적인 자연어 생성 논문으로, SC-LSTM을 이용해 샌프란시스코에 있는 호텔과 레스토랑 데이터를 가지고 실험하였다.

그림 1은 [1]에서 사용한 모델로, 의미 표현을 one-hot 형태의 벡터로 주는 단일 decoder 모델이다. 본 논문에서 제안하는 모델은 일반 GRU를 이용한 encoder 구조를 추가하여 의미 표현의 추상적인 압축을 가능하게 한 점에서 [1]의 SC-LSTM 모델과 다르다.

[3]은 encoder-decoder 기반 RA(Refinement Adjustment)-LSTM을 제안하고, [1]과 같은 데이터와 tv, laptop 도메인 데이터를 실험하여 현존 최고 성능을 내었다. 본 논문의 encoder와 달리 RA-LSTM에서 encoder는 슬롯들을 정, 역방향으로 encoding하고 decoder의 매 출력시간마다 attention mechanism을 사용하여 의미 표현을 압축한다. Decoder는 encoder에서 압축된 의미표현을 참고하여 이전 시간 출력을 재정의하고 다음 시간 입력으로 주는 refinement cell과 one-hot형태의 의미 표현을 LSTM의 출력에 더해주는 adjustment cell로 이루어져 있다. Attention mechanism과 refinement, adjustment cell을 통해 RA-LSTM은 추가적인 목적함수의 제약없이 매 시간마다 one-hot 형태의 의미 표현을 0에서 1로 조정할 수 있으며, 이는 본 논문과 구별된다.

[4]에서는 Wikipedia의 인물 페이지들이 항상 인물의 정보가 축약된 표를 가지고 있는 점을 이용하여, 표를 보고 본문의 첫 문장을 생성하는 연구를 시도하였다.

[5]는 [1], [4]에서 사용한 데이터를 GRU기반 sequence-to-sequence 모델과 신경망 기계 번역에 쓰이는 여러 기법들을 써서 비교 실험하였다.

위와 다르게 [6]은 통계 기반 모델로, 의미 표현의 대화 의도와 슬롯 별로 클래스를 두었고, 클래스마다 언어 모델링(language modeling)을 하여 특징을 잘 살린 출력을 생성할 수 있었다.

### 3. 데이터

의미 표현	inform <b>SLOT_NAME</b> hotel    stratford <b>SLOT_ADDRESS</b> 242 powell street <b>SLOT_PHONE</b> 4153977080 <b>SLOT_POSTCODE</b> 94102 </s>
출력	the <b>SLOT_NAME</b> is located at <b>SLOT_ADDRESS</b> . , <b>SLOT_POSTCODE</b> . the phone number is <b>SLOT_PHONE</b> </s>

표 2. SF(San Francisco) Hotel 데이터

의미 표현	?select <b>SLOT_NEAR</b> russian hill or marina cow hollow </s>
출력	sorry would you like a restaurant near <b>SLOT_NEAR</b> </s>

표 3. SF Restaurant 데이터

본 논문에서 사용한 실험 데이터는 [1]에서 사용한 샌프란시스코 호텔, 레스토랑 관련 데이터이다. 이 데이터들은 표 1의 형태이지만 전처리를 하여 표 2, 3과 같은 형태로 만들고 탈 어휘화(delexicalize)를 통해 출력의 슬롯 값들은 슬롯 이름으로 대체하였다.

실제 decoding시에는 탈 어휘화된 문장을 생성하고 출력이 완성된 다음 후처리를 통해 슬롯 이름들을 슬롯 값으로 바꾼다.

[1]에서 실험한 SC-LSTM은 표 2, 3과 같은 입력이 사용되지 않고 슬롯 이름과 대화 의도만 one-hot 형태로 decoder에 주어진다.

표 4는 실험에 쓰인 데이터의 의미 표현을 구성하는 정보들이다. Act type은 대화 의도이고 그 아래 칸들은 슬롯들이다. 이진 슬롯은 네, 아니오로 구성되며 앞에 별표가 오는 슬롯들은 상관 없음 값을 포함한다.

	SF Restaurant	SF Hotel
act type	inform, inform-only, reject, confirm, select, request, reqmore, goodbye	
shared	name, type, *pricerange, price, phone, address, postcode, *area, *near	
specific	*food *goodformeal *kids-allowed	*hasinternet *acceptscards *dogs-allowed

**bold**=binary slots, \*=slots can take "don't care" value

표 4. 의미 표현 구성 정보[1]

### 4. 모델

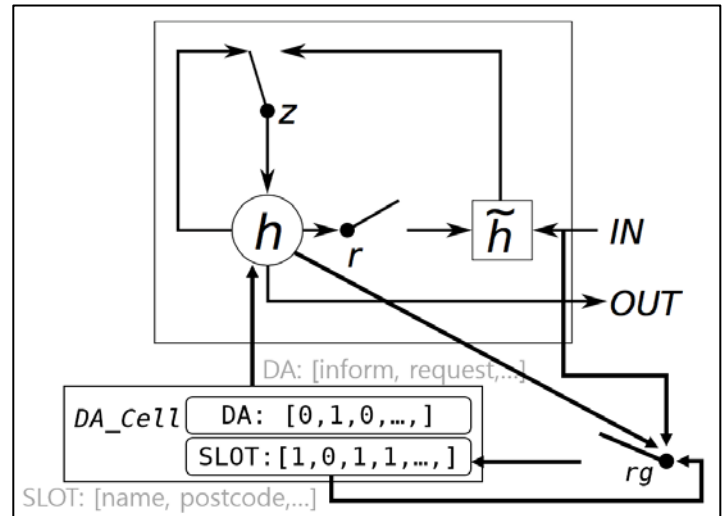


그림 2. SC-GRU 도식[2]

그림 2는 decoder로 쓰인 SC-GRU의 도식이다. 기존 GRU에 DA\_Cell이 붙은 구조로 이전 시간 슬롯 이름들이 시간에 따라 reading gate(rg)에 의해 1에서 점차 0으로 바뀌는 구조다. SC-GRU의 자세한 수식은 아래와 같다.

$$\begin{aligned}
 (1) \quad z_t &= \sigma(W_{zx}x_t + U_z h_{t-1} + W_{zc}c + b_z) \\
 (2) \quad r_t &= \sigma(W_{rx}x_t + U_r h_{t-1} + W_{rc}c + b_r) \\
 (3) \quad \tilde{h}_t &= \tanh(W_{hx}x_t + U_h(r_t \odot h_{t-1}) + W_{hc}c + b_h) \\
 (4) \quad rg_t &= \sigma(W_{rgx}x_t + U_{rg1}h_{t-1} \\
 &\quad + W_{rgc}c + U_{rg2}SLOT_{t-1} + b_{rg})
 \end{aligned}$$

$$(5) \text{SLOT}_t = \text{SLOT}_{t-1} \odot r g_t$$

$$(6) h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t + \tanh(W_{\text{hSLOT}}[DA; \text{SLOT}_t] + b_{\text{hSLOT}})$$

$x$ 는 한 단어의 벡터 표현(word embedding)이고, Encoder는 일반 GRU를 사용하며, 의미표현  $x$ 를 입력으로 받는다. 의미 표현이 압축된 형태인  $c$ 는 의미 표현을 역 방향으로 넣어 맨 마지막 GRU의 hidden state에 가중치를 곱하여 얻는다. SC-GRU 모델은 GRU로 이루어진 encoder를 사용함으로써 단순히 decoder에서 one-hot 벡터로 된 의미 표현만 보는 것보다 높은 추상화가 가능해진다.

Decoder의 입력으로 이전 시간에 생성된 단어가 들어간다.  $DA$ 와  $SLOT$ 은 모두 one-hot 형태의 벡터이다.  $DA$ 는 대화의도로서 출력되는 문장이 일관된 의도를 갖게 하기 위해 1로 고정한다.  $SLOT$ 은 의미 표현에서 등장하는 슬롯으로, 등장한 슬롯은 1로 나머지는 0으로 초기값을 주고 출력 시간마다  $rg$ 에 의해 값이 조절된다.

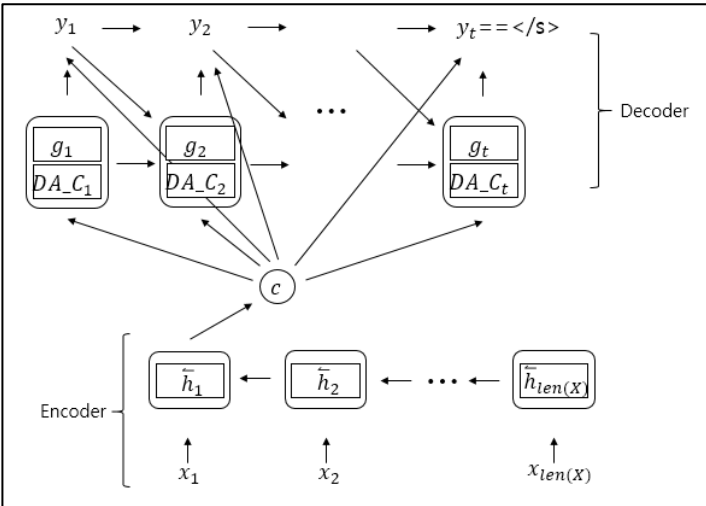


그림 3. SC-GRU encoder-decoder

그림 3은 자연어 생성을 위해 쓰인 SC-GRU encoder-decoder 모델의 전체 모습이다. [1]에서 사용한 SC-LSTM 모델과 다르게 encoder가 추가된 형태이다. Encoder의 일반 GRU를 통해 의미표현을 encoding한  $c$ 를 만들고, Decoder의 SC-GRU를 이용해 조건에 맞는 문장을 생성한다. 이때 decoder는 문장의 끝인  $\langle /s \rangle$ 를 만들면 멈춘다.

$$(7) P(y_{t+1} | y_t, y_{t-1}, \dots, y_1, c, DA, \text{SLOT}_t) = \text{softmax}(\tanh(W_{yy}y_t + W_{yh}h_t + W_{yc}c + b_y))$$

수식 (7)은  $y_{t+1}$ 의 확률 수식을 나타내고 있다. Decoder의 hidden state와 이전 시간 출력의 word embedding, 마지막으로 의미 표현의 압축형태인  $c$ 를 이용하여 출력을 결정한다.

$$(8) \mathcal{L}(\theta) = -\sum_t R_t \log y_t + \|\text{SLOT}_T\| + \sum_{t=1}^{T-1} \eta \xi^{\|\text{SLOT}_{t+1} - \text{SLOT}_t\|}$$

수식 (8)은 SC-GRU encoder-decoder 모델의 목적 함수이다.  $R$ 은 실제 정답을 의미하며 1 혹은 0이다. 수식의 첫 항은 크로스 엔트로피로  $P(R)$ 과  $P(y)$ 의 차이를 줄여준다. 두 번째 항은  $SLOT$  값이 꾸준히 줄어들게 해준다. 세 번째 항에서  $\eta$ 는  $10^{-4}$ ,  $\xi$ 는 100의 값을 가지며, 이 세 번째 항은 시간에 따른  $SLOT$  값의 변화를 최소화한다. 이러한 목적 함수의 정의로 SC-GRU는  $SLOT$ 의 값을 조절한다.

$$(9) \text{SLOT\_ERR} = \frac{p+q}{n}$$

$$(10) \mathcal{L}(\theta)_{all} = \mathcal{L}(\theta) + \lambda \cdot \text{SLOT\_ERR}$$

수식 (7)과 빔서치를 통해 문장을 생성한 후 여러 개의 문장 후보들을 슬롯 에러율을 이용하여 re-ranking한다. 수식 (9)는 슬롯 에러율의 정의를 보여준다.  $p$ 는 입력에는 등장하지만 출력에는 없는 슬롯의 개수,  $q$ 는 출력에는 등장하지만 입력에는 없는 슬롯의 개수,  $n$ 은 입력에 등장하는 슬롯의 개수이다. 구해진 슬롯 에러율을 문장 스코어  $\mathcal{L}(\theta)$ 과 합하여 새로운 스코어  $\mathcal{L}(\theta)_{all}$ 를 구한다. 본 실험에서  $\lambda$ 는 0.5로 고정하였다.

### 5. 실험 및 결과

본 논문에서 제안한 모델을 기존의 모델들과 비교하기 위해 [1]의 저자가 github[7]에 올려놓은 성능 표와 성능 측정기를 활용하였다.

본 실험에서 사용한 하이퍼 파라미터는 표 5와 같다. 단어 표현(word embedding)은 google의 word2vec[8]을 사용해 사전 학습 시킨 후 사용했으며, 모든 히든 레이어의 차원을 일치시켜 학습하였다. Dropout[9]은 과적합(over-fitting)을 방지하는 기술로 입력, 출력, GRU 내부에 적용되었다.

하이퍼 파라미터	값(차원, 확률)
단어 표현 (word embedding)	200
히든 레이어	[30, 50, 80, 100, 120, 150]
Dropout[9]	[0.0, 0.2, 0.5]

표 5. 하이퍼 파라미터

표 6, 7은 성능을 나타낸다. 성능 측정에는 스코어가 가장 높은 문장 5개를 원문과 비교하는 Top-5 BLEU 스코어를 이용하였다. SC-GRU enc-dec가 본 논문에서 실험한 모델이다. 비교 실험에 사용된 기존 모델을 설명하자면, 의미표현을 보고 규칙 기반으로 문장을 생성한 모델이 hdc(handcrafted)이며, knn은 k-nearest neighborhood, N-gram은 [6]에서 실험한 class-LM이다. 그 아래부터는

신경망 모델의 성능으로, Enc-Dec는 DA\_Cell이 들어가지 않은 LSTM기반 encoder-decoder 모델이고 H-LSTM은 SC-LSTM과 비슷하나 목적 함수로 DA\_Cell을 조절하지 않고 해당 슬롯이 출력에 등장하면 즉시 0으로 값을 바꿔버리는 휴리스틱한 방법을 쓴 모델이다.

모델	BLEU	SLOT_ERROR
HDC[1,7]	0.4260	0.00%
KNN[1,7]	0.5943	0.60%
N-gram[1,7]	0.6422	8.73%
Enc-Dec[1,7]	0.7398	2.78%
H-LSTM[1,7]	0.7466	0.74%
SC-LSTM[1,7] (baseline)	0.7525	<b>0.38%</b>
RA-LSTM[3]	<b>0.7789</b> (+0.0264)	<b>0.16%</b> (-0.22%)
SC-GRU enc-dec(Our)	0.7570 (+0.0045)	3.51% (+3.13%)

표 6. SF Restaurant 성능

모델	BLEU	SLOT_ERROR
HDC[1,7]	0.5406	0.00%
KNN[1,7]	0.6745	1.75%
N-gram[1,7]	0.7700	5.87%
Enc-dec[1,7]	0.8549	4.69%
H-LSTM[1,7]	0.8504	2.67%
SC-LSTM[1,7] (baseline)	0.8482	3.07%
RA-LSTM[3]	<b>0.8981</b> (+0.0499)	<b>0.43%</b> (-2.64%)
SC-GRU enc-dec(Our)	0.8645 (+0.0163)	2.12% (-0.95%)

표 7. SF Hotel 성능

SF Hotel의 경우, SC-GRU enc-dec 모델이 슬롯 에러율도 줄고 BLEU값도 올랐으나, SF Restaurant의 경우 BLEU 값만 상승이 있었고 슬롯 에러율은 높아졌다. 이는 word embedding과 GRU를 이용한 의미 표현의 추상적인 압축이 의미가 있음을 나타낸다. 다만 데이터에 따라 슬롯 에러율이 높아지는데 이는 식 (8)에서 의미 표현이 가지는 정보를 고려하지 않고 SLOT 값만을 기준으로 제약을 주므로 생기는 한계라 볼 수 있다. 이러한 제약에서 벗어나 의미 표현만으로 SLOT 값을 조절하는 RA-LSTM은 BLEU 값이 높음에도 불구하고 낮은 슬롯 에러율을 가진다.

데이터 셋에 따라 성능차이가 있는데 이는 KNN 성능에서 그 이유를 찾을 수 있다. 데이터 내에서 가까운 거리에 있는 문장을 찾아 답을 주는 KNN 방법은 데이터 내에서 같은 문장이 반복될수록 높은 성능을 보인다. 따라서 hotel 데이터가 restaurant 데이터에 비해 같은 문장이 반복되는 쉬운 데이터라 생각할 수 있다.

데이터	생성된 문장 (의미표현 / 생성문장)
SF Hotel	inform SLOT_NAME the carriage inn SLOT_DOGSALLOWED no </s>
	SLOT_NAME does not allow dogs </s>
SF Hotel	?request SLOT_NEAR </s>
	what area are you looking for </s>
SF Restaurant	inform SLOT_NAME taiwan restaurant SLOT_PRICERANGE cheap SLOT_FOOD taiwanese SLOT_GOODFORMEAL lunch </s>
	SLOT_NAME serves SLOT_PRICERANGE SLOT_FOOD food and is good for SLOT_GOODFORMEAL </s>
SF Restaurant	?select SLOT_PRICERANGE expensive or cheap </s>
	do you want a SLOT_PRICERANGE restaurant </s>

표 8. 생성 문장 예시

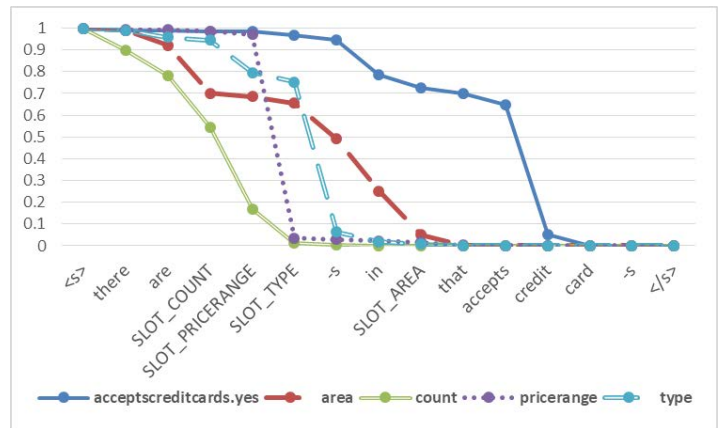


그림 4. SF Hotel의 DA\_Cell 변화

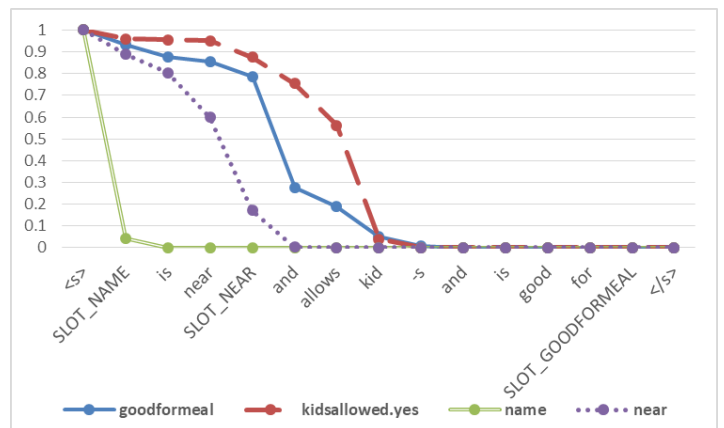


그림 5. SF Restaurant의 DA\_Cell 변화

표 8은 생성된 문장의 예시를 보여주고, 그림 4와 5는 DA Cell 내부 값들의 변화를 보여준다. 그림 4의 경우 모든 출력에 대해서 DA\_Cell이 잘 작동하는걸 볼 수 있다. 예를 들어 호텔이 카드를 받는다는 슬롯인 acceptcreditcards.yes 슬롯의 경우 맨 끝 card가 생성되기 전까지 값이 유지된다. 반면 레스토랑 데이터인 그림 5의 경우 다른 슬롯은 잘 동작하나 goodformeal 슬롯은 해당 슬롯이 생성되기 전에 값이 0으로 바뀌어 버렸다. 그러나 값이 0이 됨에도 불구하고 끝에서 생성이 된 걸로 보아 의미표현을 GRU로 압축한 값 c가 충분히 유의미함을 보여준다.

## 6. 결론

본 논문에서는 SC-GRU 기반 encoder-decoder를 이용한 자연어 생성 모델을 제안하였다. 실험결과, 본 논문에서 제안한 SC-GRU enc-dec모델이 베이스 라인보다 우수한 성능을 보였으며, One-hot 형태의 벡터와 성긴(dense) 형태의 벡터로 된 정보를 같이 사용하면 성능이 향상됨을 알 수 있었다. 그러나 현재 최고 성능을 보여주는 RA-LSTM에 비해 낮은 성능과 높은 슬롯 에러율을 가지는 점에서 의미를 고려하지 않은 의미 표현의 제약은 한계가 있음을 보였고 향후에는 이러한 문제를 해결하기 위해 연구할 예정이다.

## 감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2016R1C1B1014124)

## 참고문헌

- [1] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, Steve Young, "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems." CoRR, Vol.abs/1508.01745, 2015.
- [2] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259, 2014.
- [3] V.-K. Tran, L.-M. Nguyen, "Natural Language Generation for Spoken Dialogue System using RNN Encoder-Decoder Networks", CoNLL 2017
- [4] Remi Lebret, David Grangier, Michael Auli, "Neural Text Generation from Structured Data with Application to the Biography Domain." CoRR, Vol.abs/1603.07771, 2016.
- [5] 김건영, 이창기 "Sequence-to-sequence 모델을 이용한 자연어생성", 한국정보과학회 학술발표논문집, Vol.2017, No.06, pp.624-626, 2017.
- [6] Alice H. Oh and Alexander I. Rudnicky, "Stochastic language generation for spoken dialogue systems.", Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3, pp.27-32, 2000.
- [7] <https://github.com/shawnwun/RNNLG>
- [8] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", CoRR, Vol.abs/1301.3781, 2013.
- [9] Nitish Srivastava, Geoffrey Hinton and Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Vol.2014, pp.1929-1958, 2014".