

# 생성적 적대적 신경망을 이용한 생성기반 멀티턴 챗봇

김진태<sup>○</sup>, 김학수, 권오욱<sup>◆</sup>, 김영길<sup>◆</sup>

강원대학교 IT대학 컴퓨터정보통신공학전공<sup>○</sup>, 한국전자통신연구원<sup>◆</sup>  
{lsxo1119, nlpdrkim}@kangwon.ac.kr, {ohwoog, kimyk}@etri.re.kr

## Generative Multi-Turn Chatbot Using Generative Adversarial Network

Jintae Kim<sup>○</sup>, Harksoo Kim, Oh-Woog Kwon<sup>◆</sup>, Young-Gil Kim<sup>◆</sup>

Kangwon National University Computer and Communication Engineering<sup>○</sup>  
Electronics and Telecommunications Research Institute<sup>◆</sup>

### 요 약

기존의 검색 기반 챗봇 시스템과 다르게 생성 기반 챗봇 시스템은 사전에 정의된 응답에 의존하지 않고 채팅 말뭉치를 학습한 신경망 모델을 사용하여 응답을 생성한다. 생성 기반 챗봇 시스템이 사람과 같이 자연스러운 응답을 생성하려면 이전 문맥을 반영해야 할 필요가 있다. 기존 연구에서는 문맥을 반영하기 위해 이전 문맥과 입력 발화를 통합하여 하나의 벡터로 표현했다. 이러한 경우 이전 문맥과 입력 발화가 분리되어 있지 않아 이전 문맥이 필요하지 않는 경우 잡음으로 작용할 수 있다. 본 논문은 이러한 문제를 해결하기 위해 입력 발화와 이전 문맥을 각각의 벡터로 표현하는 방법을 제안한다. 또한 생성적 적대적 신경망을 통해 챗봇 시스템을 보강하는 방법을 제안한다. 채팅 말뭉치(55,000 개의 학습 데이터, 5,000 개의 검증 데이터, 5,260 개의 평가 데이터)를 사용한 실험에서 제안한 문맥 반영 방법과 생성적 적대적 신경망을 통한 챗봇 시스템 보강 방법은 BLEU와 임베딩 기반 평가의 성능 향상에 도움을 주었다.

주제어: 챗봇 시스템, 생성적 적대적 신경망, 문맥 반영 방법

### 1. 서론

챗봇 시스템은 사람과 컴퓨터가 자연어를 통해 대화하는 것을 말한다. 방법에 따라 검색 기반 모델과 생성 기반 모델로 구분할 수 있다. 검색 기반 모델은 미리 정의된 발화-응답 쌍의 데이터에서 입력된 문장과 유사한 발화와 쌍을 이루는 응답을 출력하는 모델이다. 미리 정의된 응답을 출력하기 때문에 문법적 오류가 적게 나타나는 장점이 있지만 입력된 문장이 미리 정의된 발화와 유사하지 않다면 적합한 응답을 못하는 문제가 발생한다. 생성 기반 모델은 대화 말뭉치가 학습된 신경망 모델을 사용해 새로운 응답을 생성하는 모델이다. 생성 기반 모델은 새로운 입력 문장에 대해 유연하게 응답을 생성하는 장점이 있지만 “응”과 같은 보편적인 응답을 생성하는 문제가 발생한다. 최근 심층 신경망(Deep Neural Network)의 발달로 다양한 생성 기반 모델에 대한 연구가 진행되고 있다. 더 나아가 상황에 적합한 응답을 생성하기 위해서 입력 문장 이전의 문맥을 고려해야 한다. 기존 연구에서는 이전 문맥을 반영하기 위해서 이전 문맥과 입력 문장을 통합하여 하나의 벡터로 표현했다. 대화 중 주제가 변하는 상황에서 이전 문맥은 잡음으로 작용할 수 있다. 이러한 이유로 이전 문맥과 입력 문장이 각각의 벡터로 표현되어야 한다. 본 논문에서는 이러한 문제를 해결하기 위해 이전 문맥과 입력 문장을 각각 인코딩하는 방법을 제안한다. 그리고 챗봇 시스템을 생성적 적대적 신경망(Generative Adversarial Network)[1]구조로 보강한 한국어 챗봇 시스템을 제안한다.

### 2. 관련 연구

기존의 생성 기반 챗봇 시스템은 문맥을 반영하지 않고 입력 문장에 대한 응답을 생성했다[2-4]. 하지만 문맥을 반영하지 않으면 문맥상 적합하지 않은 응답을 생성하는 문제가 발생한다. 이를 개선하기 위해 [5-7]은 이전 문맥과 입력 문장을 하나의 문맥 벡터(Context Vector)로 인코딩하고 인코딩한 벡터를 응답 생성에 사용했다. 응답을 생성할 때 문맥 정보가 필요하지 않은 경우 문맥 정보는 잡음으로 작용될 수 있으므로 이를 개선하는 다른 방법이 필요하다.

생성 기반 챗봇 시스템은 사용자 입력에 대해 보편적인 응답을 생성하는 문제가 있다. 그러므로 다양한 문장을 생성하는 방법이 필요하다. [8]에서 변형 오토인코더(Variational Auto-Encoder)[9]를 통해 학습된 잠재 변수(Latent Variable)가 다양한 문장 생성에 도움을 주는 것을 확인했다. 이 연구 결과를 바탕으로 [5-6]은 생성 기반 챗봇 시스템이 다양한 응답을 생성하도록 변형 오토인코더를 사용하였다. 그러나 다양한 응답을 생성하는데 도움을 줬지만 여전히 보편적인 응답을 생성하는 문제가 발생한다. [7]은 이 문제를 해결하기 위해 생성적 적대적 신경망을 사용하여 잠재 변수를 학습하였다.

자연어 생성에서 생성적 적대적 신경망은 하나의 단어만 잘못되어도 의미가 바뀌어 문장을 생성하기에 부적합하다고 알려져 있지만 생성적 적대적 신경망을 이용한 문장 생성에 대한 학습 방법[10-11], 생성적 적대적 신경망을 이용한 문장 생성에 적합하도록 순환 신경망(Recurrent Neural Network)의 구조를 변형하는 방법[12], 생성적 적대적 신경망과 강화 학습을 통합하여 응답을 생성하는 모델[13]등 성과를 보이는 연구가 진행되어 사용되기 시작하였다. 본 논문은 상황에 따라 문맥 정보와 입력 문장의 반영 비율을 조정하는 방법을 제안하고 생

성적 적대적 신경망을 통해 보강한 한국어 챗봇 시스템을 제안한다.

### 3. 생성적 적대적 신경망을 이용한 생성기반 멀티턴 챗봇

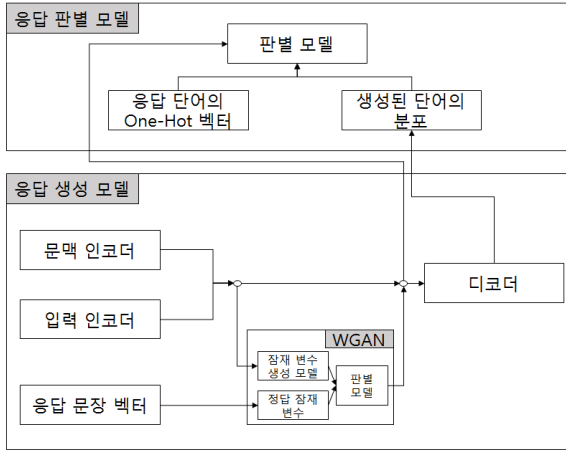


그림 1 제안 모델의 전체 구조도

그림 1은 제안 모델의 전체 구조도를 보여준다. 본 논문에서 제안한 챗봇 시스템은 응답 생성 모델과 응답 판별 모델로 구성된다. 응답 생성 모델은 인코더-디코더 모델로 구성된다. 기존 연구에서 사용된 문맥 반영 방법과 다르게 입력 문장과 이전 문맥을 개별적인 인코더로 구성한다. 문맥 인코더는 이전 문맥으로부터 문맥 벡터를 생성하고 입력 인코더는 입력 문장으로부터 입력 벡터를 생성한다. 잠재 변수는 생성된 벡터들을 입력받아 생성적 적대적 신경망을 통해 학습한다. 생성된 입력 벡터와 문맥 벡터, 잠재 변수를 결합(Concatenate)하여 인코딩 벡터를 생성하고 디코더를 통해 문장을 생성한다. 응답 판별 모델은 단어의 분포로 디코딩 벡터를 생성하고 입력된 인코딩 벡터와 함께 적합한 출력인지 판별한다.

#### 3.1 응답 생성 모델

그림 2는 인코더-디코더로 구성된 응답 생성 모델의 상세 구조도이다. 본 논문에서 제안하는 응답 생성 모델

은 문맥 반영 단계와 잠재 변수 단계, 디코더 단계로 구성된다. 문맥 반영 단계에서는 입력 문장과 이전 문맥을 인코딩한다. 잠재 변수 단계에서 다양한 응답 생성을 위한 잠재 변수를 학습한다. 디코더 단계에서 문맥을 반영하여 입력 문장에 대한 적합한 응답을 생성한다.

##### 3.1.1 문맥 반영 단계

입력 문장에 따라 응답 생성에 필요한 문맥 벡터와 입력 벡터의 반영 수준이 다르다. 특정 입력 문장이 입력된 경우 응답 생성을 위해 문맥 정보가 필요하지 않은 경우가 있고, 응답 생성에 필요한 정보가 문맥에 포함되어 있는 경우도 있다. 본 논문은 이러한 상황을 반영하고자 응답 생성 모델에서 입력 문장과 이전 문맥을 각각 인코딩하고 게이트(Gate)를 통해 조절한다. 입력 문장이 문맥의  $k$  번째 일 때 입력 인코더는  $k$  번째 발화를 인코딩하여 입력 벡터  $i$ 를 생성한다.  $i$ 는 식 1의 방법을 사용해 생성하고 반영 비율을 조절한다.

$$\begin{aligned} i &= BiEnc(w_1, w_2, \dots, w_n) \\ gate &= \sigma(W_g i + b_g) \\ transfer &= ReLU(W_t i + b_t) \\ I &= gate * transfer \end{aligned} \quad (1)$$

식 1의  $w_n$ 은 입력 문장의  $n$ 번째 단어를 의미한다.  $BiEnc$ 는 양방향 순환 신경망[14]을 GRU(Gate Recurrent Unit)[15]로 설정한 입력 인코더를 의미한다.  $i$ 는 입력 문장의 단어열을 입력 인코더의 입력으로 사용해 인코딩된 입력 벡터이다.  $gate$ 는 활성화 함수  $sigmoid$ 를 사용하여 0~1 사이의 값을 가진다.  $transfer$ 는 활성화 함수  $ReLU$ 를 사용하여  $gate$ 를 통과하기 위한 값을 가진다.  $I$ 는  $gate$ 와  $transfer$ 를 곱한 값으로  $i$ 의 반영 비율이 조절된 입력 벡터를 의미한다. 문맥 벡터는 문맥의  $k$  번째 이전의 각 발화를 발화 벡터로 생성하는 발화 인코더와 문맥 벡터를 생성하는 문맥 인코더로 구성된다. 문맥 인코더는  $k$  번째 이전의 발화 벡터들을 입력 받아 문맥 벡터  $c$ 로 생성한다.  $c$ 는 식 2의 방법을 사용해 생성하고 반영 비율을 조절한다.

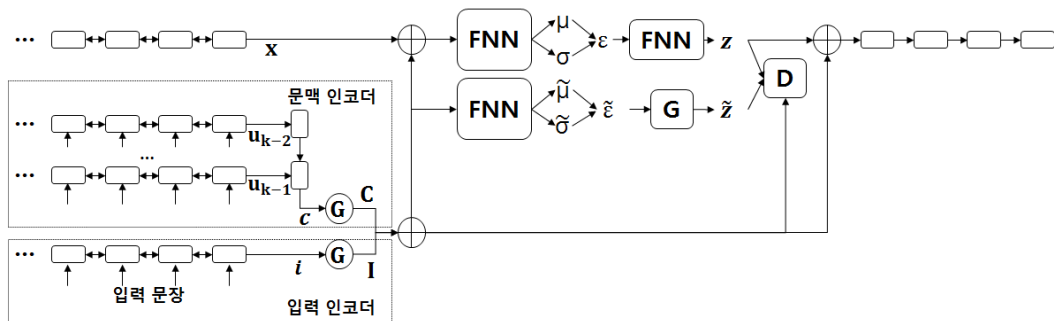


그림 2 응답 생성 모델

$$\begin{aligned}
U_{k-1} &= BiEnc(w_1^{k-1}, w_2^{k-1}, \dots, w_n^{k-1}) \\
c &= Enc(U_1, U_2, \dots, U_{k-1}) \\
gate &= \sigma(W_g c + b_g) \\
transfer &= ReLU(W_t c + b_t) \\
C &= gate * transfer
\end{aligned} \quad (2)$$

식 2의  $w_n^{k-1}$ 은 문맥의  $k-1$ 번째 발화의  $n$ 번째 단어를 의미한다.  $BiEnc$ 는 양방향 순환 신경망을 GRU로 설정한 발화 인코더를 의미한다.  $U_{k-1}$ 는 발화 인코더가 문맥의  $k-1$ 번째 발화를 인코딩한 발화 벡터이다.  $Enc$ 는 순환 신경망을 GRU로 설정한 문맥 인코더를 의미한다.  $c$ 는 이전 문맥의 발화 벡터들을 문맥 인코더의 입력으로 사용해 인코딩된 문맥 벡터이다.  $gate$ 는 활성화 함수  $sigmoid$ 를 사용하여 0~1 사이의 값을 가진다.  $transfer$ 는 활성화 함수  $ReLU$ 를 사용하여  $gate$ 를 통과하기 위한 값을 가진다.  $C$ 는  $gate$ 와  $transfer$ 를 곱한 값으로  $c$ 의 반영 비율이 조절된 문맥 벡터를 의미한다. 반영 비율이 조절된  $I$ 와  $C$ 를 입력 벡터와 문맥 벡터로 사용한다.

### 3.1.2 잠재 변수 학습 단계

잠재 변수는 다양한 문장 생성을 할 수 있게 한다고 알려져 있다[8]. 본 논문에서는 [7]의 방법을 사용해 잠재 변수를 학습한다. 그림 2의  $z$ 는 응답에서 추출된 응답 잠재 변수이다. 식 3의 방법을 사용해 생성한다.

$$\begin{aligned}
v &= [x; C; I] \\
\begin{pmatrix} \mu \\ \log \sigma^2 \end{pmatrix} &= FNN(v) \\
\epsilon &\sim N(\epsilon; \mu, \sigma^2) \\
z &= FNN(\epsilon)
\end{aligned} \quad (3)$$

식 3의  $x$ 는 응답을 양방향 순환 신경망을 GRU로 설정한 모델로 인코딩한 응답 벡터이다.  $v$ 는  $x$ 와  $C$ ,  $I$ 를 결합한 벡터를 의미한다.  $v$ 를 전방 전달 신경망(FeedForward Neural Network)의 입력으로 사용해  $\mu$ 와  $\log \sigma^2$ 을 구한다. 그 다음  $\mu$ 와  $\sigma^2$ 를 통해 가우시안 잡음  $\epsilon$ 를 생성한다. 생성된  $\epsilon$ 를 전방 전달 신경망의 입력으로 사용해  $z$ 를 생성한다. 그림 2의  $\tilde{z}$ 은 잠재 변수 생성 모델  $G$ 에서 생성한 추정 잠재 변수이다. 식 4의 방법을 사용해 생성한다.

$$\begin{aligned}
\tilde{v} &= [C; I] \\
\begin{pmatrix} \tilde{\mu} \\ \log \tilde{\sigma}^2 \end{pmatrix} &= FNN(\tilde{v}) \\
\tilde{\epsilon} &\sim N(\epsilon; \tilde{\mu}, \tilde{\sigma}^2) \\
\tilde{z} &= G(\tilde{\epsilon})
\end{aligned} \quad (4)$$

식 4의  $\tilde{v}$ 는  $C$ 와  $I$ 를 결합한 벡터를 의미한다.  $\tilde{v}$ 를 전방 전달 신경망의 입력으로 사용해  $\tilde{\mu}$ 와  $\log \tilde{\sigma}^2$ 을 구한다. 그 다음  $\tilde{\mu}$ 와  $\tilde{\sigma}^2$ 를 통해 가우시안 잡음  $\tilde{\epsilon}$ 를 생성한다.

다. 생성된  $\tilde{\epsilon}$ 를 잠재 변수 생성 모델  $G$ 의 입력으로 사용해  $\tilde{z}$ 를 생성한다. 잠재 변수 판별 모델  $D$ 는  $C$ 와  $I$ 를 결합한 벡터와 잠재 변수를 입력 받는다.  $C$ 와  $I$ 가 결합한 벡터가 입력될 때 같이 입력된 잠재 변수가  $z$ 인지  $\tilde{z}$ 인지 판별한다.  $G$ 와  $D$ 는 생성적 적대적 신경망의 한 종류인 WGAN(Wasserstein Generative Adversarial Networks)[16]을 통해  $z$ 와 유사한 최적의  $\tilde{z}$ 를 생성하도록 학습된다.  $G$ 와  $D$ 는 전방 전달 신경망 모델을 사용한다.

### 3.1.3 디코더 단계

디코더는 잠재 변수와  $C$ ,  $I$ 를 결합한 인코딩 벡터를 받아 응답을 생성한다. 학습 단계에서  $\tilde{z}$ 는 학습이 진행되고 있으므로 응답 생성에 필요한 잠재 변수로 적합하지 않다. 그러므로 학습 단계에 적합한 잠재 변수인  $z$ 를 사용해 학습한다. 예측 단계에서는 학습된  $\tilde{z}$ 를 사용하여 응답을 생성한다. 디코더는 순환 신경망을 GRU로 설정한 모델을 사용한다.

### 3.2 응답 판별 모델

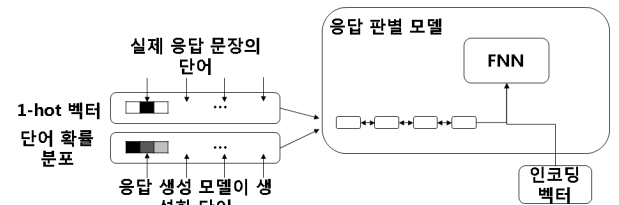


그림 3 응답 판별 모델

그림 3은 응답 판별 모델의 구조도이다. [12]와 같이 실제 응답 문장의 단어를 one-hot 벡터로 표현한 값을 정답 단어 확률 분포라 가정하고, 응답 생성 모델이 생성한 단어 분포를 확률로 변환한 분포를 추정 단어 확률 분포라 가정한다. 응답 판별 모델은 인코딩 벡터와 단어 확률 분포를 입력받는다. 응답 판별 모델은 단어 확률 분포를 입력받아 디코딩 벡터를 생성한다. 디코딩 벡터는 양방향 순환 신경망을 GRU로 설정한 모델을 통해 생성한다. 생성된 디코딩 벡터와 인코딩 벡터를 전방 전달 신경망의 입력으로 사용해 입력된 단어 확률 분포가 추정 단어 확률 분포인지 판별한다. 응답 생성 모델은 교차 엔트로피(Cross Entropy)를 이용하여 학습한다. 추가적으로 응답 생성 모델과 응답 판별 모델간의 관계를 WGAN 통해 학습하여 응답 생성 모델의 성능을 향상시킨다. 즉, 응답 생성 모델에서의 WGAN은 잠재 변수간의 대립이 되고 전체 모델의 WGAN은 응답 생성 모델과 응답 판별 모델 간의 대립이 되어 학습한다.

## 4. 실험 및 평가

### 4.1 데이터 구성

본 논문의 실험을 위해 33,254개의 담화를 포함한 담

화 말뭉치를 사용한다. 답화 말뭉치는 문맥이 포함되지 않은 27,676개의 답화와 문맥이 포함된 5,578개의 답화로 구성된다. 문맥이 포함된 답화의 평균 길이는 7.73이다. 데이터의 수를 확장하기 위해 본 논문에서 제안한 챗봇 시스템의 이전 문맥 길이를 2로 고정하고 문맥이 포함된 5,578개의 답화를 그림 4와 같은 방법을 사용해 확장한다. 확장된 데이터와 기존의 문맥이 포함되지 않은 데이터를 통합한 전체 데이터의 수는 65,260개이다. 이 중 55,000개는 학습 데이터 5,000개는 검증 데이터, 5,260개는 평가 데이터로 사용한다.

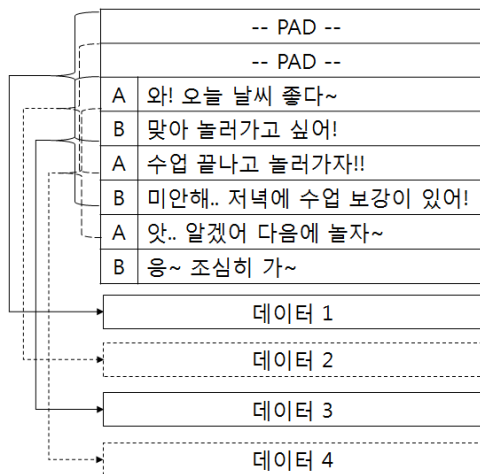


그림 4 데이터 확장 방법 예시

## 4.2 실험 방법

실험은 4가지의 모델을 실험했다. 기존 연구의 문맥 반영 방법을 사용한 Baseline과 발화와 이전 문맥을 분리한 개별 인코딩 모델, 본 논문에서 제안하는 문맥 반영 방법인 개별 인코딩 모델에 게이트를 적용한 제안 문맥 반영 모델, 본 논문에서 제안하는 챗봇 시스템인 WGAN을 통해 보강한 제안 모델이다. 4가지 모델은 모두 동일한 단어 임베딩(Word Embedding)을 사용하였고 뉴스 기사와 위키백과 문서가 포함된 약 28GB 문서를 사용해 학습된 GloVe[17] 300차원 벡터를 사용했다. 본 논문에서 제안한 챗봇 시스템을 평가하기 위해 단어 중복 기반 평가와 임베딩 기반 평가를 진행한다. 단어 중복 기반 평가는 BLEU[18]와 BLEU에 스무딩7(Smoothing7)방법이 적용된 BLEU-Smoothing7[19]을 성능 평가에 사용한다. BLEU는 기계 번역 모델의 성능을 평가하기 위해 고안된 평가 방법이다. BLEU-Smoothing7은 BLEU를 문장 기반 평가에 적합하도록 변형한 평가 방법이다. 임베딩 기반 평가는 [20]의 임베딩 평균 방법, [21]의 임베딩 그리디(Greedy)방법, [22]의 임베딩 극(Extrema)방법을 사용하여 평가한다. 임베딩 기반 평가는 정답 응답과 챗봇 시스템이 생성한 응답의 단어 임베딩을 통해 계산된다. 사용된 단어 임베딩은 챗봇 시스템 학습에 사용된 단어 임베딩과 동일하다.

## 4.3 단어 중복 기반 평가 결과

각 평가 데이터의 입력에서 모델별로 10개의 응답 표본을 얻고 BLEU점수와 BLEU-Smoothing7점수를 측정했다. [7]의 성능 평가와 같이 10개의 응답 중 가장 높은 점수를 재현율(Recall), 10개의 응답 평균 점수를 정확률(Precision)으로 정의했다.

표 1 BLEU 점수 측정 결과

모델	BLEU 점수		
	재현율	정확률	F1 점수
Baseline	0.4363	0.3636	0.3966
개별 인코딩 모델	0.6086	0.3891	0.4747
제안 문맥 반영 모델	<b>0.6187</b>	0.3905	0.4788
제안 모델	0.6105	<b>0.4001</b>	<b>0.4834</b>

표 1은 BLEU 점수 측정 결과이다. 제안 문맥 반영 모델은 가장 높은 재현율을 보이고 본 논문에서 제안한 제안 모델은 가장 높은 정확률과 F1 점수를 보였다.

표 2 BLEU-Smoothing7 점수 측정 결과

모델	BLEU-Smoothing7 점수		
	재현율	정확률	F1 점수
Baseline	0.1950	0.1637	0.1780
개별 인코딩 모델	0.2954	0.1945	0.2346
제안 문맥 반영 모델	<b>0.3034</b>	0.1978	0.2395
제안 모델	0.2994	<b>0.2018</b>	<b>0.2411</b>

표 2은 BLEU-Smoothing7 점수 측정 결과이다. 제안 문맥 반영 모델은 가장 높은 재현율을 보이고 제안 모델은 가장 높은 정확률과 F1 점수를 보였다.

## 4.4 임베딩 기반 평가 결과

각 평가 데이터의 입력에서 모델별로 10개의 응답 표본을 얻고 각 임베딩 기반 점수를 측정했다. 4.3절의 평가 방법과 같이 10개의 응답 중 가장 높은 점수를 재현율, 10개의 응답 평균 점수를 정확률로 정의했다.

표 3 임베딩 평균 점수 측정 결과

모델	임베딩 평균 점수		
	재현율	정확률	F1 점수
Baseline	0.8452	0.8113	0.8279
개별 인코딩 모델	0.8903	0.8344	0.8615
제안 문맥 반영 모델	<b>0.8905</b>	0.8367	0.8628
제안 모델	0.8880	<b>0.8401</b>	<b>0.8634</b>

표 3은 [20]에서 제안한 임베딩 기반 평가 방법인 임베딩 평균 점수 측정 결과이다. 임베딩 기반 점수에서도 문맥과 입력 문장을 한번이 인코딩한 Baseline보다 분리하여 인코딩한 모델의 성능이 높다. 제안 문맥 반영 모

델이 가장 높은 재현율을 보였다. 제안 모델은 가장 높은 정확률과 F1 점수를 보였다.

표 4 임베딩 그리디(Greedy) 점수 측정 결과

모델	임베딩 그리디 점수		
	재현율	정확률	F1 점수
Baseline	0.6370	0.5913	0.6133
개별 인코딩 모델	<b>0.7071</b>	0.6186	<b>0.6599</b>
제안 문맥 반영 모델	0.7065	0.6178	0.6592
제안 모델	0.7000	<b>0.6196</b>	0.6574

표 4은 [21]에서 제안한 임베딩 기반 평가 방법인 임베딩 그리디 점수 측정 결과이다. 임베딩 그리디 점수에서는 개별 인코딩 모델이 가장 높은 재현율과 F1 점수를 보였다. 제안 모델은 가장 높은 정확률을 보였다.

표 5 임베딩 극(Extrema) 점수 측정 결과

모델	임베딩 극 점수		
	재현율	정확률	F1 점수
Baseline	0.6025	0.5658	0.5836
개별 인코딩 모델	<b>0.6575</b>	<b>0.5738</b>	<b>0.6128</b>
제안 문맥 반영 모델	0.6542	0.5700	0.6092
제안 모델	0.6446	0.5697	0.6048

표 5은 [22]에서 제안한 임베딩 기반 평가 방법인 임베딩 극 점수 측정 결과이다. 임베딩 극 점수에서는 개별 인코딩 모델이 전체적으로 가장 높은 점수를 보였다.

#### 4.5 실험 결과 분석

본 논문에서 제안한 문맥 반영 방법을 사용한 제안 문맥 반영 모델이 기존의 문맥 반영 방법을 사용한 Baseline보다 전체적인 성능에서 우수한 결과를 보였다. 그리고 제안 모델은 많은 성능 측정 부분에서 가장 높은 성능을 보였다. 추가적으로 본 논문에서 제안한 모델과 다른 방법이 어떤 차이를 보이는지 예제를 통해 보여준다.

표 6 실험 결과 예제 1

A	소 화제 이 라도 먹 어 ~
B	그렇 어야 하 겠 네
A	따뜻 하 니 물 많이 마시 고 ~
Baseline	그렇 니 것 이 다 !
	그렇 니가 ... 이 다 ~
개별 인코딩 모델	그래 알 았 어
	그래 그래 ~
제안 문맥 반영 모델	그래 물 많이 마시 어야 하 겠 네
	그래 알 았 어 ~
제안 모델	그래 알 았 어 그러 르게 !
	그래 물 많이 마시 어야 하 겠 다 ~

표 6은 소화와 관련된 대화에 대한 모델의 응답 예제이다. 각 모델에서 2개의 응답을 생성했다. Baseline 모델이 생성한 응답은 모두 적합하지 않다. 나머지 모델이 생성한 응답은 모두 적합하다.

표 7 실험 결과 예제 2

A	오늘 아침 부터 갑자기 기침 을 너무 많이 하 아 .
Baseline	감기 걸리 면 낮 지 .
	감기 걸리 면 무시 받 겠 아 .
개별 인코딩 모델	그럼 내일 꼭 쉬 어야 하 아 .
	큰일 날 뻔 하 니 일 이 니데 ?
제안 문맥 반영 모델	감기 걸리 는 것 이 중요 하 니 거 이 야
	감기 어서 낫 길 바라 니다 .
제안 모델	감기 걸리 니 적 있 니 ?
	감기 걸리 었 나 ?

표 7은 감기와 관련된 대화에 대한 모델의 응답 예제이다. 각 모델에서 2개의 응답을 생성했다. Baseline 모델이 생성한 두 응답은 모두 적합하지 않다. 개별 인코딩 모델이 생성한 두 응답중 하나는 적합하지만 다른 하나는 적합하지 않다. 제안 문맥 반영 모델이 생성한 두 응답중 하나는 적합하지만 다른 하나는 적합하지 않다. 제안 모델이 생성한 두 응답 모두 적합하다.

#### 5. 결론

본 논문은 이전 문맥과 입력 문장을 각각 인코딩하는 방법을 제안하고 WGAN을 이용하여 챗봇 시스템을 보강하는 방법을 제안했다. 실험 결과 이전 문맥과 입력 문장을 통합하여 인코딩 하는 것 보다 본 논문에서 제안한 각각 인코딩 하는 방법이 더 높은 성능을 보였다. 또한 WGAN을 이용해 챗봇 시스템을 보강하는 방법은 많은 성능 측정에서 효과적인 것을 보였다.

#### 감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. ( R0126-15-1117, 언어학습을 위한 자유발화형 음성 대화처리 원천기술 개발). 또한 2017년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017M3C4A7068188)

#### 참고문헌

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *Advances in neural information processing systems*, 2014
- [2] 김시형, 김학수, “의사 형태소 단위 채팅 시스템” ,

- 제 28회 한글 및 한국어 정보처리 학술대회 논문집, pp. 263-267, 2016
- [3] 김진태, 김시형, 김학수, 이연수, 최맹식, “색인어 인코딩과 음절 디코딩에 기반한 생성 채팅 모델”, 제 29회 한글 및 한국어 정보처리 학술대회 논문집, pp.125-129, 2017
- [4] 김진태, 김학수, 최맹식, 이연수, 권오욱, 김영길, “비지도 사전 학습을 이용한 한국어 채팅 시스템”, 2018 한국컴퓨터종합학술대회 논문집, pp.631-633, 2018
- [5] X. Shen, H. Su, S. Niu and V. Demberg, Improving Variational Encoder-Decoders in Dialogue Generation, *arXiv preprint arXiv:1802.02032*, 2018
- [6] T. Zhao, R. Zhao and M. Eskenazi, Learning discourse-level diversity for neural dialog models using conditional variational autoencoders, *arXiv preprint arXiv:1703.10960*, 2017
- [7] X. Gu, K. Cho, J. Ha and S. Kim, DialogWAE: Multi modal Response Generation with conditional Wasserstein Auto-Encoder, *arXiv preprint arXiv:1805.12352*, 2018
- [8] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz and S. Bengio, Generating sentences from a continuous space, *arXiv preprint arXiv:1511.06349*, 2015
- [9] D. Kingma and M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114*, 2013
- [10] M. Moradshahi and U. Contractor, Language Modeling with Generative Adversarial Networks, *arXiv preprint arXiv:1804.02617*, 2018
- [11] O. Press, A. Bar, B. Bogin, J. Berant and L. Wolf, Language generation with recurrent generative adversarial networks without pre-training, *arXiv preprint arXiv:1706.01399*, 2017
- [12] S. Rajeswar, S. Subramanian, F. Dutil, C. Pal and A. Courville, Adversarial Generation of natural language, *arXiv preprint arXiv:1705.10929*, 2017
- [13] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter and D. Jurafsky, Adversarial learning for neural dialogue generation, *arXiv preprint arXiv:1701.06547*, 2017
- [14] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, pp. 2673-2681, 1997
- [15] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555*, 2014
- [16] M. Arjovsky, S. Chintala and L. Bottou, Wasserstein gan, *arXiv preprint arXiv:1701.07875*, 2017
- [17] J. Pennington, R. Socher and C. Manning, Glove: Global vectors for word representation, *Proceedings of the 2014 conference on empirical methods in natural language processing*, pp. 1532-1543, 2014
- [18] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, BLEU: a method for automatic evaluation of machine translation, *Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics*, pp. 311-318, 2002
- [19] B. Chen and C. Cherry, A systematic comparison of smoothing techniques for sentence-level bleu, *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 362-367, 2014
- [20] J. Wieting, M. Bansal, K. Gimpel and K. Livescu, Towards universal paraphrastic sentence embeddings, *arXiv preprint arXiv:1511.08198*, 2015
- [21] V. Rus and M. Lintean, A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics, *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, Association for Computational Linguistics*, pp. 157-162, 2012
- [22] G. Forgues, J. Pineau, J. M. Larcheveque and R. Tremblay, Bootstrapping dialog systems with word embeddings, *Proceedings of the Nips, modern machine learning and natural language processing workshop*, Vol 2, 2014