

# Copy Mechanism과 Input Feeding을 이용한 End-to-End 한국어 문서요약

최경호<sup>o</sup>, 이창기

강원대학교

{gangsparkle, leeck}@kangwon.ac.kr

## End-to-end Document Summarization using Copy Mechanism and Input Feeding

Kyoungcho Choi, Changki Lee  
Kangwon National University

### 요 약

본 논문에서는 Sequence-to-sequence 모델을 생성요약의 방법으로 한국어 문서요약에 적용하였으며, copy mechanism과 input feeding을 적용한 RNN search 모델을 사용하여 시스템의 성능을 높였다. 인터넷 신문 기사를 수집하여 구축한 한국어 문서요약 데이터 셋(train set 30291 문서, development set 3786 문서, test set 3705문서)으로 실험한 결과, input feeding과 copy mechanism을 포함한 모델이 형태소 기준으로 ROUGE-1 35.92, ROUGE-2 15.37, ROUGE-L 29.45로 가장 높은 성능을 보였다.

주제어: 딥러닝, Sequence to sequence, 문서요약, sentence summarization, RNN

### 1. 서론

문서요약(sentence summarization)은 주어진 문서로부터 문서의 핵심 내용을 보존하면서 보다 짧게 축약된 문서를 생성하는 것이다.

기존의 문서요약에 사용된 방법들은 크게 전체 문서의 내용을 압축하여 원문에 없는 새로운 문장을 생성하는 생성요약과 문서내의 문장들 중 중요하다고 판단되는 문장을 찾아내는 추출요약으로 구분 할 수 있다[1]. 그러나 추출요약은 문서 내에 문서 전체를 대표하는 문장이 없을 경우에 치명적 문제를 야기할 수 있다는 문제가 있다. 한국어를 바탕으로 한 기존 연구에서는 tf-idf를 바탕으로 한 확률 기반의 추출요약 방법이 많이 사용되어 왔다[2].

Sequence-to-sequence 모델은 기계번역에서 많이 연구되어온 모델로, RNN(Recurrent Neural Network)을 이용하여 입력 문장의 의미를 함축하는 context vector를 만들고(encoding), 이를 다시 다른 RNN을 통하여 출력 문장을 생성(decoding)하는 방법이다[3,4]. 따라서 Sequence-to-sequence 모델은 별도의 휴리스틱 등을 이용한 규칙과 자질들을 시스템에 추가할 필요가 없으며, 구문분석 전처리과정이 없는 end-to-end 방식으로 설계할 수 있다.

Attention mechanism은 Sequence-to-sequence 모델의 디코딩(decoding) 과정에서 각 시간별 출력을 예측할 때 입력 열에서 해당 출력을 결정하는데 필요하다고 판단되는 입력 시퀀스의 hidden state를 참조하여 사용하는 방법이다[5].

Copy mechanism은 Sequence-to-sequence 모델의 디코딩 과정에서 문장을 생성할 때 필요한 어휘가 출력 사전(output vocabulary)에 없는 문제(Out-of-Vocabulary)와 고유명사들의 출력 확률이 작아지는 문제를 해결하기 위해 고안된 방법으로, 출력에 필요한 어휘를 입력 열에서 찾아 출력 없이 복사(copy)하는 사용하는 방법이다[6]. Copy mechanism은 attention mechanism과 유사한 형태의 copy attention을 별도로 두어, 디코딩 과정에서 각 시간별 출력 어휘를 예측할 때 출력 사전에 있는 어휘들의 확률과 함께 입력 열 중에서 copy attention 점수가 가장 높은 어휘를 그대로 출력할 확률도 함께 계산한다.

본 논문에서는 copy mechanism 과 input feeding[5]을 포함한 Sequence-to-sequence 모델을 생성요약의 방법으로 한국어 문서요약에 적용하고 평가하여, 제안하는 모델이 규칙이나 전처리 과정 없이도 좋은 성능을 얻을 수 있음을 보인다.

## 2. 관련 연구

### 2.1 문서요약의 성능측정지표

요약문의 성능평가는 시스템 번역문과 정답 번역문의 일치여부가 중요한 기계번역과 다르게 시스템 요약문의 정답에서의 정보누락 여부가 중요하다. 이러한 이유로 문서요약에서는 ROUGE(Recall-Oriented Understudy for Gisting Evaluation)라는 성능지표를 주로 사용한다. ROUGE는 정답 요약문(reference summary)과 시스템 요약문(candidate summary)간의 재현율을 바탕으로 평가한다[7].

ROUGE는 다양한 변형이 존재한다. 본 논문에서는 N-gram을 기반의 ROUGE-N과 LCS(Longest Common Subsequence)기반의 ROUGE-L를 이용하여 시스템을 평가한다. ROUGE-N을 수식으로 표현하면 다음과 같다.

$$\frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

$Count_{match}(gram_n)$ 는 미리 제작된 정답 요약문인 *ReferenceSummaries*의 n-gram 들 중의 하나인  $gram_n$ 이 시스템 요약문으로 만들어진 n-gram에 포함되어 있으면 1, 그렇지 않으면 0을 반환한다. ROUGE-L을 수식으로 표현하면 다음과 같다.

$$R_{lcs} = \frac{LCS(X, Y)}{m}$$

$$P_{lcs} = \frac{LCS(X, Y)}{n}$$

$$F_{lcs} = \frac{2RP}{R + P}$$

X는 형태소(토큰) 단위로 분리된 시스템 요약문을 의미하고, Y는 형태소 단위로 분리된 정답 요약문을 의미한다.  $LCS(X, Y)$ 는 X와 Y의 최장 공통 부분수열(연속으로 일치하는 최장의 형태소 배열)의 길이를 뜻하고, m은 정답 요약문의 길이를 뜻하고, n은 시스템 요약문의 길이를 뜻한다.

### 2.2 영어권 문서요약 연구

영어권에서 문서요약은 DUC(Document Understanding Conference) 등의 컨퍼런스를 통해 지속적으로 연구되어 왔으며, 문서에 등장하는 문장 중 문서의 내용을 가장 잘 대표하는 문장을 찾아내는 추출요약 방법이 많이 사용되었다. 또한 topic modeling과 워드넷과 같은 시소러스를 활용하여 추출요약의 성능을 향상시키는 방법들이 연구되었다[8].

최근 딥러닝(deep learning)을 활용한 기계번역 모델인 Neural Machine Translation(NMT)[5,6]이 활발히 연구되면서, 다른 자연언어처리 분야에도 딥러닝을 적용하는 연구가 활발해지고 있다. 이에 따라 영어권에서는 NMT에 사용되는 Sequence-to-sequence 모델을 활용한 생성요약 기반의 문서요약 방법이 연구되고 있다[5].

### 2.3 한국어 문서요약 연구

한국어 문서요약 연구들은 tf-idf를 기반으로 한 확률 기반의 추출요약 방법들과, 휴리스틱을 동원하여 만든 규칙들을 이용하는 방법들을 사용해 왔다[1,2]. 최근에는 워드넷 등의 시소러스 자원을 이용하여 추출된 문장 내의 어휘를 바꾸어 문서의 내용을 더 잘 축약하게 만든 방법도 연구되었다.[2]

## 3. 문서요약 모델

### 3.1 RNN encoder-decoder 모델

RNN encoder-decoder 모델은 GRU(Gated Recurrent Unit), LSTM(Long Short-Term Memory)등의 RNN을 이용하여  $P(y|x)$ 를 최적화하는 모델이다[4]. RNN Encoder-decoder 모델은 입력 문장을 고정된 차원의 context vector로 인코딩한 후 이 context vector로부터  $P(y|x)$ 를 최대화 하는 출력 문장을 생성한다.

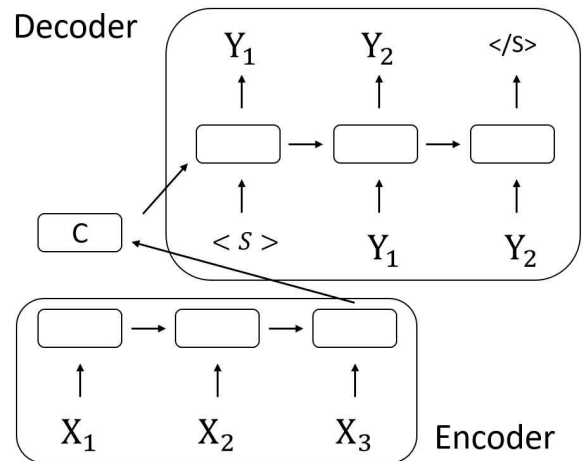


그림 1. RNN encoder-decoder 모델

### 3.2 RNN search 모델

RNN encoder-decoder 모델은 입력되는 문장 또는 문서의 길이와 상관없이 항상 고정된 차원의 실수 vector의 형태로 context vector를 만드는데, 이로 인해 입력이 길어질 경우 입력 문서의 정보를 누락하는 문제가 있다. RNN search 모델은 다음 그림과 같이 디코더의 각 단어 생성 시에 attention mechanism을 이용하여 필요한 입력 열의 hidden state를 추가로 참조하여 context vector를 만들어 RNN encoder-decoder의 문제를 해결하였다.

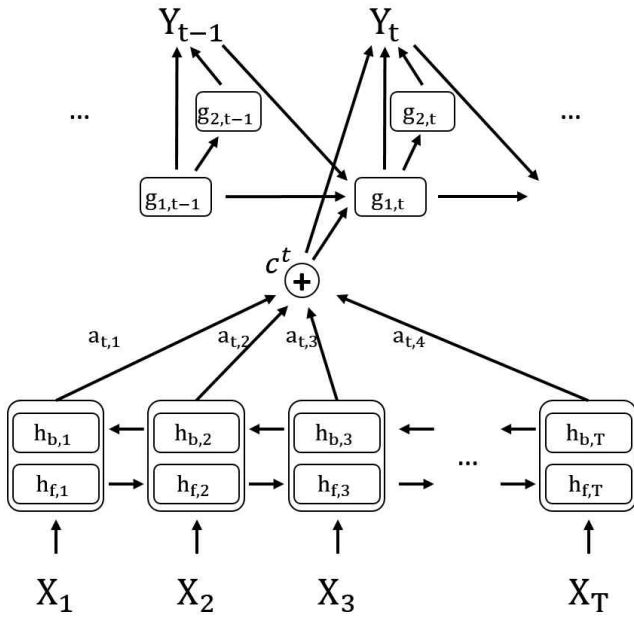


그림2. RNN search 모델

논문에서는 RNN search 모델의 인코더로 bidirectional GRU(Gated Recurrent Unit)를 사용하였고 [10]. 모델 전체를 수식으로 표현하면 다음과 같다.

$$\begin{aligned} h_{f,i} &= GRU_{forward}(E_{src}(x_i), h_{f,i-1}) \\ h_{b,i} &= GRU_{backward}(E_{src}(x_i), h_{b,i+1}) \\ h_i &= [h_{f,i}, h_{b,i}] \end{aligned}$$

$$\begin{aligned} e_i^t &= f_{ATT}(g_{1,t-1}, h_i) \\ a_i^t &= \frac{\exp(e_i^t)}{\sum_{j=1}^{T_s} \exp(e_j^t)} \\ c^t &= \sum_{i=1}^{T_s} a_i^t h_i \end{aligned}$$

$$\begin{aligned} z &= \sigma(W_z E_{tgt}(y_{t-1}) + U_{1,z} g_{1,t-1} + W_{zc} c^t + b_z) \\ r &= \sigma(W_r E_{tgt}(y_{t-1}) + U_{1,r} g_{1,t-1} + W_{rc} c^t + b_r) \\ m &= f(W_m E_{tgt}(y_{t-1}) + U_m(r \odot g_{1,t-1}) + b_m) \\ g_{1,t} &= (1 - z) \odot g_{1,t-1} + z \odot m \\ g_{2,t} &= \text{relu}(W_{g2} g_{1,t} + b_{g2}) \end{aligned}$$

$$y_t = \text{argmax}(\text{softmax} \begin{pmatrix} W_{yg2} g_{2,t} + W_{yg} g_{1,t} \\ + W_y E_{tgt}(y_{t-1}) \\ + W_{yc} c^t + b_y \end{pmatrix})$$

위 수식에서  $h_{f,i}$ 는 입력열을 정방향으로 수행한 GRU의  $i$ 번째 hidden state를 의미하고,  $h_{b,i}$ 는 역방향으로 수행한 GRU의  $i$ 번째 hidden state를 의미하며,  $[h_{f,i}, h_{b,i}]$ 는 concatenate를 의미한다.  $E_{tgt}(y_{t-1})$ 는 이전 시간의 디코

딩 결과로 생성된 단어인  $y_{t-1}$ 의 word embedding이다.  $f_{ATT}$ 는 attention weight( $a_i^t$ )를 결정하기 위한 FFNN(Feed-Forward Neural Network)이며, 디코더의 직전 시간의 첫 번째 레이어 hidden state( $g_{1,t-1}$ )와 인코더의 hidden state( $h_i$ )를 입력으로 사용하고, tanh를 활성화 함수로 사용한다.  $c^t$ 는 현재 시간의 context vector이다.  $g_{1,t}$ 는 디코더의 첫 번째 히든 레이어의  $t$ 번째 hidden state를 의미하며,  $g_{2,t}$ 는 ReLU를 이용하는 디코더의 두 번째 히든 레이어를 의미한다. 디코더의 최종 출력( $Y_{t-1}$ )은 두 디코더의 hidden state와 직전 시간의 디코더 출력( $E_{tgt}(y_{t-1})$ ), 그리고 context vector에 각각 가중치를 곱하고, softmax를 사용하여 출력 사전에 있는 각 어휘의 생성 확률을 구한다.

### 3.3 Input feeding을 추가한 RNN search 모델

Input feeding은 디코더의 이전 시간에 결정된 두 번째 레이어의 hidden state 정보를 현재 시간 attention 계산과정의 입력과, 디코더의 첫 번째 레이어에서 입력으로 사용함으로써, 신경망을 수직, 수평으로 복잡하게 만들어 준다.

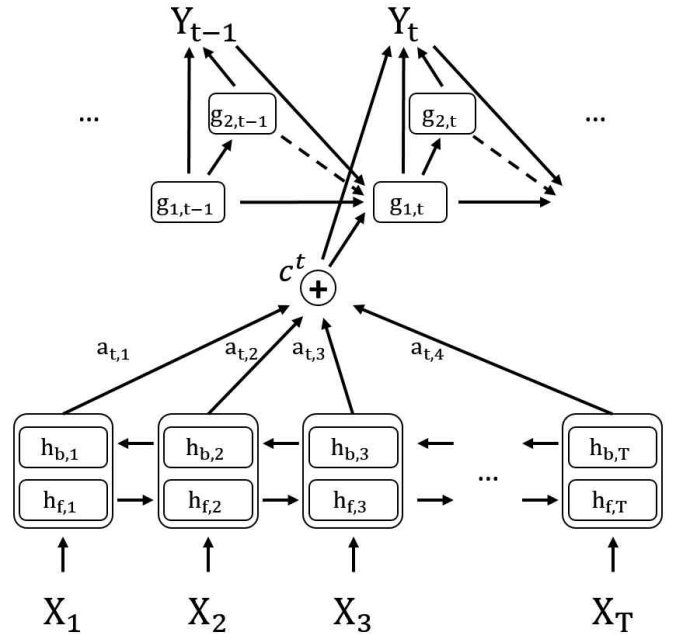


그림3. Input feeding을 적용한 RNN search

Input feeding을 추가한 RNN search 모델은 다음 그림 3과 같이 나타낼 수 있고, 앞서 설명한 RNN search 수식에서 attention을 위한  $e_i^t$ 와, 디코더 부분인  $z$ ,  $r$ ,  $m$ ,  $y_t$ 를 다음 수식으로 변형한 형태로 정의 된다.

$$\begin{aligned}
e_i^t &= f_{ATT}(E_{tgt}(y_{t-1}), g_{1,t-1}, g_{2,t-1}, h_i) \\
a_i^t &= \frac{\exp(e_i^t)}{\sum_{j=1}^T \exp(e_j^t)} \\
c^t &= \sum_{i=1}^{T_s} a_i^t h_i \\
z &= \sigma \left( W_z E_{tgt}(y_{t-1}) + U_{1,z} g_{1,t-1} \right. \\
&\quad \left. + U_{2,z} g_{2,t-1} + W_{z,c} c^t + b_z \right) \\
r &= \sigma \left( W_r E_{tgt}(y_{t-1}) + U_{1,r} g_{1,t-1} \right. \\
&\quad \left. + U_{2,r} g_{2,t-1} + W_{r,c} c^t + b_r \right) \\
m &= f \left( W_m E_{tgt}(y_{t-1}) + U_{2,m} g_{2,t-1} \right. \\
&\quad \left. + U_m (r \odot g_{1,t-1}) + W_{m,c} c^t + b_m \right) \\
g_{1,t} &= (1 - z) \odot g_{1,t-1} + z \odot m \\
g_{2,t} &= \text{relu}(W_{g2} g_{1,t} + b_{g2}) \\
y_t &= \text{argmax}(\text{softmax} \left( \begin{array}{c} W_{yg2} g_{2,t} + W_{yg} g_{1,t} \\ + W_y E_{tgt}(y_{t-1}) \\ + W_{y,c} c^t + b_y \end{array} \right))
\end{aligned}$$

위 수식과 같이 input feeding 모델은  $f_{ATT}$ 와 디코더 첫 번째 히든 레이어( $g_{1,t}$ )의 입력으로 직전 시간 두 번째 히든레이어의 hidden state( $g_{2,t}$ )를 사용하는 특징이 있다.

### 3.4 Copy net 모델

Copy mechanism은 디코딩 과정에서 문장을 생성할 때 필요한 어휘가 출력 사전에 없어 발생하는 문제 (Out-of-Vocabulary)와 고유명사의 출력 확률이 낮아지는 문제를 해결하기 위해 고안된 방법으로, 출력에 필요한 어휘를 입력 시퀀스에서 찾아 출력에 복사하는 방법이다[6]. 본 논문에서는 input feeding이 추가된 RNN search 모델에 다음과 같이 copy mechanism을 추가하였다.

$$\begin{aligned}
e_{copy_j}^t &= f_{ATT_{COPY}}(E_{tgt}(y_{t-1}), g_{1,t-1}, g_{2,t-1}, h_i) \\
s_t &= W_{sh2} g_{2,t} + W_{sh} g_{1,t} \\
&\quad + W_{sy} E_{tgt}(y_{t-1}) \\
&\quad + W_{sc} c^t + b_s \\
P(y_t|X) &= \begin{cases} \frac{1}{Z} \left( \exp(s_t) + \sum_{j: x_j=y_t} \exp(e_{copy_j}^t) \right), & y_t \in X \\ \frac{\exp(s_t)}{Z}, & otherwise \end{cases}
\end{aligned}$$

위 수식에서  $h_i$ 는 인코더에서  $h_{f,t}$ 와  $h_{b,t}$ 를 concatenate

한 값이고,  $E_{tgt}(y_{t-1})$ 는 이전 시간의 디코딩 결과로 생성된 단어인  $y_{t-1}$ 의 word embedding이다.  $g_{1,t-1}$ 는 디코더의 첫 번째 히든레이어의 t-1 번째 hidden state이고,  $g_{2,t-1}$ 는 디코더의 두 번째 히든레이어의 t-1 번째 hidden state이다.  $s^t$ 는 앞서 설명한 input feeding 모델의  $y_t$ 를 구하기 위해 사용하는 수식과 같으며 각 어휘가 출력될 점수 값을 의미한다.  $e_{copy_j}^t$ 는 copy를 위한 attention 점수이며, 입력 시퀀스에 있는 각 어휘가 현재 디코더 time step의 결과로 출력될 스코어 값을 의미한다. Z는 점수 값을 확률 값으로 바꾸기 위한 상수이다. 따라서 디코더의 최종 출력인  $y_t$ 는 각 어휘의 출력 점수에 입력열에 등장한 어휘의 copy attention 점수를 더한 값에 비례한다. 그림4는 copy net 모델을 나타낸다.

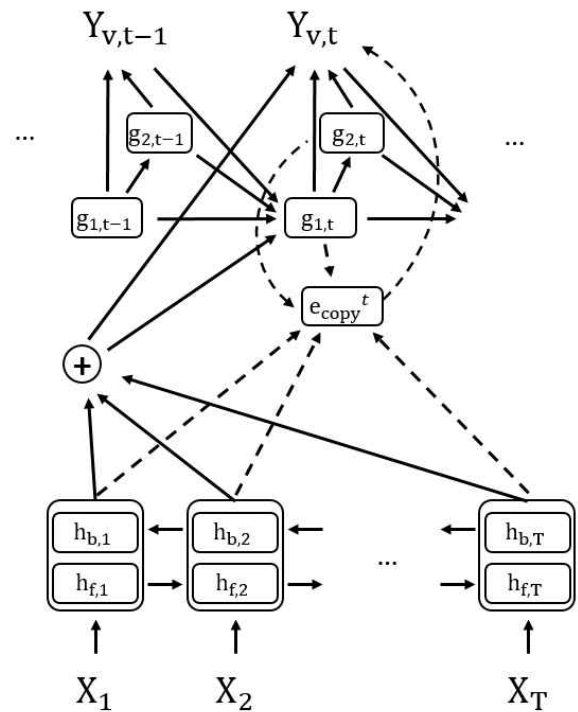


그림4. Copy mechanism과 input feeding을 적용한 RNN search 모델

## 4. 실험

문서요약 데이터 셋 구축을 위해 비교적 기사 형식이 규칙적인 인사이트[9] 인터넷 신문 기사를 수집하여, 기사가 작성한 요약이 있는 경우 해당 요약을 정답요약문으로 사용하고, 없을 경우 기사의 첫줄을 요약문으로 사용하였으며, 한 문장으로만 작성되어 있는 기사는 제외하였다. 실험에 사용한 데이터 셋은 30,291 문서의 학습 셋과 3,786 문서의 개발 셋, 3,705 문서의 평가 셋으로 구성되었다.

정답 요약문과 문서를 형태소 단위로 나누어 각 모델 별로 실험하였고, ROUGE를 사용하였다. 형태소 단위 토큰은 총 19,792개의 어휘를 사용하였고, 200차원으로 만든 워드임베딩을 사용하여 실험하였다.

다음 표에서 h는 각 모델의 인코더와, 디코더의 히든 레이어에서 사용한 유닛의 개수이고, ROUGE-1은 형태소

단위 uni-gram으로 평가한 ROUGE-N 성능이고, ROUGE-2는 bi-gram으로 평가한 ROUGE-N 성능이다. GRU search는 3.2에서 소개한 모델이고, GRU search + Input feeding은 3.3에서 소개한 모델의 성능을 나타내며, GRU search + Input feeding + Copy는 3.4에 기술한 모델의 성능을 나타낸다.

표1. 형태소기반 문서요약 성능

	h	ROUGE-1	ROUGE-2	ROUGE-L
GRU search	300	29.9	10.67	24.71
GRU search +Input feeding	300	29.55	10.52	24.25
GRU search +Copy	300	35.41	15.26	29.24
GRU search +input feeding +Copy	300	<b>35.92</b>	<b>15.37</b>	<b>29.45</b>

실험결과 input feeding + copy 모델이 ROUGE-L 29.45로 가장 우수한 성능을 보였다. GRU search + Input feeding 모델은 GRU search 모델에 비해 약간의 성능 하락이 있었으나, Copy net과 결합한 경우 Copy net만 쓴 경우보다 성능이 약간 상승하였다.

그림5.와 그림6.은 copy 모델을 사용한 형태소기반 실험의 출력 예제를 나타낸다. 그림5.와 그림6.에서 상단의 장문은 요약의 대상이 되는 신문 기사이고, Reporter는 기자가 작성한 정답 요약문이고, System은 copy 모델이 해당 기사를 입력으로 출력한 요약문이다.

지난 13일 두 사람 측근에 따르면 주상욱과 차예련은 지난달 함께 뉴욕 여행을 다녀온 것으로 알려졌다. 주상욱과 차예련은 각각 인스타그램을 통해 뉴욕 여행 인증샷을 남기는 등 여전히 달달한 연애를 이어가고 있음을 공개하기도 했다. 두 사람이 함께 찍힌 모습은 없지만 2인분의 음식 사진과 서로를 찍어준 듯한 모습에서 여느 연인과 다른 행복함이 느껴지기도 했다. 한편 두 사람은 MBC '화려한 유혹'에서 만나 연인 사이로 발전했으며, 지난 3월 열애 사실을 공식 인정했다.

Reporter	GRU search	GRU search +Input feeding +Copy mechanism
배우 주상욱과 차예련이 여전히 달달한 사랑을 키워 가고 있다.	<b>배우</b> 구혜선과 안재현이 열애 중인 것으로 알려졌다.	<b>배우</b> 주상욱과 차예련이 뉴욕 여행을 다녀온 것으로 알려졌다.

그림5. Copy mechanism을 이용한 시스템의 출력결과1  
(기사 출처 : [www.insight.co.kr/newsRead.php?ArtNo=65188](http://www.insight.co.kr/newsRead.php?ArtNo=65188))

지난 16일 트위터에는 로드킬로 숨진 친구를 안전한 곳에 옮긴 뒤 묵묵히 자리를 지킨 길고양이의 사진이 게재되며 많은 이들의 가슴을 먹먹하게 했다. 공개된 사진 속 고양이는 도로에서 차에 치여 처참히 쓰러진 친구를 발견하고 그 자리에 멈춰다. 앙상한 팔다리로 길고양이는 사력을 다해 친구를 안전한 곳으로 옮겼다. 고양이의 놀라운 행동은 여기서 끝이 아니었다. 고양이는 마치 떠난 친구를 애도하듯 하염없이 사체를 바라봤다. 이내 바로 옆에 자리를 잡고 묵묵히 친구의 곁을 지켰다. 그 모습이 혼자 떠난 친구의 마지막 가는 길을 지켜주는 것 같아 보는 이들의 눈시울을 붉게 만들었다. 사진을 본 누리꾼들은 "비록 말 못하는 동물이지만 친구를 잃은 슬픔과 고통은 사람과 다르지 않은 것 같다"는 반응을 보였다.

Reporter	GRU search	GRU search +Input feeding +Copy mechanism
길고양이는 안타깝게 숨진 친구의 곁을 차마 떠나지 못했다.	길을 잃고 쓰러진 친구를 구하기 위해 친구를 잃은 고양이의 안타까운 사연이 알려졌다. 보는 이들의 마음을 훈훈하게 했다.	로드킬로 숨진 친구의 곁을 지키는 길고양이의 모습이 포착되었다.

그림6. Copy mechanism을 이용한 시스템의 출력결과2  
(기사 출처 : [www.insight.co.kr/newsRead.php?ArtNo=19545](http://www.insight.co.kr/newsRead.php?ArtNo=19545))

그림5.와 그림6.을 통해, copy net 모델이 입력문서에서 단어들을 복사해 오는 것을 볼 수 있으며(밑줄 친 부분), 입력문서에 없는 단어들도 생성해 내는 것을 볼 수 있다(□부분). 그림5.에서 GRU search 모델은 “주상욱과 차예련”을 “구혜선과 안재현”으로 잘못 생성하였으나 input feeding과 copy mechanism을 사용한 모델은 “주상욱과 차예련”으로 잘 생성한 것을 확인할 수 있다.

## 5. 결론

본 논문에서는 copy mechanism과 input feeding을 결합한 Sequence-to-Sequence 모델을 이용하여 생성요약 방법으로 한국어 문서요약에 적용하였다. 실험결과 copy mechanism과 input feeding을 결합한 모델이 가장 좋은 성능을 보였다.

본 논문에서 구축한 데이터 셋에 있는 기사에서 종종 기자가 작성한 요약문이 기사의 내용을 대표하지 못하는 경우가 있었다. 이러한 문제들을 노이즈로 가정하고 실험하였지만 이는 시스템의 성능 저하 원인이 될 수 있고, 정확한 시스템 성능을 평가하기 어렵다는 문제가 있다.

향후연구로는 데이터 셋을 정제하여 무결한 데이터 셋을 구축하여 시스템의 성능을 향상시키고, 정확한 성능지표를 구할 것이다. 그리고 구축한 데이터 셋을 이용하여 한국어 문서요약에 적합한 모델을 연구할 것이다.

## 감사의 글

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임. (No.R0101-16-0062, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발)

## 참고문헌

[1] 윤재민, 정유진, 이종혁, “육하원칙 활성화도를 이

- 용한 신문기사 자동추출요약” , 정보과학회논문지: 소프트웨어 및 응용, 제31권, 제4호, pp. 505-515, 2004.
- [2] 김철원, 박선, “의사연관피드백과 용어 가중치에 의한 문서요약” , 한국정보통신학회논문지, 제16권, 제3호, pp 533-540.
- [3] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
- [4] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078, 2014.
- [5] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025, 2015.
- [6] Gu, Jiatao, et al. "Incorporating copying mechanism in sequence-to-sequence learning." arXiv preprint arXiv:1603.06393, 2016.
- [7] Lin, Chin-Yew. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004.
- [8] D. Wang, et al, “Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization,” ACM SIGIR conference on Research and development in information retrieval, pp. 307-314, 2008.
- [9] <http://www.insight.co.kr/>