

시퀀스-투-시퀀스를 활용한 데이터 기반 문장 생성 모델 비교

김영석 · 이병훈 · 강민지 · 한성원[†]

고려대학교 산업경영공학부

Comparison of Data 2Text Generation models using Sequence-to-Sequence

Young-Seok Kim · Byung-Hun Lee · Kang Minji · Sung Won Han

School of Industrial and Management Engineering, Korea University

In this study, we compare the modified encoding method and attention method using a sequence-to-sequence model in the Data2Text task. In addition, we propose a model that complements the shortcomings. Data2Text refers to the task of generating description from tabular data. The data consists of a set of values and fields, and there are significant performance differences depending on how the model learns these structures. Therefore, we compare two representative models utilizing sequence-to-sequence under the same conditions to find a more effective methodology for learning the structure of the data. In addition, it adds a copy mechanism to improve performance by allowing the output of words that are not in the word vocabulary, such as proper nouns. As a result of the experiment, encoding methods using values, fields, position information and dual attention, which combines values and field attention, showed the best performance. In addition, performance improvement in the sequence-to-sequence model was confirmed when the copy mechanism was utilized.

Keywords: Sequence-to-Sequence, Data 2 Text, Natural Language Generation, Attention, Copy Mechanism

1. 서론

데이터를 표 형식으로 저장하는 것은 정보를 일목요연하게 표현할 수 있는 하나의 방법이 되어 널리 사용되고 있다. 그러나 표 형식의 데이터는 문장으로 표현된 것이 아니므로 의도 혹은 의미가 왜곡될 수 있고 정보의 양이 많아질수록 불필요한 정보까지 추가되어 정확한 의미 전달에 어려움을 줄 수 있다. 예를 들어, 정형화된 기상 측정값을 그대로 전달하는 것보다는 측정값 중 유의미한 값들로 생성된 일기예보를 전달하는 것이 효과적일 것이며, 스포츠 경기의 모든 기록을 표로 전달하는 것보다는 중요한 기록으로 생성된 요약문을 전달하는 것이 이해하기 쉬운 것이다(Lebret *et al.*, 2016).

다른 대표적인 예시로 군의 보고서가 있다. 우리 군에서는 전장 시뮬레이션을 통해 여러 전장 상황을 모의할 수 있으며

이 과정에서 다양한 로그 데이터가 생성된다. 생성된 로그 데이터는 표 형식으로 된 데이터로 정리할 수 있으며 일기예보 그리고 경기기록 요약문처럼 전장 상황 보고서(요약문)로 표현할 수 있다. 이러한 보고서는 지휘관의 의사결정 능력향상을 위한 훈련에 사용할 뿐만 아니라 보고서로부터 의사결정을 하는 지휘관을 모방할 수 있는 AI 학습에도 사용될 수 있다(Lee *et al.*, 2021).

이렇게 표 혹은 데이터를 자연어로 표현하는 과업을 데이터-투-텍스트(Data-to-Text : Data2Text)라고 한다. Data2Text 생성 연구에서는 표의 정보를 값과 필드로 구분하여 명명하는데, 필드는 변수명에 해당하며 값은 해당 변수의 값을 의미한다. Data2Text는 쉽고 단순하지만 좋은 성능을 보이는 규칙에 기반한 방법과 규칙을 학습하여 텍스트를 생성하는 딥러닝에 기반한 방법이 있다. 규칙에 기반한 방법은 일반적인 요약문 생

본 연구는 방위사업청과 방위산업기술지원센터의 지원으로 수행되었습니다(UC200020D).

[†] 연락처 : 한성원 교수, 02841, 서울특별시 성북구 안암로 145 고려대학교 산업경영공학부, Tel : 02-3290-3384, Fax : 02-3290-4550,

E-mail : swhan@korea.ac.kr

2022년 4월 29일 접수; 2022년 7월 25일 수정본 접수; 2022년 7월 28일 게재 확정.

성의 평가지표에서는 좋은 성능을 보이지만 사람이 평가한 유창성, 명백성 그리고 정확성에서는 딥러닝에 기반한 방법이 더 좋은 성능을 보인다(Chris *et al.*, 2018). 또한 규칙에 기반한 방법은 텍스트 생성을 위해 수동으로 작성된 템플릿과 규칙이 필요하여 상황이 복잡해지고 분야가 확장되면 어려움을 겪는다. 반면에, 딥러닝에 기반한 방법은 데이터에 기반하여 학습하므로 더 자연스러운 문장을 생성할 수 있고 유지보수에도 더욱 장점이 있다.

딥러닝에 기반한 Data2Text는 시퀀스-투-시퀀스(Sequence-to-Sequence : Seq2Seq) 모델을 사용한 다양한 방법론이 존재한다(Sutskever *et al.*, 2014). 표의 정보를 인코딩하는 방법과 특히, 어텐션(Attention)을 어떻게 적용하는지에 따라 큰 성능 차이를 보인다. Seq2Seq을 사용한 대표적인 선행연구로 Liu *et al.*(2018)의 듀얼 어텐션(Dual attention)을 적용한 연구와 Sha *et al.*(2018)의 하이브리드 어텐션(Hybrid attention)을 적용한 연구가 있다. 그러나 입력과 인코딩 방법이 달라 어텐션의 성능을 직접적으로 비교하기에는 어려움이 있다.

본 연구에서는 Seq2Seq을 사용한 대표적인 두 선행연구의 모델에 동일한 입력과 인코딩 방법을 적용하여 기존의 바닐라 어텐션(Vanilla Attention)에서 변형된 어텐션의 성능을 비교한다. 그리고 이를 통해 Data2Text에서 효과적인 어텐션 방법론을 찾고자 한다. 또한, 선행연구의 모델에서의 단점을 보완할 수 있는 방법을 제안하고자 한다. 두 선행연구에서 활용한 데이터인 Wikibio 데이터(Lebret *et al.*, 2016)를 활용하여 연구를 진행했으며, 요약문 생성 분야에서 주로 사용되는 평가지표인 ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin *et al.*, 2004) 그리고 BLEU (Bilingual Evaluation Understudy Score) (Papineni *et al.*, 2002)로 평가하였다.

본 연구는 총 5장으로 구성된다. 2장에서는 Data2Text 모델에 대한 선행연구와 다양하게 변형된 방법론을 소개한다. 3장에서는 본 연구에서 사용된 데이터에 관해 설명하며 4장에서는 사용한 방법론을 설명한다. 5장에서는 사용한 모델의 성능을 비교하고 결과를 분석하며 6장에서는 결과에 대한 요약과 본 연구의 한계점 그리고 향후 연구 방향을 제안한다.

2. 선행 연구

2.1 규칙 기반 모델(Rule based Model)

과거의 Data2Text 생성 연구는 주어진 입력(표)에서 필요한 값을 추출하여 정해진 형식대로 글을 생성하는 규칙 기반(rule-based)의 모델들을 사용하였다(Dale *et al.*, 2003; Reiter *et al.*, 2005; Green, 2006; Galanis and Androutsopoulos, 2007; Turner *et al.*, 2010). 즉, 표로부터 특정 속성을 선택하고 이미 고정된 문장 구조에 선택된 값들을 채워 텍스트를 생성하는 것이다. 그러나 최근에는 딥러닝 모델이 발전하면서 신경망으

로 규칙 자체를 학습하는 연구들이 진행되고 있다.

2.2 시퀀스-투-시퀀스 기반 모델(Sequence-to-Sequence based Model)

신경망을 사용한 모델은 대표적으로 인코더(encoder)와 디코더(decoder)를 연결한 end-to-end 방식의 Seq2Seq 모델이 있다. Seq2Seq은 대표적으로 문서 요약, 언어 번역 그리고 대화 생성 등의 목적으로 활용되는 모델이다. 문장을 입력으로 받은 인코더는 정보를 축약하고 디코더에 보내며, 축약된 정보를 받은 디코더는 순차적으로 단어를 출력한다. Data2Text 생성 연구에서는 필드 값을 문장과 같이 나열하여 인코더의 입력으로 활용하고, 디코더에서 표의 정보를 바탕으로 생성된 단어들이 문장을 이룬다.

Lebret *et al.*(2016)은 Seq2Seq 모델을 사용하여 지역적 조건(Local Conditioning)과 전역적 조건(Global Conditioning)을 결합한 조건부 신경망 언어 모델을 제안하였다. 지역적 조건은 이미 생성된 단어의 필드 정보를 나타내고, 전역적 조건은 모든 입력 데이터 표의 필드 정보를 나타낸다. 해당 연구는 위키 피디아로부터 얻은 전기에 관련된 표와 설명문의 첫 번째 문장으로 구성된 WIKIBIO 데이터를 사용하였다. 그러나 Lebret *et al.*(2016)과 같은 신경망 구조는 문장이 길어지면 그 구조를 이해하거나 생성하는 데 한계가 있다(Puduppully *et al.* 2019). 따라서 Puduppully *et al.*(2019)은 문장을 생성하기 위해 입력 데이터의 콘텐츠 선택(content selection)과 콘텐츠 계획(content planning)을 통합한 신경망 구조를 제안하였다. 그리고 장단기 메모리(Long Short-Term Memory: LSTM) 구조를 기반한 디코더를 사용하여 문장을 생성하였다.

2.3 어텐션 메커니즘(Attention Mechanism)

Seq2Seq 모델은 인코더에서 정보를 압축하는 과정에서 정보 손실이 생기고 입력의 길이가 길어질수록 입력의 앞쪽의 정보가 소실되는 문제가 있다. 어텐션 메커니즘(Attention Mechanism)은 이러한 문제점을 출력 단어를 예측할 때 모든 입력 단어들의 정보를 다시 한 번 참고하여 보완한다(Luong *et al.*, 2015). 예를 들면 다음으로 생성할 단어가 '나이'와 관련된 단어라면 입력된 표에서의 나이에 대한 정보에 집중하게 되는 것을 말한다.

선행 연구에서는 기존의 어텐션 메커니즘을 적용한 연구와 바닐라 어텐션에서 변형된 어텐션 메커니즘을 적용한 연구들이 있다. Mei *et al.*(2016)은 일기예보 데이터인 WEATHERGov에 LSTM 구조를 기반으로 한 Seq2Seq 모델에 어텐션 메커니즘을 적용하였다. 일기예보문을 생성할 때 입력된 날씨 데이터와 관련된 토큰을 출력할 때는 어텐션 메커니즘을 통해 해당 토큰과 관련된 입력 데이터에 더 집중함을 확인하였다. 그 뿐만 아니라 Sha *et al.*(2018)은 사람이 표의 정보를 참고하여

글을 작성할 때는 순서를 고려하여 작성하게 된다는 특징을 반영하기 위한 하이브리드 어텐션 메커니즘을 제안하였다. 하이브리드 어텐션 메커니즘은 입력에 대한 기존의 어텐션 메커니즘에 링크 기반 어텐션 메커니즘(Link-based Attention Mechanism)을 더한 방법이다. 링크 기반 어텐션 메커니즘은 사람이 글을 쓸 때, 순서를 반영하여 작성한다는 특징을 반영할 수 있도록 하는 역할을 한다. 예를 들어 ‘거주지 : A, 이름 : B’와 같은 정보를 토대로 글을 작성하면 ‘저는 A에 사는 B입니다.’라고 쓸 것이다. 이처럼 일반적으로 사는 위치를 말한 후에 본인의 이름을 설명할 것이며, 링크 기반 어텐션 메커니즘은 글을 작성하는 순서를 반영하는 변형된 어텐션 메커니즘의 역할을 한다.

2.4 복사 메커니즘(Copy Mechanism)

복사 메커니즘은 Gu *et al.*(2016)이 제안한 방법론으로 사람이 대화할 때 사람 이름과 같은 고유명사를 반복하는 경향을 반영하고자 고안되었다. 이러한 고유명사는 자주 등장하지 않거나 단어 사전에 존재하지 않는 경우가 많아 출력되지 못하는 문제가 발생한다. 이를 해결하기 위해 제안된 복사 메커니즘은 단어를 출력할 때 단어 사전으로부터 단어를 생성하기도 하지만, 입력 데이터에서 단어를 복사하여 출력할 수 있게 하는 방법론이다. Gu *et al.*(2016)은 단어 사전에서의 생성확률과 입력으로부터의 복사확률을 구해 단어를 생성할 것인지 복사할 것인지를 학습하게 하였다. See *et al.*(2017)의 복사 메커니즘은 입력에서 등장한 단어들이 추가된 확장된 단어 사전의 분포를 통해 단어를 생성한다. 디코더가 단어를 생성할 때 단어 사전의 분포와 생성확률을 곱하고, 입력 단어들의 어텐션 값과 복사확률을 곱해 이들을 더하는 방법으로 복사 메커니즘

을 적용한다. 이러한 방법은 자주 등장하지 않는 단어의 문제의 해결책으로 주로 사용되고 있다.

Chen *et al.*(2019)은 사전 학습 모델인 Radford *et al.*(2019)의 GPT를 디코더로 활용하여 Data2Text에서 복사 메커니즘을 적용한 모델을 제안하였다. GPT는 매우 큰 다양한 도메인의 학습 데이터로 학습한 모델로 최근 다양한 자연어 과업에서 활용되고 있는 모델이다. 또한, 적은 수의 데이터만을 학습에 사용하고 성능을 평가하는 few-shot 환경에서 복사 메커니즘을 적용하여 Data2Text에서 입력으로 주어지는 표의 정보를 잘 반영하도록 하였다.

3. 데이터셋

데이터는 Lebret *et al.*(2016)이 공개한 벤치마크 데이터인 WIKIBIO 데이터를 활용하였다. WIKIBIO는 위키피디아(Wikipedia)에 기록되어있는 인물들에 대한 전기로 구성되어 있다. <Figure 1>은 WIKIBIO 데이터를 구성하는 위키피디아의 기사에 대한 예시이다. 오른쪽 표는 “Walter Extra”라는 인물의 정보를 요약한 표이며, 왼쪽의 본문은 해당하는 인물에 대한 설명이다. 데이터는 생년월, 국적 그리고 직업의 정보를 포함하고 있는 오른쪽 표를 입력으로 하고, 해당 인물에 대한 설명인 왼쪽 본문의 첫 문장이 정답 요약문으로 구성된다.

WIKIBIO 데이터의 입력 표와 정답 요약문의 통계량은 <Table 1>과 같다. 먼저 입력 표의 평균 단어 수는 53.1이며 정답 요약문의 평균 단어 수는 26.1이다. 또한 정답 요약문에 포함된 입력 표의 평균 단어 수는 9.5이며 입력 표의 평균 필드 수는 19.7이다. 해당 데이터의 80%는 Train Set, 10%는 Validation Set 그리고 나머지 10%는 Test Set으로 사용했다.

The image shows a Wikipedia article for Walter Extra. At the top, it says 'Walter Extra' and 'From Wikipedia, the free encyclopedia'. Below that, there is a warning box: 'This biography of a living person needs additional citations for verification. Please help by adding reliable sources. Contentious material about living persons that is unsourced or poorly sourced must be removed immediately, especially if potentially libelous or harmful. Find sources: "Walter Extra" - news - newspapers - books - scholar - JSTOR (June 2010) (Learn how and when to remove this template message)'. The main text starts with: 'Walter Extra is a German award-winning aerobatic pilot and chief aircraft designer who founded the aerobatic aircraft manufacturer Extra Flugzeugbau (Extra Aircraft Construction).^{[1][2]} Extra was trained as a mechanical engineer. He began his flight training in gliders, transitioning to powered aircraft to perform aerobatics. He built and flew a Pitts Special aircraft and later built his own Extra EA-230.^[3] Extra began designing aircraft after competing in the 1982 World Aerobatic Championships. His aircraft constructions revolutionized the aerobatics flying scene and still dominate world competitions. The German pilot Klaus Schrodt won his world championship title flying an aircraft made by the Extra firm. Walter Extra has designed a series of performance aircraft which include unlimited aerobatic aircraft and turboprop transports.^[4]

On the right side, there is a table with the following information:

Walter Extra	
Born	April 1954
Nationality	German
Occupation	Aircraft designer and manufacturer

Below the table is a photo of an aircraft with the caption: 'An Extra 300 during landing.'

At the bottom, there is a 'References' section with four entries:

- ^ "Home Page". Extra Aircraft. Retrieved 7 June 2010.
- ^ McClellan, J. Mac. "The Extra 500 Now Has a Chance". *Flying Magazine*. Retrieved 7 June 2010.
- ^ Peter Hirschman (December 2013). "Aviation Legends Walter Extra". *AOPA Pilot*. 62.
- ^ Peter Hirschman (December 2013). "Aviation Legends Walter Extra". *AOPA Pilot*. 61.

Figure 1. Example of Wikipedia article for WIKIBIO Dataset

Table 1. Dataset Statistics

		Value
Mean	Tokens per input table	53.1
	Tokens per sentence	26.1
	Input table token per sentence	9.5
	Fields per input table	19.7
Number of Data	Train Set	582,659
	Validation Set	72,831
	Test Set	72,831
	Total	728,321

3.1 평가지표

모델의 성능 평가를 위한 평가지표는 루지(ROUGE : Recall-Oriented Understudy for Gisting Evaluation)와 블루(BLEU: Bilingual Evaluation Understudy) 점수를 사용한다. 루지 점수는 정답 요약문과 모델 요약문의 N그램 recall에 기반하여 계산하며 precision 혹은 recall과 precision의 조화평균인 f1 score에 기반한 계산도 가능하다. 그 과정은 Eq (1)과 같다.

$$\begin{aligned}
 ROUGE_N - recall &= \frac{\sum_{S \in Reference} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in Reference} \sum_{gram_n \in S} Count(gram_n)} \\
 ROUGE_N - precision &= \frac{\sum_{S \in Reference} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in Candidate} \sum_{gram_n \in S} Count(gram_n)} \quad (1) \\
 ROUGE_N - f1score &= 2 \times \frac{ROUGE_N precision \times ROUGE_N recall}{ROUGE_N precision + ROUGE_N recall}
 \end{aligned}$$

첫 번째 recall의 경우, 분모의 $\sum_{S \in Reference} \sum_{gram_n \in S} Count(gram_n)$ 는 모델 요약문의 N그램의 수를 의미하며 분자의 $\sum_{S \in Reference} \sum_{gram_n \in S} Count_{match}(gram_n)$ 는 모델 요약문과 정답 요약문의 N그램이 겹치는 수를 의미한다. 두 번째 precision의 경우, 분모는 정답 요약문의 N그램의 수를 의미하고 분자는 첫 번째와 동일하게 모델 요약문과 정답 요약문의 N그램이 겹치는 수를 의미한다.

블루 점수는 루지 점수와 비슷하게 정답 요약문과 모델 요약문의 N그램을 통한 순서쌍이 겹치는 수를 측정한다. 그러나 블루 점수는 N그램 precision에 기반하여 계산하고 예측문장 길이에 대한 페널티가 존재한다는 차이점이 있으며 Eq (2)와 같은 방법으로 계산한다.

$$\begin{aligned}
 BLEU_N &= \frac{\sum_{S \in Reference} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in Candidate} \sum_{gram_n \in S} Count(gram_n)} \quad (2) \\
 BLEU &= \min(1, \frac{Candidate length}{Reference length}) (\prod_{i=1}^4 BLEU_i)^{\frac{1}{4}}
 \end{aligned}$$

$BLEU_N$ 의 경우 Eq(1)의 $ROUGE_N - precision$ 과 동일하며, $BLEU_N$ 을 4그램까지 계산하여 곱하고 예측문장 길이가 짧을수록 값이 커지는 것을 방지하기 위해 예측문장 길이를 정답 요약문의 길이로 나누고 1과 비교하여 더 작은 값을 곱함으로써 페널티를 부여한다.

4. 방법론

Data2Text에서 Seq2Seq과 바닐라 어텐션을 활용한 여러 선행 연구가 존재하나, 기존의 바닐라 어텐션을 Data2Text에 맞게 변형한 어텐션을 적용한 선행연구는 두 선행연구가 대표적이다. 따라서, 본 연구에서는 Seq2Seq을 활용한 두 선행연구의 방법론을 동일한 조건으로 비교하여 Data2Text에서 효과적인 어텐션 방법론을 찾고자 하며, 나아가 선행연구의 단점을 보완할 수 있는 방법론을 제안하고자 한다. Data2Text는 디코더의 역할 뿐만 아니라 인코더가 표의 정보를 추약하는 방법 또한 매우 중요하다. Liu *et al.*(2018)은 값과 필드를 추약하는 필드 게이트 테이블 인코더(Field-Gating Table Encoder)를 제안하였다. 또한 값과 필드 정보를 활용한 듀얼 어텐션 메커니즘(Dual-Attention Mechanism)을 통해 표의 구조를 이해하는 데 강점을 갖는 모델을 제안하였다. 반면, 앞선 선행연구에서 Sha *et al.*(2018)의 하이브리드 어텐션 메커니즘 기반 모델은 사람의 글쓰기 특징을 반영했다는 특징이 있다. 따라서 기존의 바닐라 어텐션에서 변형된 어텐션 성능을 비교함으로써 Data2Text에서 효과적인 어텐션 방법론을 찾고, 복사 메커니즘을 적용하여 기존 선행연구의 단점인 사전에 없는 단어를 출력할 수 없는 문제를 해결하고자 한다.

4.1 Structure-aware Seq2Seq (Dual Attention based Model)

모델 (Liu *et al.*, 2018)

Structure-aware Seq2Seq 모델은 입력으로 값(value), 필드(field) 그리고 필드 위치(field position) 및 역위치(field reverse position) 값을 받는다. 서론에서 언급한 바와 같이 값은 표에서

의 값을 의미하고 필드는 값에 대한 변수명을 의미한다. 필드 위치 및 역위치는 각 값의 위치를 의미하는데 가장 앞에서부터 세었을 때의 위치와 가장 뒤에서 세었을 때의 위치값의 쌍으로 구성된다.

예를 들어, <Figure 2>의 왼쪽과 같은 표가 주어졌을 때 첫 행에 “walter extra”는 값의 단어가 2개이므로 “walter”와 “extra” 2개의 단어로 나누어 입력으로 사용된다. 필드는 모두 “name”으로 함께 입력으로 사용되며 필드 위치 및 역위치는 각각의 위치를 계산하여 1과 2 그리고 2와 1이 입력으로 사용된다.

이러한 입력들은 학습 과정에서 같이 학습되는 룩업 테이블 (Lookup table)을 통해서 임베딩 벡터로 변환된다. 이때 값, 필

드 그리고 필드 위치 및 역위치는 각각 구분된 룩업 테이블을 가지므로 값과 필드가 같더라도 다른 임베딩을 갖게 되는 특징이 있다. 마지막으로 필드의 임베딩과 필드 위치 및 역위치의 임베딩은 연결(concat)하여 하나의 임베딩으로 사용한다. 즉, 최종적으로 값의 임베딩 벡터와 연결된 필드의 임베딩 벡터가 모델의 입력 임베딩 벡터로 사용된다.

<Figure 3>은 본 연구에서 사용한 Liu *et al.*(2018)의 모델을 도식화한 그림이다. 이 모델은 LSTM 구조에서 변형된 인코더와 바닐라 어텐션에서 변형된 어텐션을 적용한 디코더로 구성된 Seq2Seq 구조를 갖는다. 인코더가 표의 구조를 인코딩하여 디코더에 넘기고 디코더는 인코더의 은닉 상태(hidden state)를

	Value	Fields	Fields Position	Fields reverse Position
	walter	name	1	2
	extra	name	2	1
	1954	birth_date	1	1
	german	nationality	1	1
	aircraft designer and manufacturer	occupation	1	4
	designer	occupation	2	3
	and	occupation	3	2
	manufacturer	occupation	4	1
	walter	article_title	1	2
	extra	article_title	2	1

name	walter extra
birth_date	1954
nationality	german
occupation	aircraft designer and manufacturer
article_title	walter extra

Label	Walter Extra is a German award-winning aerobatic pilot and chief aircraft designer who founded the aerobatic aircraft manufacturer Extra Flugzeugbau (Extra Aircraft Construction).
-------	---

Figure 2. Example of Values, Fields and Fields Position Used as Input

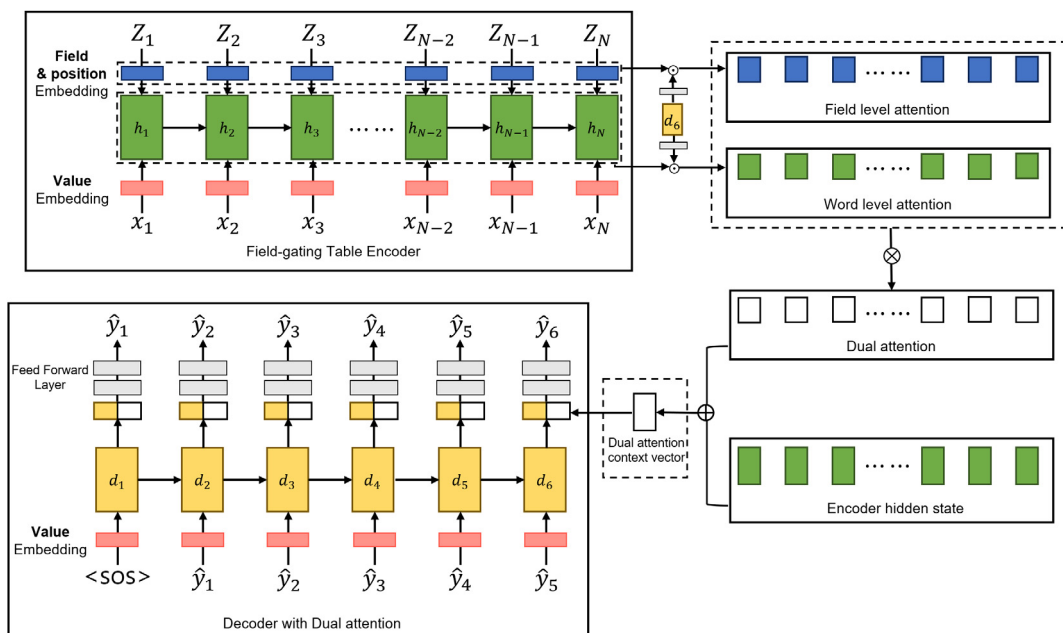


Figure 3. Structure-aware Seq2Seq Model

참조하여 어텐션 점수를 계산한다. 계산된 어텐션 점수와 인코더의 은닉 상태를 가중 합하여 문맥 벡터(context vector)로 압축하고 디코더의 은닉 상태와 연결되어 피드 포워드 층(feed forward layer)을 통과한다. 그 후 한 번 더 피드 포워드 층을 지나고 소프트맥스(softmax) 활성화 함수를 거쳐 단어사전에 대한 확률 분포를 계산하여 가장 높은 확률을 가진 단어를 출력 단어로 예측하게 된다.

(1) 필드 게이트 테이블 인코더(Field-Gating Table Encoder)

필드 게이트 테이블 인코더는 값의 임베딩 벡터와 필드의 임베딩 벡터를 인코딩하는 LSTM 구조 인코더이다. 기존의 LSTM 구조는 값의 임베딩 벡터로 셀 상태(cell state)를 업데이트한다. 그러나 필드 게이트 테이블 인코더는 셀 상태를 업데이트할 때 필드의 임베딩 벡터를 추가로 더해 업데이트하며, 이는 Eq. (3)과 같다.

$$\begin{aligned} f_n &= \sigma(W_{xf}x_n + W_{hf}h_{n-1} + b_f) \\ i_n &= \sigma(W_{xi}x_n + W_{hi}h_{n-1} + b_i) \\ o_n &= \sigma(W_{xo}x_n + W_{ho}h_{n-1} + b_o) \\ g_n &= \tanh(W_{xg}x_n + W_{hg}h_{n-1} + b_g) \\ l_n &= \sigma(W_{zl}z_n + b_l) \\ k_n &= \tanh(W_{zk}z_n + b_k) \\ c_n &= f_n \odot c_{n-1} + i_n \odot g_n + l_n \odot k_n \end{aligned} \quad (3)$$

x_n, h_{n-1} 은 각각 n번째 입력의 임베딩 벡터와 이전 입력값으로부터 출력된 은닉 상태 벡터를 의미하며 z_n 은 필드의 임베딩 벡터를 의미한다. 또한, 각 W 는 임베딩 벡터에 곱해지는 가중치 행렬, 각 b 는 편향 그리고 σ, \tanh 는 각각 시그모이드 함수와 하이퍼볼릭탄젠트 함수를 의미한다. 계산된 f_n, i_n 은 삭제 게이트와 입력 게이트라고 부르며 0과 1 사이의 값을 갖고 각각 이전 입력값의 셀 상태를 삭제 시키는 역할과 현재 입력값의 상태를 기억하는 역할을 한다. g_n 은 -1과 1 사이의 값을 갖고 현재 입력의 셀 값(cell value)을 제안하는 역할을 한다. 기존의 LSTM 구조에 추가된 l_n 은 0과 1 사이의 값으로 셀 상태에 필드의 정보가 저장될 양을 의미하며, k_n 은 필드 게이트(field gate)라고 부르며 셀 상태에 더할 필드의 값을 제안하는 역할을 한다.

앞선 삭제 게이트, 입력 게이트 등을 아다마르 곱(\odot)을 통해 셀 상태를 계산하며 계산된 셀 상태는 하이퍼볼릭탄젠트 함수를 지나 출력 게이트라고 불리는 o_n 과 곱해져 현재 입력값에서의 은닉 상태 벡터인 h_n 을 출력하며 그 과정은 Eq (4)와 같다.

$$h_n = o_n \odot \tanh(c_n) \quad (4)$$

(2) 듀얼 어텐션 디코더(Decoder with Dual Attention)

듀얼 어텐션은 인코더의 은닉 상태와 디코더의 은닉 상태를 통해 매 타임 스텝마다 집중해야 하는 입력 단어를 학습하는

방법인 바닐라 어텐션에서 변형된 방법이다. 바닐라 어텐션은 어텐션 점수를 계산하는 여러 방법이 있지만 본 연구에서는 내적(dot product) 방법을 사용하였다. 계산된 인코더의 은닉 상태와 디코더의 은닉 상태가 각각 완전 연결 층을 통과하고 하이퍼볼릭탄젠트 함수를 지난 후 곱해져 두 은닉 상태의 유사도를 측정하는 점수로 사용된다. 해당 점수는 소프트맥스 함수를 지나고 인코더의 은닉 상태와 가중합 되어 문맥 벡터가 되며 디코더의 은닉 상태와 연결되어 사용된다. 이러한 과정은 Eq (5)와 같다.

$$\begin{aligned} score(d_t, h_n) &= \tanh(W_s h_n) \odot \tanh(W_d d_t) \\ \alpha_{tn} &= \frac{\exp^{score(d_t, h_n)}}{\sum_{k=1}^N \exp^{score(d_t, h_k)}}; \alpha_t = \sum_{n=1}^N \alpha_{tn} h_n \\ v_t &= \tanh(W_z [d_t, \alpha_t]) \end{aligned} \quad (5)$$

여기서 d_t, h_n 은 각각 t번째 타임 스텝의 디코더 은닉 상태와 n번째 입력의 인코더 은닉 상태를 의미한다. α_{tn} 은 t번째 디코더의 은닉 상태와 n번째 인코더 은닉 상태의 유사도를 의미하며 α_t 은 입력으로 주어진 시퀀스의 길이인 N개의 인코더의 은닉 상태 유사도를 가중 합한 문맥 벡터를 의미한다.

이렇게 계산된 v_t 는 완전 연결 층과 소프트맥스 활성화 함수를 거쳐 생성할 단어에 대한 단어 사전의 확률 분포를 출력하게 된다. 그 과정은 Eq (6)과 같으며 H는 인코더의 은닉 상태, Z는 필드의 임베딩을 의미하며 $w_{<t}$ 는 t번째 타임 스텝 이전에 생성된 단어를 의미한다.

$$P(w_t | H, Z, w_{<t}) = \text{softmax}(W_O v_t) \quad (6)$$

듀얼 어텐션은 이러한 바닐라 어텐션에 필드의 임베딩 벡터를 추가로 활용한다. 필드의 임베딩 벡터와 t번째 타임 스텝의 디코더 은닉 상태의 유사도를 측정하고 이를 기존 바닐라 어텐션 점수에 가중합 한다. 이렇게 계산된 어텐션 점수는 표의 모든 정보를 고려하여 문맥적인 정보뿐만 아니라 표의 구조를 이해하는 데 도움을 주며 Eq (7)과 같은 과정으로 표현된다.

$$\begin{aligned} score(d_t, z_n) &= \tanh(W_s z_n) \odot \tanh(W_d d_t) \\ \beta_{tn} &= \frac{\exp^{score(d_t, z_n)}}{\sum_{k=1}^N \exp^{score(d_t, z_k)}} \\ \delta_{tn} &= \frac{\alpha_{tn} \cdot \beta_{tn}}{\sum_{k=1}^N \alpha_{tk} \cdot \beta_{tk}}; \tilde{\alpha}_t = \sum_{i=1}^N \delta_{tn} h_n \end{aligned} \quad (7)$$

z_n 은 인코더의 n번째 입력에서 필드의 임베딩 벡터를 의미하며 β_{tn} 은 필드 단계(field-level)의 어텐션 점수를 의미한다. 앞선 Eq (5)에서의 문맥 벡터인 α_t 는 Eq (7)의 $\tilde{\alpha}_t$ 로 대체되어 최종 단어사전의 확률 분포를 계산하게 된다.

4.2 Order-Planning Neural Text Generation(Hybrid Attention based Model)모델(Sha et al., 2018)

Order-Planning Neural Text Generation 모델(Sha et al., 2018)은 사람의 글쓰기 과정에 집중한 모델이다. 대부분의 글쓰기는 계획을 갖고 정해진 순서에 맞게 작성이 되므로 이를 어텐션 메커니즘을 통해 반영하고자 하였다.

모델은 <Figure 4>와 같은 구조로 구성된다. Structure-aware Seq2Seq 모델과 같은 LSTM 구조에 기반한 Seq2Seq 구조를 갖고 있으며 기존의 바닐라 어텐션에서 변형된 어텐션인 하이브리드 어텐션(Hybrid Attention)과 복사 메커니즘을 적용했다는 특징이 있다.

(1) 인코더(Encoder)

인코더는 Structure-aware Seq2Seq 모델과 동일하게 입력으로 값 그리고 필드를 받는다. 또한 학습 과정에서 학습되는 분리된 룩업 테이블을 통해 같은 단어더라도 다른 임베딩을 갖는 공통점이 있다. 그러나 필드 위치와 역위치는 받지 않는다는 차이점이 있으며 최종적으로 값의 임베딩 벡터와 필드의 임베딩 벡터를 연결하여 모델의 입력 임베딩 벡터로 사용한다.

룩업 테이블을 거쳐 생성된 모델의 입력 임베딩 벡터는 일반적인 LSTM 구조를 갖는 인코더를 거치며 그 과정은 Eq (8)과 같다.

$$\begin{aligned}
 f_n &= \sigma(W_{xf}x_n + W_{hf}h_{n-1} + b_f) \\
 i_n &= \sigma(W_{xi}x_n + W_{hi}h_{n-1} + b_i) \\
 o_n &= \sigma(W_{xo}x_n + W_{ho}h_{n-1} + b_o) \\
 g_n &= \tanh(W_{xg}x_n + W_{hg}h_{n-1} + b_g) \\
 c_n &= f_n \odot c_{n-1} + i_n \odot g_n \\
 h_n &= o_n \odot \tanh(c_n)
 \end{aligned}
 \tag{8}$$

x_n 는 n번째 입력의 값과 필드의 임베딩 벡터가 연결된 모델의 입력 임베딩 벡터를 의미하며 h_{n-1} 은 이전 입력의 인코더 은닉 상태를 의미한다. f_n, i_n, o_n 은 각각 삭제, 입력 그리고 출력 게이트를 의미하고 g_n 은 n번째 입력에서 제안하는 셀 값(cell value)을 의미한다. c_n 은 셀 상태(cell state) 그리고 h_n 은 현재 타임 스텝의 인코더 은닉 상태를 의미하며, 이들의 역할은 Eq (3)에서 설명한 바와 같다.

(2) 하이브리드 어텐션 디코더(Decoder with Hybrid Attention)

하이브리드 어텐션은 표의 전반적인 구조를 이해하는 데 도움을 주는 콘텐츠 기반 어텐션(Content-Based Attention)과 필드의 관계를 이해하는 데 도움을 주는 링크 기반 어텐션(Link-Based Attention)으로 구성된다. 그리고 자가 적응 게이트(self-adaptive gate)를 통한 두 어텐션의 가중합을 통해서 최종 어텐션 점수를 계산한다.

먼저 콘텐츠 기반 어텐션은 4.1.2의 듀얼 어텐션과 유사하게 입력값에 대한 어텐션 점수뿐만 아니라 필드의 임베딩 벡터를 함께 활용한 어텐션이다. 단, 어텐션 점수를 구할 때 t번째 디코더의 은닉 상태가 아닌 t-1 번째 디코더의 은닉 상태를 사용하며 그 과정은 Eq (9)와 같다.

$$\begin{aligned}
 \alpha_{tn}^{value} &= h_n W_v d_{t-1} \\
 \alpha_{tn}^{field} &= z_n W_f d_{t-1} \\
 \alpha_{tn}^{content} &= \frac{\exp(\alpha_{tn}^{value}) \alpha_{tn}^{field}}{\sum_{k=1}^N \exp(\alpha_{tk}^{value}) \alpha_{tk}^{field}}
 \end{aligned}
 \tag{9}$$

h_n, z_n 은 각각 인코더의 은닉 상태와 필드의 임베딩 벡터를 의미하며 d_{t-1} 은 t-1 번째 타임 스텝의 디코더 은닉 상태를 의

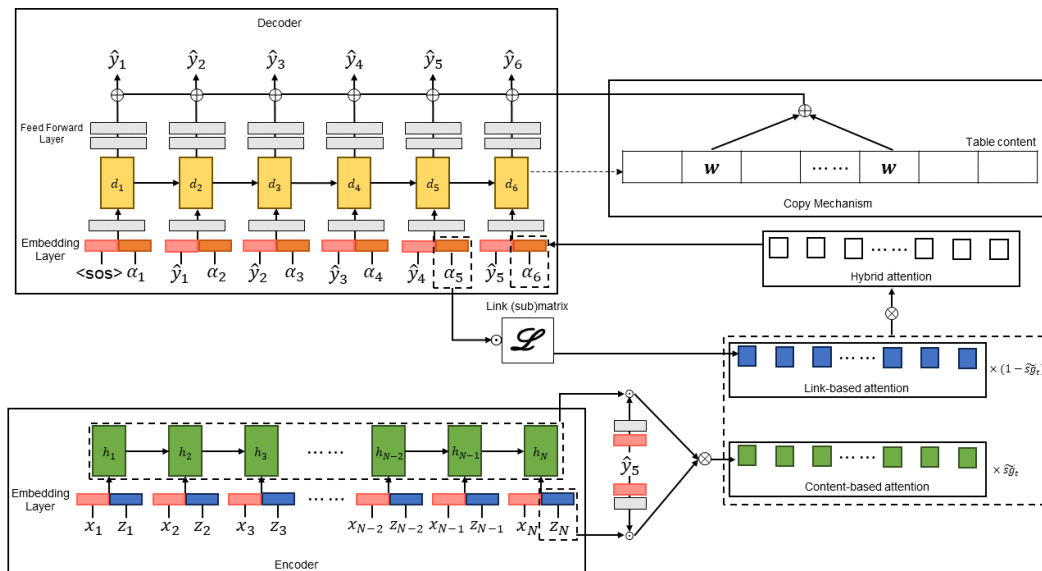


Figure 4. Order-Planning Neural Text Generation Model

미한다. 이렇게 계산된 $\alpha_{tn}^{content}$ 는 콘텐츠 기반 어텐션 점수이며 값이 클수록 이전 타임 스텝에서 생성된 단어와 관련 있음을 의미한다.

두 번째로 링크 기반 어텐션에서는 필드의 관계를 표현하기 위해 링크 행렬(Link Matrix)을 정의한다. 링크 행렬 L 은 f_N 을 학습 데이터에 있는 필드의 수라고 할 때, $f_N \times f_N$ 의 차원을 갖는 행렬이다. 이 행렬의 원소 $L[f_m, f_n]$ 는 필드 f_m 이후에 f_n 이 언급될 가능성을 의미한다. 즉, 링크 행렬은 m 번째 필드 이후에 n 번째 필드가 등장할 가능성을 의미하며 학습 과정에서 같이 학습되는 행렬이다. 링크 기반 어텐션은 이러한 링크 행렬과 이전 타임 스텝에서의 어텐션 점수를 활용하며 이는 Eq (10)과 같은 과정으로 계산한다.

$$\alpha_{tn}^{link} = \frac{\exp\left\{\sum_{m=1}^N \alpha_{t-1,m} \cdot L[f_m, f_n]\right\}}{\sum_{k=1}^N \exp\left\{\sum_{m=1}^N \alpha_{t-1,m} \cdot L[f_m, f_k]\right\}} \quad (10)$$

$\alpha_{t-1,m}$ 은 $t-1$ 번째 타임 스텝에서 m 번째 필드의 어텐션 점수를 의미하며 $L[f_m, f_n]$ 은 필드 f_m 이후에 f_n 이 언급될 가능성을 의미한다. 따라서 이전 타임 스텝에서 어텐션 점수가 높았던 m 번째 필드 이후에 n 번째 필드가 등장할 가능성이 높다면 링크 기반 어텐션 점수가 커질 것이다. 이를 통해 필드간의 관계를 학습하게 된다.

마지막으로 하이브리드 어텐션은 학습 과정에서 같이 학습되는 자가 적용 게이트를 통해 콘텐츠 기반 어텐션과 링크 기반 어텐션의 가중합으로 계산된다. 자가 적용 게이트는 콘텐츠와 링크 모두를 반영하여 Eq (11)의 과정으로 계산된다.

$$sg_t = \sigma(W[d_{t-1}; f_t^{sum}; y_{t-1}]) \quad (11)$$

$$f_t^{sum} = \sum_{n=1}^N \alpha_{tn}^{link} z_n$$

d_{t-1} 은 이전 타임 스텝에서의 디코더 은닉 상태를 의미하며, y_{t-1} 은 이전 타임 스텝에서 생성된 값의 임베딩을 의미한다. f_t^{sum} 은 링크 기반 어텐션인 α_{tn}^{link} 과 필드의 임베딩 벡터인 z_n 의 가중합을 의미한다. Eq (11)의 과정으로 계산된 자가 적용 게이트 sg_t 는 콘텐츠와 링크의 관계를 학습하게 되며 시그모이드 함수를 통해 0부터 1 사이의 값을 갖는 확률로 표현된다. 이를 통해 최종 하이브리드 어텐션이 계산되며 그 과정은 Eq (12)과 같다.

$$\alpha_{tn}^{hybrid} = \widetilde{sg}_t \cdot \alpha_{tn}^{content} + (1 - \widetilde{sg}_t) \cdot \alpha_{tn}^{link} \quad (12)$$

$$\widetilde{sg}_t = 0.2sg_t + 0.5$$

이때, Sha *et al.*(2018)은 실험결과 sg_t 값이 너무 작아 하이브리드 어텐션이 링크 기반 어텐션에 과하게 치중됨을 확인하여 실험적으로 sg_t 값을 \widetilde{sg}_t 와 같이 조정하였다. 최종적으로 Eq

(9)와 Eq (10)의 과정으로 계산된 콘텐츠 기반 어텐션과 링크 기반 어텐션에 조정된 자가 적용 게이트 \widetilde{sg}_t 를 곱해 하이브리드 어텐션 점수를 계산하게 된다.

일반적인 디코더는 이전에 생성된 단어의 임베딩을 입력으로 받아 은닉 상태를 생성하고 어텐션 점수와 인코더 은닉 상태를 가중합한 문맥 벡터와 연결하여 완전 연결 층을 지나 단어 사전의 확률 분포를 출력한다. 그러나 Sha *et al.*(2018)의 디코더는 입력으로 이전에 생성된 단어의 임베딩 뿐만 아니라 문맥 벡터를 함께 연결하여 완전 연결층을 지난 벡터를 사용한다. 또한 디코더를 거쳐 생성된 디코더 은닉 상태를 한 번 더 완전 연결 층을 통과시켜 최종 단어 사전의 확률 분포를 출력한다. 자세한 과정은 Eq (13)과 같다.

$$x_t^* = \tanh(W^*[\alpha_t; y_{t-1}])$$

$$d_t = LSTM(x_t^*, d_{t-1}) \quad (13)$$

$$p_{vocab}(w) = W_g d_t + b_g$$

α_t, y_{t-1} 은 각각 문맥 벡터와 이전 타임 스텝에서 생성된 단어의 임베딩을 의미하며 d_t 는 현재 타임 스텝에서의 디코더 은닉 벡터를 의미한다. 이렇게 계산된 $p_{vocab}(w)$ 는 현재 보유하고 있는 단어 사전의 확률분포를 의미한다.

(3) 복사 메커니즘(Copy Mechanism)

Sha *et al.*(2018)은 흔히 사용되는 방법과는 다른 복사 메커니즘을 적용하였다. 그 과정은 Eq(14)와 같으며, 복사확률인 $p_{copy}(w)$ 는 어텐션 점수를 구하는 것과 유사하게 인코더 은닉 상태와 디코더 은닉 상태의 유사도를 측정하는 방식으로 계산한다. 또한, $I_{w_n=w}$ 는 불리언 변수(boolean variable)로 단어 w 가 n 번째 입력의 값과 같은지 여부를 나타내는 변수를 의미한다.

$$p_{tn}^{copy} = \sigma(h_n W_c) d_t$$

$$p_{copy}(w) = \sum_{n=1}^N p_{tn}^{copy} \cdot I_{w_n=w} \quad (14)$$

$$p_t(w) = softmax(p_{vocab}(w) + p_{copy}(w))$$

복사 메커니즘을 통해 계산된 $p_{copy}(w)$ 는 입력 값의 확률분포를 의미하며 이를 Eq(11)에서 계산한 단어 사전의 확률분포인 $p_{vocab}(w)$ 에 더해 소프트 맥스 활성화함수를 거쳐 확장된 단어 사전의 확률 분포인 $p_t(w)$ 를 계산하게 된다. 이렇게 계산된 $p_t(w)$ 는 기존 단어 사전에 있는 단어가 입력 값과 같으면 더 높은 확률을 갖게 되며, 기존 단어 사전에 없던 단어도 확률분포를 갖게 된다.

4.3 데이터 기반 문장 생성 모델 비교

4.3에서는 Data2Text에서 더 효과적인 어텐션 방법론을 찾고자 한다. Liu *et al.*(2018)과 Sha *et al.*(2018)은 각각 듀얼 어텐

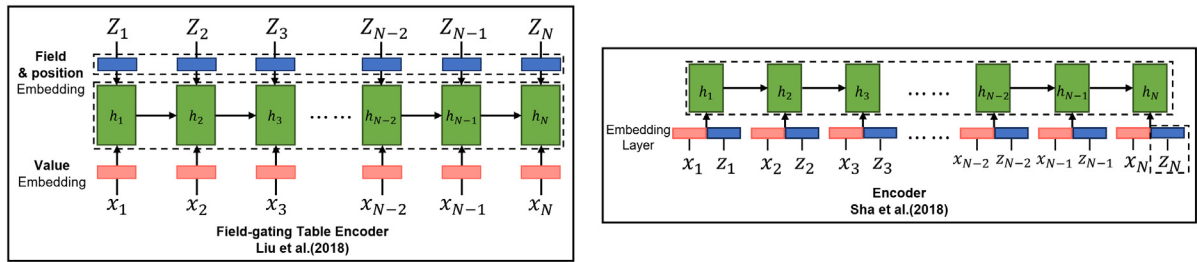


Figure 5. Comparison of Field-gating Table Encoder(Liu et al., 2018) and Encoder(Sha et al., 2018)

선과 하이브리드 어텐션을 제안하였다. 그러나 <Figure 5>와 같이 다른 입력과 인코더 구조로 인해 두 어텐션 방법론을 직접적으로 비교하기에 어려움이 있다. Liu et al.(2018)의 Structure aware Seq2Seq 모델은 입력으로 값, 필드뿐만 아니라 필드 위치 및 역위치를 활용하였으며 기존 LSTM에서 변형된 인코더인 필드 게이트 테이블 인코더를 제안하였다. 반면, Sha et al.(2018)의 Order-planning Neural Text Generation 모델은 필드 위치 및 역위치를 활용하지 않았으며 인코더 역시 일반적인 인코더를 활용하였다. 따라서 두 모델 중 더 효과적인 어텐션 방법론을 찾기 위해 동일한 조건으로 어텐션 방법론을 비교하고자 한다. 따라서, Sha et al.(2018)의 모델 구조에서 기존의 인코더를 Liu et al.(2018)의 필드 게이트 테이블 인코더로 변경하고, 필드 위치 및 역위치를 입력 데이터로 동일하게 활용하여 어텐션 방법론을 제외한 모든 구조를 동일하게 구성한다. 이로써 Data2Text에서 성능에 있어 효과적인 어텐션 방법론을 찾고자 한다.

4.4 Dual Attention Seq2Seq with Copy Mechanism

마지막으로 4.4에서는 <Figure 6>과 같이 복사 메커니즘을 적용한 모델을 제안한다. Liu et al.(2018)은 사전에 없는 단어를 출력할 수 없는 문제인 OOV(Out-of-Vocabulary) 문제를 해결하는 방안으로 듀얼 어텐션 점수를 활용하였다. 출력 단어가 사전에 없는 단어인 UNK(Unknown)인 경우 듀얼 어텐션의 점수가 가장 높은 입력 단어로 후처리하는 방법을 제안하였다. 그러나 이러한 방법은 반드시 UNK가 출력되어야 적용되며 학습이 아닌 후처리를 이용한 방법이므로 한계가 있다. 따라서 이를 보완하고자 듀얼 어텐션 점수를 활용하여 See et al.(2017)의 복사 메커니즘을 적용한다.

복사 메커니즘의 복사 혹은 생성 확률은 문맥 벡터, 디코더의 은닉 벡터 그리고 디코더의 입력 임베딩 벡터를 연결하고 완전 연결 층과 시그모이드 함수를 거쳐 계산된다. 이 과정을 통해 0부터 1의 범위를 갖는 복사 또는 생성 확률을 학습하게

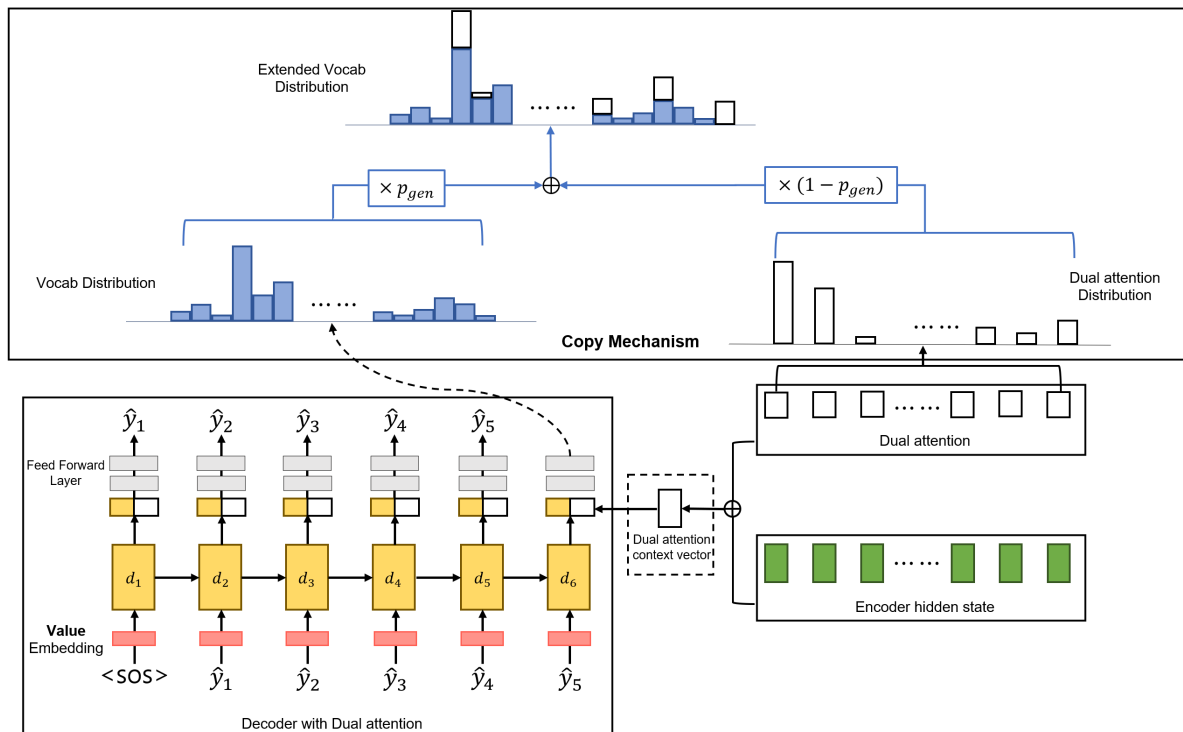


Figure 6. Dual Attention Seq2Seq with Copy Mechanism

된다. 그다음 Eq (5)와 Eq(7)의 과정으로 계산된 단어 사전의 확률 분포에 단어가 생성될 확률을 곱하고, 듀얼 어텐션 점수에 단어가 복사될 확률을 곱한다. 그리고 둘을 더해 최종적으로 입력 단어까지 포함한 확장된 단어 사전의 확률 분포를 계산하게 되며 그 과정은 Eq (15)과 같다.

$$p_{\text{gen}} = \sigma(W_{\alpha} \tilde{\alpha} + W_d d_t + W_s s_t + b_{\text{gen}})$$

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{n=1}^N \delta_{tn} \quad (15)$$

$\tilde{\alpha}$, d_t 그리고 s_t 는 각각 앞에서 구한 문맥 벡터, 디코더 은닉 벡터 그리고 디코더의 입력 임베딩 벡터를 의미하며 δ_{tn} 은 듀얼 어텐션 점수를 의미한다. 마지막으로 p_{gen} 은 단어의 생성 확률을 의미하며 이러한 과정으로 한정된 단어 사전에서 고유 명사와 같은 자주 등장하지 않는 단어도 생성할 수 있게 된다.

추가로, Chen *et al.*(2019)은 모델이 복사 해야 하는 위치를 집중하여 학습하도록 복사 손실 함수(Copy loss)를 제안하였다. Data2Text의 특성상 비슷한 도메인에서의 데이터는 유사한 위치에서 입력 단어를 출력해야 하는 경우가 많다. 따라서 본 연구에서도 복사 손실 함수를 적용해 보다 정확한 위치에서 입력 단어를 복사하여 성능을 높이고자 하였으며 최종 손실 함수는 Eq (16)과 같다.

$$L = L_o + \lambda \sum_{y_t \in V} 1 - p_{\text{gen}}^t \quad (16)$$

L_o 는 기존의 손실 함수를 의미하며 y_t , V 그리고 p_{gen}^t 는 각각 t 번째 타임 스텝에서의 타겟 단어와 단어 사전 그리고 단어의 생성 확률을 의미한다. 마지막으로 λ 는 복사 페널티(copy penalty)로 0부터 1의 값을 갖는 초 매개변수(Hyperparameter)이며 복사 손실 함수를 적용할 비율을 의미한다. 이러한 복사

손실 함수의 추가를 통해 단어를 생성해야 하는 위치에서 복사 확률을 최소로 하는 방향으로 학습하게 된다.

5. 실험결과

5.1 하이퍼 파라미터

초매개변수(Hyper parameter)의 경우 기존 선행연구의 값을 사용하였으며 선행연구의 값이 없는 경우 실험적으로 가장 좋은 성능을 보이는 값으로 설정하였다. <Table 2>는 본 연구에서 활용한 초매개변수이다.

이때 Order-Planning Neural Text Generation 모델의 경우 Liu *et al.*(2018)의 모델과는 다르게 입력으로 필드 위치 및 역위치를 사용하지 않았다. 따라서 입력을 동일하게 설정하기 위해 필드 위치 및 역위치를 활용하는 과정에서 필요한 임베딩 크기를 실험적으로 좋은 성능을 보인 50으로 설정하였다. 또한, 복사 페널티의 경우 역시 실험적으로 가장 좋은 성능을 보인 0.3으로 설정하였다.

5.2 모델 비교

<Table 3>은 Sha *et al.*(2018)의 모델에 대해서 입력 데이터 및 인코딩 구조를 Liu *et al.*(2018)의 모델과 동일하게 설정하고, Sha *et al.*(2018)의 복사 메커니즘을 제외하여 성능을 비교한 결과이다. 필드와 필드 위치정보를 입력으로 추가로 활용하였을 때와 필드 게이트 테이블 인코더를 활용하였을 때 성능이 향상되었다. 그러나 Liu *et al.*(2018)의 듀얼 어텐션 모델이 BLEU-4점수와 ROUGE-4-F1점수에서 모두 더 좋은 성능을 보였다. 따라서 Data2Text에서 사람의 글 쓰는 방법을 반영한 하이브리드 어텐션보다 표의 구조를 이해하는 데 강점을 갖는 듀얼 어텐션 구조가 더 효과적임을 확인하였다.

Table 2. Hyperparameter Configuration

	Structure aware Seq2Seq (Liu <i>et al.</i> , 2018)	Order-Planning Neural Text Generation (Sha <i>et al.</i> , 2018)
Hidden size		500
Embedding size		400
Field Embedding size	50	400
Position Embedding size	5	50
Batch size		32
Vocab size		20003
Field Vocab size		1480
Position Vocab size		31
Learning rate		0.0005
Optimizer		Adam
Copy Penalty		0.3

Table 3. BLEU-4 and Rouge-4-F1 for Order-planning Neural Text Generation(Sha *et al.*, 2018) with Field-gate Table Encoder

	BLEU-4	ROUGE-4-F1
Hybrid Attention (Order-planning Seq2Seq)	43.004 ± 0.2024	38.205 ± 0.2061
+ field + pos	43.4861 ± 0.2118	38.3287 ± 0.2132
+ field-gate	43.6038 ± 0.1923	38.6039 ± 0.1997
Dual Attention (Structure aware Seq2Seq)	44.4964 ± 0.1044	40.2843 ± 0.1044

<Table 4>는 Seq2Seq 모델에 복사 메커니즘을 적용하였을 때의 성능을 비교한 결과이다. 상위 3줄은 복사 메커니즘을 적용하지 않은 모델의 성능이며 하위 3줄은 복사 메커니즘을 적용한 모델의 성능을 의미한다. 또한 ±는 성능의 평균에 대한 표준편차를 의미한다. Seq2Seq 기반 모델에 복사 메커니즘을 적용하였을 때 듀얼 어텐션을 적용한 모델과 비슷한 수준까지 성능이 향상됨을 확인하였다. BLEU-4기준 입력에 필드 위치 및 역위치를 함께 활용한 모델에서 3.58% 그리고 필드-게이트 테이블 인코더를 적용한 모델에서 1.42%의 성능향상을 보였다. 그러나 듀얼 어텐션을 적용한 모델에서는 성능이 소폭 감소하는 것을 확인하였는데, 0.2%의 차이로 오차범위 내의 차이였다. 또한, BLEU와 ROUGE 평가지표가 N그림이 겹치는 수를 반영한 평가지표이므로 단순 수치만으로 평가하기에는 어려움이 있다. 때문에 정성 평가를 함께 진행하였으며 <Table

6>의 결과와 같이 특정 단어를 반복하는 문제에 대해서 강건한 모습을 확인할 수 있었다. 추가적으로 차후 파라미터가 더 많은 모델인 트랜스포머(Transformer)(Vaswani *et al.*, 2017) 및 사전 학습 기반 모델에 복사 메커니즘을 적용할 시 성능향상을 기대할 수 있을 것으로 예상된다.

마지막으로 <Table 5>는 복사 메커니즘 적용 여부에 따른 예측 결과이다. 먼저, 복사 메커니즘을 적용한 모델이 “hubert de la poer gough” 라는 입력으로부터 주어지는 이름을 잘 출력한 것을 확인할 수 있다. 또한 복사 메커니즘을 적용하지 않은 모델의 경우 “군사대학의 지휘관”(commandant of the royal military college), “영국 해군”(royal navy) 그리고 “영국 공군”(royal air force) 등의 맞지 않는 정보에 대한 단어를 생성한 것을 확인하였다. 이에 반해 복사 메커니즘을 적용한 모델의 경우 정답 문장과 비교적 비슷한 정보를 갖는 문장을 생성한 것을 확인할 수 있다.

Table 4. BLEU-4 and Rouge-4-F1 for structure-aware seq2seq model(Liu *et al.*, 2018) and Model with Copy mechanism

	BLEU-4	ROUGE-4-F1
Seq2Seq + field + pos	42.8171 ± 0.1044	38.2472 ± 0.1072
+ field-gate	43.7533 ± 0.1043	39.4515 ± 0.1072
+ dual attention	44.4964 ± 0.104	40.2843 ± 0.1068
Seq2Seq + field + pos + copy	44.3493 ± 0.1031	40.2021 ± 0.1065
+ field-gate + copy	44.3751 ± 0.1042	40.0465 ± 0.1072
+ dual attention + copy	44.4024 ± 0.1039	40.0286 ± 0.107

Table 5. Comparison of Generated Descriptions with or without Copy Mechanism

Label	Generated Description
	general sir hubert de la poer gough, gcb, gcmg, kcvo -lrb- 12 august 1870 - 18 march 1963 -rrb-, was a senior officer in the british army in the first world war.
Seq2Seq + field + pos	general sir hubert gough, kcmg -lrb- 12 august 1870 - 18 march 1963 -rrb- was a british army officer who became commandant of the royal military college, lions.
+ field-gate	general sir hubert thomas gough gcb -lrb- 12 august 1870 - 18 march 1963 -rrb- was an officer in the royal navy who served in the royal navy during the early nineteenth century.
+ dual attention	general sir hubert de la poer gough -lrb- 12 august 1870 - 18 march 1963 -rrb- was a british army officer who served in the royal air force, royal air force, royal air force, royal air force, royal cross, the royal observer regiment, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, royal air force, and royal observer rifles.
Seq2Seq + field + pos + copy	general sir hubert de la poer gough gcb -lrb- 12 august 1870 - 18 march 1963 -rrb- was a british army officer who became governor general of the british army.
+ field-gate + copy	general sir hubert de la poer gough vc, gcb -lrb- 12 august 1870 - 18 march 1963 -rrb- was a british army officer who became general officer commanding command.
+ dual attention + copy	general sir hubert de gough gcb -lrb- 12 august 1870 - 18 march 1963 -rrb- was a senior british army officer during the first world war.

- (2021), Technical Trends of AI Military Staff to Support Decision-Making of Commanders, *Electronics and Telecommunications Trends*, **36**(1), 89-98.
- van der Lee, C., Krahmer, E., and Wubben, S. (2018, November), Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods, In *Proceedings of the 11th International Conference on Natural Language Generation*, 35-45.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014), Sequence to sequence learning with neural networks, In *Advances in Neural Information Processing Systems*, 3104-3112.
- Lin, C. Y. (2004, July), Rouge: A package for automatic evaluation of summaries, In *Text summarization branches out*, 74-81.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W. J. (2002, July), Bleu: A method for automatic evaluation of machine translation, In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311-318.
- Liu, T., Wang, K., Sha, L., Chang, B., and Sui, Z. (2018, April), Table-to-text generation by structure-aware seq2seq learning, In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Puduppully, R., Dong, L., and Lapata, M. (2019, July), Data-to-text generation with content selection and planning, In *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**(1), 6908-6915.
- Luong, M. T., Pham, H., and Manning, C. D. (2015), Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025.
- Mei, H., Bansal, M., and Walter, M. R. (2015), What to talk about and how? selective generation using lstms with coarse-to-fine alignment, arXiv preprint arXiv:1509.00838.
- Sha, L., Mou, L., Liu, T., Poupart, P., Li, S., Chang, B., and Sui, Z. (2018, April), Order-planning neural text generation from structured data, In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- See, A., Liu, P. J., and Manning, C. D. (2017), Get to the point: Summarization with pointer-generator networks, arXiv preprint arXiv:1704.04368.
- Gu, J., Lu, Z., Li, H., and Li, V. O. (2016), Incorporating copying mechanism in sequence-to-sequence learning, arXiv preprint arXiv:1603.06393.
- Chen, Z., Eavani, H., Chen, W., Liu, Y., and Wang, W. Y. (2019), Few-shot NLG with pre-trained language model, arXiv preprint arXiv:1904.09521.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,... and Polosukhin, I. (2017), Attention is all you need, *Advances in Neural Information Processing Systems*, **30**.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019), Language models are unsupervised multitask learners, *OpenAI blog*, **1**(8), 9.
- Li, J., Zhang, C., Chen, X., Cao, Y., Liao, P., and Zhang, P. (2019, July), Abstractive text summarization with multi-head attention, In *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1-8.
- Fan, A., Lewis, M., and Dauphin, Y. (2018), Hierarchical neural story generation, arXiv preprint arXiv:1805.04833.

저자소개

김영석 : 동국대학교 통계학과에서 2021년 학사학위를 취득하고 고려대학교에서 산업경영공학과 석사과정에 재학 중이다. 연구 분야는 NLP, Recommendation System, Deep Learning이다.

이병훈 : 서울과학기술대학교 산업정보시스템학과에서 2021년 학사학위를 취득하고, 고려대학교에서 산업경영공학과 석·박사과정에 재학 중이다. 연구 분야는 Machine Learning, Self-supervised Learning, Multimodal Deep Learning이다.

강민지 : 홍익대학교 산업공학과에서 2019년 학사학위를 취득하고 고려대학교 산업경영공학과에서 2022년 석사학위를 취득하였다. 연구 분야는 Computer Vision, Deep Learning이다.

한성원 : 고려대학교 산업 시스템 정보 공학과에서 2003년 학사학위를 취득하였다. Georgia Institute of Technology에서 2006년 Operation Research 석사학위, 2007년 Statistics 석사학위, 2010년 Mathematics 석사학위를 취득하고 2010년 Industrial Engineering 과 Statistics 박사학위를 취득하였다. University of Pennsylvania, Department of Biostatistics and Epidemiology에서 Post-doctoral Researcher(2010.07-2012.06), Hoffmann-La Roche Inc., Department of Non-clinical Safety에서 Post-doctoral Fellow(2012.07-2013.08), New York University, Department of Population Health에서 Research Scientist(2013.08-2015.12), New York University, Department of Population Health에서 Senior Research Scientist(2016.01-2016.02)을 역임하고, 2016년부터 고려대학교 산업경영공학부 교수로 재직하고 있다. 연구 분야 중 방법론 분야는 probabilistic graphical model, network analysis, deep learning 등이 있으며, 응용 분야로는 바이오 의료, 소재 정보학, 품질 모니터링, 텍스트 마이닝이 있다.