|  |
| --- |
| 碩士學位 請求論文 (표지)  指導敎授 李 知 炯 |
|  |
| 문맥을 따른 문장 생성을 위한 언어 학습 모델 |
|  |
| 成均館大學校 一般大學院  DMC 工學科  朴 景 韓 |

|  |
| --- |
| 碩士學位 請求論文 (내표지)  指導敎授 李 知 炯 |
|  |
| 문맥을 따른 문장 생성을 위한 언어 학습 모델  Language learning model for context-based sentence generation |
|  |
| 成均館大學校 一般大學院  DMC 工學科  朴 景 韓 |

|  |
| --- |
| 碩士學位 請求論文 (심사청구서)  指導敎授 李 知 炯 |
|  |
| 문맥을 따른 문장 생성을 위한 언어 학습 모델  Language learning model for context-based sentence generation |
| 이 論文을 工學 碩士學位請求論文으로 提出합니다.  2017 年 4 月 日 |
|  |
| 成均館大學校 一般大學院  DMC 工學科  朴 景 韓 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | (인정서) | |  |
|  | 이 論文을 朴景韓의 工學  碩士學位 論文으로 認定함.(16p) | |  |
|  |  | |  |
|  | 2017 年 6 月 日 | |  |
|  |  | |  |
|  |  | 審査委員長 김 문 현 |  |
|  |  |  |  |
|  |  | 審査委員 정 윤 경 |  |
|  |  |  |  |
|  |  | 審査委員 이 지 형 |  |

목차

**목 차 ⅰ**

**표 목차ⅲ**

**그림 목차ⅳ**

**논문 요약ⅴ**

**제 1장 서론1**

1-1. 연구 배경과 연구 방법론1

A. 연구 배경1

B. 연구 목적과 방법2

**제 2장 배경 지식 및 관련 연구4**

2-1. 배경 지식4

A. RNN(Recurrent Neural Network) 4

B. LSTM(Long Short Term Memory Network) 6

A. RNN(Recurrent Neural Network) 9

2-2 관련 연구11

A. 문장 표현을 위한 방법 연구 11

A. 문맥 정보를 이용하여 문장을 생성하는 방법 연구 13

**제 3장 문맥에 따른 문서 생성을 위한 언어 학습 모델 제안14**

3-1 문맥의 흐름을 인식하고 학습할 수 있는 방법14

3-2 문맥에 따른 문서 생성 모델16

A. 문장을 표현하는 문장 벡터 생성 방법16

B. 문장 백터 클러스터링20

C. 문맥 흐름 생성 학습 모델 23

3-3 문장 생성 모델26

**제 4장 실 험29**

4-1. 실험 데이터 및 평가 척도 29

A. 실험 데이터 분석 척도 29

B. 평가 척도 척도 30

4-2. 실험 방법 및 결과31

A. 실험 방법31

B. 실험 결과32

**제 5장 결 론35**

**참고 문헌37**

**Abstract 40**

**표목차**

Table 1 학습을 위한 자료 설명 ........................................................................ 30

Table 2 Seed 문장을 통해 생성된 문장 예제...................................................... 33

Table 3 주제별 생성된 문장과 요약문 사이의 유사도 비교 ............................. 34

**그림목차**

그림. 1 Recurrent Neural Network Model …........................................................ 5

그림. 2 Structure of RNN .................................................................................... 7

그림. 3 Structure of LSTM ................................................................................. 8

그림. 4 Sequence to Sequence Model ...............................................................10

그림. 5 문서 정보를 고려한 언어 모델 ................................................................13

그림. 6 RNN Encoder Decoder Model ...............................................................19

그림. 7 문서 내 문장들의 클러스터링 .................................................................22

그림. 8 클러스터링 시퀀스를 구하기 위한 RNN 모델 ........................................25

그림. 9 문맥을 고려한 문장 생성 RNN 구조 ....................................................... 27

**논문요약**

**논 문 제 목**

본 논문에서는 문맥에 맞는 문장 생성을 위한 언어 학습 모델을 제안한다. 자연어 처리 연구는 다방면에서 진행되었고, 기계 번역, 문서 요약, 감성 분석 등 많은 분야에 진전을 이뤄왔다. 뿐만 아니라 주어진 텍스트를 학습한 뒤 이를 이용하여 새로운 문장을 생성하는 연구로 발전하고 있다. 하지만 개별적인 단어의 일반적인 패턴을 학습하기 때문에 주제나 의도를 갖은 문장을 생성하기 어렵다는 한계가 있다. 이러한 문제를 해결하기 위하여 주어진 텍스트의 문맥을 분석하고 이에 기반하여 문장을 생성하는 개선된 문장 생성 방안을 제시한다. 본 논문에서는 문맥에 근거한 문장을 생성하기 위해 크게 세 단계 방법을 제시한다. 첫 번째로 문장의 의도를 분석하기 위해 주어진 문장을 표현할 수 있는 문장 벡터 생성 방법을 제시한다. 두 번째로 생성된 문장 벡터들 사이의 관계성을 분석하기 위하여 문장 벡터를 의미 별로 분류하고 이를 기반으로 주어진 문장간의 관계를 분석 함으로써 문맥을 이해하는 방법을 제시한다. 두 번째 방법을 통해서 주어진 텍스트의 문맥을 학습할 수 있고 이에 근거하여 문맥을 생성할 수 있다. 세 번째로 주어진 문맥과 이전 문장간의 관계를 분석하여 문맥에 맞는 문장을 생성할 수 있는 방법을 제시하였다.

**주제어: Language learning model, Text generation, Sentence context, RNN encoder decoder, RNN**

**제 1장 서 론**

**1-1. 연구배경과 연구방법론**

A. 연구배경

최근 다양한 딥 러닝 기법이 발달하면서 자연어 처리 연구 또한 많은 발전을 이루었다. 딥 러닝에 기반한 자연어 처리는 기존의 확률 기반이나 규칙 기반의 접근보다 언어 모델의 학습이 용이하고 성능 면에서 좋은 성과를 보여주었다. 그 결과 기계 해석, 문서 요약, 문서 감성 분류 등 기존에는 어려웠던 분야에서 발전을 이루었고 다양한 문제를 해결하는데 응용이 되고 있다[1, 2, 3]. 또한 딥 러닝을 이용한 자연어 처리 연구는 비단 기존 텍스트를 분석하는 것뿐 아니라, 기존 텍스트를 학습한 뒤 이를 이용하여 새로운 문장을 생성하는 방향으로도 전개되었다. 문서 내 단어 사이의 규칙이나 관계 혹은, 출현 빈도 등을 이용하여 언어 모델을 생성 후, 생성된 언어 모델을 이용하여 본 텍스트와 유사한 글을 쓰는 연구 또한 진행되고 있다[4]. 이러한 연구는 딥 러닝을 통한 창작 활동이라는 면에서 흥미로울 뿐 아니라, 다양한 분야로 확장될 수 있는 가능성을 갖고 있다.

하지만, 딥 러닝에 기반한 일반적인 언어 모델 대부분은 단어 사이의 일반적인 관계를 학습하여 문장을 생성하기 때문에 주제나 의도를 갖고 문장을 생성하기 힘들다는 한계가 있다. 즉 글쓰기의 기본적인 개념인, 문맥과 주제의 흐름에 따른 문장의 생성이라는 관점에서 볼 때 기존의 딥 러닝 연구에 기반한 글쓰기는 개별적인 단어의 관계에 의한 문장 생성이라는 한계를 갖고 있다. 즉 글쓰기란 글을 쓰는 목적을 이해하고 흐름에 맞는 문장을 생성 및 배치하여야 하는데, 일반적인 모델로는 전체적인 문맥의 흐름이란 정보를 학습하기가 힘들기 때문이다.

B. 연구 목적과 방법

글의 맥락에 맞는 글을 쓰는 것은 일정한 문맥의 흐름에 따라서 필요한 문장 및 단어를 생성하는 것이다. 글에는 글 전체에 흐르는 일정한 문맥이 있고, 이 문맥에 따라 문장을 선택하거나 단어를 선택하여 글을 완성한다. 딥 러닝을 이용해서 주제 혹은 일정한 문맥에 맞는 글을 쓰기 위해서는 학습의 대상이 되는 텍스트의 문맥의 흐름을 학습하고, 또한 주제에 맞는 적절한 문장을 선택 및 생성하는 방법을 학습하는 것이 필요하다. 즉 단편적인 단어 사이의 관계 및 출현 빈도 등의 학습을 넘어서 글 전체적으로 흐르는 문맥을 학습하고 이를 근거로 주제에 맞는 글쓰기를 위한 언어 모델을 생성해야 한다. 본 논문에서는 텍스트의 문맥의 흐름을 학습할 수 있고, 또한 이를 기반으로 문장을 학습할 수 있는 방법을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안 방법의 배경 지식인 RNN, LSTM 및 RNN Encoder-Decoder과 언어 생성 모델과 관련된 연구에 대해서 살펴본다. 3장에서는 본 논문에서 제안하는 문맥의 흐름에 따른 문서 학습 및 생성 모델에 대해서 설명하고, 4장에서는 제안 방법의 실험 방법 및 실험 결과를 기술한다. 마지막 5장에서는 결론 및 향후 연구에 대해서 서술한다.

**제 2장 배경 지식 및 관련 연구**

본 장에서는 인공 신경망 중 자연어와 같은 시퀀스 데이터를 처리하기 위해 사용되는 LSTM- RNN(Long Short Term Memory Network - Recurrent Neural Network)에 대한 설명과 이를 응용한 RNN Encoder-Decoder 모델에 대해서 설명한다. 또한 LSTM-RNN과 RNN의 특징을 이용하여 주제 및 문맥에 맞는 문장을 생성하는 연구들에 대해서 설명한다.

**2-1 배경 지식**

A. RNN(Recurrent Neural Network)

자연어 처리에 필요한 인공 신경망 모델 중 하나인 RNN(Recurrent Neural Network)은 순환적인 구조를 가지는 인공 신경망이다. 모든 입력과 출력이 독립적으로 동작하는 기존 신경망과는 다르게 내부에 현재 상태를 저장하는 공간이 있어서 출력이 이전 계산 결과에 영향을 받는 순환적인 구조를 갖고 있다. 순환적인 구조로 인해서 RNN은 순차적인 정보를 처리하는데 적합한 구조를 가지며, 음성 인식, 번역, 이미지 설명, 순차적 데이터를 처리하는데 유리하다[5]. 그림 1 은 RNN 의 구조를 나타내고 있으며, 왼쪽에 있는 그림은 RNN의 본래 구조이며, 오른쪽의 그림은 순환적인 구조를 펼쳐 보인다.



그림. 1 Recurrent Neural Network Model

는에서의 입력을 나타내며, 는 내부의 상태 값을 나타내며, 식(1) 와 같이 계산된다. 상태는 시간 에서의 상태 값과 시간의 입력 값에 의해서 결정되며, 함수는 , 혹은 와 같은 비선형 함수가 사용된다.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

식(1) 와 같이 상태 값는 다음 계산을 위해 인공 신경망으로 다시 입력이 되는 순환적인 구조를 가지며, 이 점을 고려하여 그림 1 의 오른쪽과 같이 펼쳐 놓으면, RNN이 일반적인 인경 신경망 구조처럼 생각될 수 있다. 그림 1 의 *W, U, V*는 인공 신경망이 데이터를 통해 학습하면서 결정되는 가중치들이며, 상태 값 와 결과값 는 각각 *W, V*의 가중치에 영향을 받는다. 는 시간 에서 출력을 나타내며, 다음과 수식(2) 와 같이 표현될 수 있다.

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

시간에서의 결과 값인 는 내부 상태 값인 값과 가중치에 의하여 결정된다. RNN 은 순차적으로 들어오는 입력 값 을 이용해서 내부 상태 값인 를 지속적으로 변경하고, 또한 를 이용하여 결과 값 o를 출력한다. 즉 RNN 은 기존 인공 신경망과 다르게 임의의 길이를 가진 입력을 받을 수 있으며 내부 상태 값을 사용하여 과거 입력 값이 지속적으로 미래의 결과에 영향을 미칠 수 있는 특징이 있다.

B. LSTM(Long Short Term Memory Network)

이전 정보를 활용할 수 있다는 점에서 RNN은 일반 인공 신경망 구조에 비해 큰 장점을 갖고 있지만, 입력 데이터가 길어지게 되다면, 과거의 입력이 미래의 결과에 영향을 미치지 못할 수 있다는 단점이 존재한다. RNN 은 학습 시, 인공 신경망 학습에 사용되는 back propagation 연산과 유사한 BTPP(Back-propagation through time) 이라는 연산을 사용하여 각 필요한 파라미터의 값을 구한다. 하지만, 입력과 출력의 레이어 간격의 크기가 커질수록, 의존성이 약해지는 vanishing gradient 라는 문제가 있으며, LSTM이 이러한 문제를 해결하였다[6]. 일반적인 RNN 구조는 그림 2 와 같은 구조가 반복되는 형태를 가진다. 입력 값과 상태 값이 각각 연산을 거친 후 *tanh*연산을 거친 후, 이 결과가 다음 모듈로 연결되는 비교적 간단한 구조를 가진다. 하지만 시간이 많이 지난 입력 값과 상태 값을 기억할 수 있는 부가적인 정보가 존재하지 않는다.



그림. 2 Structure of RNN[17]

LSTM 도 RNN과 유사하게 하나의 연산 모듈이 반복적으로 사용되는 구조를 갖고 있지만 다른 모듈 구조를 갖고 있다. LSTM이 일반 RNN과 다른 큰 차이점은 한 개의 상태 값을 유지하는 RNN과 달리 두 개의 상태 값을 갖고 있고, 이 상태 값을 변형하는 여러 개의 gate들이 있다는 점이며, 두 개의 상태 값과 게이트들은 장기 의존성 문제를 해결하기 위해서 사용된다. 그림 3의 두 개의 상태 값 중 상단에 있는 Cell state는 LSTM셀을 거치면서 작은 변화만을 가지면서 다음 레이어에 전달되는 상태 값이다. 이것은 장기 의존성을 해결하기 위한 방법 중 하나로써 여러 레이어를 지나도 과거 값이 잊혀지지 않도록 하는 목적이 있다.

LSTM 내부의 각 연산 *sigmoid*연산을 거치며, *sigmoid*연산은 각 게이트의 연산 결과가 상태에 미치는 영향의 유무를 결정하게 된다. *sigmoid*연산은 0과 1사이의 값을 출력하며, 연산의 결과가 0에 가까운 결과가 나온다면 해당 입력은 LSTM셀에 영향을 거의 미치지 않고, 1에 가까운 값이 출력된다면 해당 입력은 LSTM에 원하는 변화를 끼칠 수 있다.



그림. 3 Structure of LSTM[17]

C. RNN(Recurrent Neural Network) Encoder-Decoder

RNN Encoder-Decoder 모델은 RNN 네트워크 두 개를 연속적으로 연결한 모델로써 RNN 모델 각각 인코더와 디코더 역할을 한다. RNNED 모델은 RNN 모델이 갖고 있는 한계를 극복하기 위한 방법이며, 기계 변역과 같은 특수한 경우에 사용된다[1, 7]. 일반 RNN 모델은 입력과 출력의 길이가 같아야만 RNN 연산이 가능하다는 한계가 있다. 즉 다른 두 언어를 번역할 때 각 언어마다 필요한 단어의 수가 다르기 때문에 일반적인 RNN 을 사용하여 기계 학습을 구현하는 것은 매우 어려운 일이다. 그러나 RNN Encoder-Decoder은 두 개의 RNN이 각각 입력과 출력을 받고, 상태 백터를 전달함으로써 문제를 해결했으며, 문장의 길이가 다른 서로 다른 언어를 번역할 수 있게 한다.

그림 4은 Sequence to Sequence 모델의 구조로써 RNN Encoder-Decoder 모델을 통해서 구현할 수 있다. 그림 4에서는 다른 길이의 입력 *(A,B,C)*을 받아서 *(W,X,Y,Z)*을 생성하는 문제로 RNN Encoder-Decoder 모델을 통해서 구현할 수 있다.

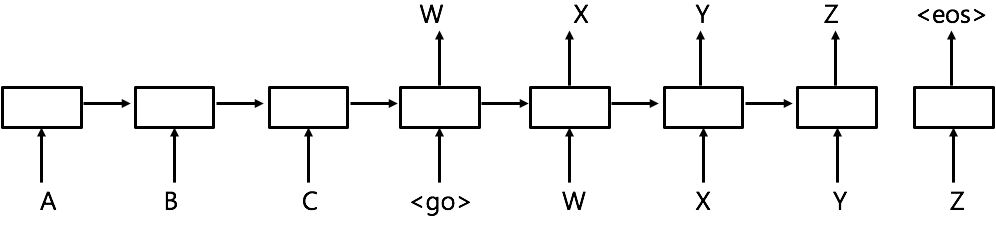


그림. 4 Sequence to Sequence Model[1]

문장이 인코더에 입력되면, 임의의 길이 입력은 RNN내부 상태 차원인 *n*-차원의 고정된 길이의 벡터로 인코딩이 수행되며, 상태 값은 일반 RNN와 같이 수식 (3)을 통해 결정된다.

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

인코더를 통한 인코딩 결과는 인코더 RNN의 마지막 상태 값이며 인코더의 길이가 *T* 일 때, 수식 (4)로 표현될 수 있다.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

수식(4) 에서 은 문장 벡터를 의미하며, 하나의 입력이 인코더를 통해 인코딩 된 결과를 의미하며 그 값은 RNN의 마지막 상태 값이다. 디코더는 초기 상태 값과 출력에서 나온 출력 벡터를 이용하여 결과를 생성한다.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

디코더의 상태 값은 인코더에서 생성된 벡터 값과 이전 상태 값, 그리고 이전 스텝에서 출력된 결과 값에 의해서 결정된다. 수식(5)에서 은 인코더의 마지막 상태 값이므로 디코더는 인코더의 길이에 영향을 받지 않고 마지막 상태 값에 의해서만 영향을 받게 된다[1].

**2-1 관련 연구**

본 절에서는 문맥에 맞는 문장 생성을 위하여 문장의 표현하기 위해 진행되었던 연구에 대해서 살펴본 뒤, 단어 사이의 관계 정보뿐 아니라 기타 추가적인 정보를 사용하여 문장을 생성했던 기존의 관련 연구를 살펴본다.

A. 문장 표현을 위한 방법 연구

문장은 임의의 길이를 가진 데이터이며, 임의의 길이를 가진 데이터를 이용하여 연구를 하기는 어려우므로 문장을 다른 연구에 활용하고자 한다면, 이를 고정된 형태로 표현할 수 있는 방법이 필요하다. 자연어 처리에서 문장을 일정한 길이의 값으로 변환하는 것은 많은 연구에서 진행되었다. 기계 번역이나, 이미지 캡셔닝, 그리고 대화 시스템 등 다양한 연구 및 시스템에서 문장을 활용하기 위해 문장 벡터를 만들어서 활용하였다[2, 8, 9]. 문장 벡터를 만드는 방법은 연구의 목적과 응용에 따라 다르며, 주어진 데이터에 따라 지도 혹은 비지도 학습으로 나뉜다.

Choi 외 2인은 문장을 표현하기 위하여 RNN Encoder-Decoder 를 사용하여 임의의 길이의 문장을 고정된 길이의 벡터로 사상하였으며 사상된 문장 벡터를 이용해서 스토리 텔링에 활용하였다[10]. RNN Encoder-Decoder 를 통해서 벡터를 생성하기 위해서 Encoder 의 입력과 Decoder 의 출력을 같은 문장으로 사용되었으며, 결과 문장 벡터는 고정된 길이를 가지면서 문장을 표현하는 유일한 값으로 활용되었다.

Palangi외 7인은 웹 문서에 존재하는 문장의 문장 임베딩 값을 구하기 위하여 LSTM-RNN 모델을 사용하며 사용자의 로그를 이용하는 약한 지도 학습 모델을 제안하였다[11]. 제안 모델을 통해서 문장의 중요한 정보만 문장 벡터에 포함되도록 모델을 제안했으며 문장 벡터만을 통해 문서의 정보를 추출하는 등의 응용에서 우수한 성능을 보였다. 문장 벡터를 생성하는 많은 연구에서 RNN 을 활용하거나 RNN 의 변형된 버전을 사용하여 문장에서 문장벡터로 변환하였다. 이는 문장의 길이가 일정하지 않은 특성에 기인한 것이고, 이를 효과적으로 처리하기 위해서 가변 길이 처리에 용이한 RNN 모델을 사용하여 모델을 제안한 것이다.

B. 문맥 정보를 이용하여 문장을 생성하는 방법 연구

Kim 외 1인은 문서 주제에 따른 문장 생성을 위하여 단어의 문맥 정보를 포함한 Many-to-one 모델과, 각 단어를 학습 시 문서에 대한 정보를 같이 학습 시키기 위하여 그림 5 과 같은 주제를 제안하였다[12]. 그림 5 에 나타나 있듯이 LSTM 에 입력되는 정보는 단어 정보 *w* 이외에 문서 정보인 *d*가 추가로 입력되어 있다. 이는 같은 단어라 하더라도 문서 정보에 따라 학습이 달라짐을 의미하며, 단어 생성 시 단어 정보 이외에 문서 정보에 의해서 영향을 받게 된다. 그 결과 기본 구조인 Many-to-one 모델보다 주제와 관련성의 측면에서 뛰어난 보습을 보였으며 문장의 표현력에서도 우수한 모습을 보였다.



그림. 5 문서 정보를 고려한 언어 모델[12]

**제 3장 문맥에 따른 문서 생성을 위한 언어 학습 모델 제안**

본 장에서는 문서의 문맥을 학습하고 이를 이용하여 문맥에 맞는 글쓰기를 할 수 있는 언어 학습 모델을 제안한다. 첫 번째 단계는 문서 내 문장들이 관계를 분석하여 이를 문맥의 흐름으로 인식하고 학습할 수 있는 방법에 대해서 제안하고 두 번째 단계에서는 이를 이용한 문서 생성 모델을 제시한다. 세 번째 단계에서는 문서 생성 모델에 기반하여 문장을 학습하고 생성하는 문장 생성 모델을 제안한다. 네 번째 단계에서는 문장을 학습하지 않고, 클러스터 정보와 이전 문장 정보를 활용해서 문장을 선택하여 문서를 생성하는 모델을 제시한다.

**3-1 문맥의 흐름을 인식하고 학습할 수 있는 방법**

문맥은 글이나 문장에 표현된 의미의 앞뒤 연결을 의미한다 [13]. 또한 문장과 문장이 이어지면서 전달되는 중심적인 의미나 논리적 연관 관계를 의미한다 [13]. 즉 문맥에 맞는 문서를 생성한다는 것은 문장 사이의 논리적 연관성이 충분히 있고, 내용적으로 연결이 되어 있는 문장이 포함된 문서를 생성하는 의미이다. 문서 안의 문맥은 문장들 사이의 연관성으로 이해할 수도 있지만 반대로 글에 흐르는 문맥이 있고 그에 적합한 문장을 선택하는 것으로 이해할 수도 있다. 이렇게 생각한다면, 우선 문서에 흐르는 문맥의 흐름을 학습한 뒤 이 학습된 정보를 이용하여 해당 문맥을 만족하는 문장을 생성하면 문서에 흐르는 문맥에 따른 문서 생성이 가능할 것이다.

문서는 서로 다른 의미를 갖는 다수의 문장으로 구성되어 있지만, 그 중 여러 문장이 서로 유사한 의미를 갖기도 한다. 문맥이란 문장들의 의미적 논리 관계를 의미하므로, 비록 똑같은 문장은 아니지만 의미가 유사한 문장으로 이루어진 의미적 논리 관계를 갖는 문장들의 집합은 서로 비슷한 문맥을 갖고 있다고 할 수 있다. 이 유사한 문장들의 집합은 그 의미가 정확하게 같지는 않지만, 비슷한 의미를 가지므로 하나의 의미 집합으로 간주될 수 있다.

그러므로 여러 문서에 포함되어 있는 의미가 유사한, 즉 하나의 의미 집합으로 간주된 문장들의 의미적 논리 관계를 파악하면 여러 문서에 흐르는 문장들의 개략적인 문맥을 파악할 수 있고, 이를 학습하여 필요한 문맥을 생성할 수도 있다.

문장들의 의미 집합이 확인되면 문서 내에 순차적으로 나열되어 있는 문장들은 그 문장이 속하는 의미 집합으로 치환될 수 있고, 하나의 문서는 그 의미 집합의 연속적인 나열로 표현될 수 있다. 즉 다수의 문장으로 구성된 문서는 의미 집합의 나열로 구성될 수 있고, 이 나열은 일정한 문맥 혹은 의미의 흐름으로 이해될 수 있다. 그리고 개별 문장은 그 의미의 흐름을 표현하기 위해 비슷한 의미를 가진 문장들 가운데 가장 적합한 문장으로 선택되어 생성된다고 볼 수 있다.

**3-2 문맥에 따른 문서 생성 모델**

주어진 문서의 문맥의 흐름을 학습한 뒤, 학습에 기반하여 문맥을 흐름을 생성할 수 있는 문서 모델을 제안한다. 이를 위해서, 주어진 글의 문장을 분석한 뒤, 각 문장이 어떤 의미를 갖고 있는지 파악하여 문장들을 유사한 의미 그룹으로 분류한다. 그렇다면 각 문장은 그 의미가 포함되는 그룹으로 나타낼 수 있고, 그 결과를 이용해서 각 문서를 순차적인 문장의 나열에서 의미 그룹의 나열로 치환 할 수 있다. 각 문장을 의미에 따라 군집화 하고, 이를 이용해서 문서를 문맥의 흐름으로 인식하는 이 과정을 문서 모델이라 부를 수 있다.

A. 문장을 표현하는 문장 벡터 생성 방법

문맥의 흐름을 학습하기 위해서는 문서 내 문장들의 의미적 논리적 구성을 파악해야 한다. 문서 내 단어의 사용 빈도 및 단어 사이의 관계의 패턴을 학습하여 문장을 생성하는 모델에서는 문서 내 문맥의 흐름을 파악하기 힘들다. 기존 모델 은 개별 단어가 출현하기 위해 필요한 사전 조건을 학습하기 때문에 개별 단어의 관계성이라는 부분적인 관점에만 초점을 맞추어서 학습을 진행한다[4]. 그러므로 문맥의 흐름이라는 전체적인 관점에서의 언어 모델은 기존 관점과 다른 접근 방법이 필요하다.

문맥의 흐름을 파악하기 위하여 개별 단어들과의 관계보다 문장 사이의 관계에 초점을 맞추어서 문맥의 흐름을 파악하고자 한다. 문서의 문맥의 흐름을 학습하기 첫 번째로, 문서 내 각 문장의 의미를 벡터 공간 상에 표현한다. 텍스트 내 단어를 벡터 공간으로 표현하는 방법 중에 하나로 임베딩 방법이 있다[14].

임베딩은 데이터를 하나의 스칼라 값이 아니라 벡터 공간상의 특정한 영역으로 매핑하는 것이다. 예로, 워드 임베딩은 단어를 가능한 모든 단어 중에 하나로 표현하는 one-hot 벡터와 다르게 단어의 의미를 벡터 공간으로 수치화 하여 표현한 방법이다. 이로 인해서 워드 임베딩은 단순한 단어의 순서가 아니라 유의어, 반대어 등의 의미가 수치적으로 표현할 수 있고, 기타 자연어 처리 응용의 기반이 된다. 워드 임베딩과 유사하게 문장을 벡터로 표현하는 임베딩 기법은 다양하게 사용되고 있다[11, 15]. 벡터 임베딩은 문장의 긍정, 부정뿐 아니라 감정, 의도를 파악하는데 활용되며, 문장과 이미지와의 관계, 퀴즈 답하기 등에서 활용이 되고 있다[9, 16]. 문서의 문맥의 흐름을 파악하기 위해 각 문장을 문장을 표현하는 워드 임베딩으로 표현한다.

문장을 워드 임베딩으로 표현하기 위해서 시퀀스 투 시퀀스 모델을 구현하는 Recurrent Neural Network Encoder-Decoder(RNNED) 모델을 사용한다. RNNED는 Cho 외 6인이 소개한 언어 모델로써 기계 번역 등 여러 응용에 활용되는 모델이다[1]. RNNED는 RNN 2개가 연속적으로 연결된 구조로써 2개의 RNN이 각각 인코더와 디코더 역할을 한다. 인코더는 입력을 처리하고 디코더는 결과를 생성하는 모델로써, 그 예로 기계 번역 시 번역의 대상이 되는 언어가 인코더에 입력이 되고 번역의 결과가 되는 언어가 디코더에서 출력하는 구조이다. 여기서 인코더와 디코더 사이에 전달되는 정보는 인코더의 마지막 상태 백터이며 이 상태 벡터가 디코더의 초기 상태 값으로 사용되어 디코더가 원하는 결과를 출력하도록 언어 모델을 학습시킨다. 수식(6)은 문장이 인코더에 의해서 문장벡터로 변환되는 과정이다.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

*U*와 *W* 는 각각 입력과 출력의 가중치이며, 상태 값은 이전 상태 값과 현재 입력 값에 의해서 결정된다. 그리고 인코더의 마지막 상태 값은 문장 벡터가 되면 수식 (7) 과 같이 표현된다.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

수식(7) 에서 *T* 는 인코더의 마지막 스텝을 의미하며, 문장은 마지막 상태 값으로 인코딩 된다.



그림. 6 RNN Encoder Decoder Model

그림 6은 RNNED 모델의 구조를 도식적으로 표현하고 있다. 하단에 있는 구조가 인코더 역할을 하는 RNN 이며, 각각 입력을 받아들인 후, 마지막 상태 값이 문장 벡터(sentence vector)가 된다. 그림 6 의 상단에 있는 구조는 디코더 역할을 하는 RNN으로써 문장 벡터를 입력으로 받아, 출력으로 본래 입력으로 받았던 단어들을 출력한다. 문장 벡터 *vs* 가 원래 문장으로 디코딩 되는 과정은 수식(8), 수식(9) 과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  | (8) | |
|  |  | | | (9) |

수식(8)은 디코더가 첫 번째 스텝에서 동작하는 과정으로, 초기 상태 값은 인코더에서 생성한 문장 벡터가 사용되고, 입력 값은 초기 시작 토큰이 입력으로 사용된다. 수식(9)는 그 이후에 진행되는 스텝의 과정으로 이전 상태 값과 이전 결과 값이 입력으로 사용된다. 수식(8)과 (9)에서 사용되는 가중치는 인코더에서 사용했던 가중치와 다른 값을 사용한다.

입력 문장은 인코더를 통하여 마지막으로 생성되는 상태 값, 즉 문장 벡터로 생성이 되며, 이 문장 벡터는 다시 디코더를 통해 원래 문장으로 생성될 수 있다. 인코더와 디코더 사이에 전달되는 상태 벡터는 원래 문장을 표현하는 고유한 벡터 값으로 사용될 수 있으므로 문장 벡터라고 간주한다.

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

RNNED 를 통해서 문장 벡터를 얻기 위해서는 수식(10)를 만족해야 한다. 수식(10)에서 *x*와 *y* 는 각각 입력과 출력을 의미하며, 는 RNNED 에 포함되어 있는 파라미터들을 의미한다. 입력 *x*에 대한 출력 *y*의 조건부 확률이 최대가 되도록 수식(10)를 학습한다.

B. 문장 벡터 클러스터링 방법

문장 벡터를 이용하여 각 문장들을 의미 기반으로 클러스터링 하는 기법을 설명한다. 문장 벡터는 RNNED 통하여 생성되었고, 문장의 고유한 값으로 사용할 수 있다고 가정하였다. 문맥의 흐름을 학습하기 위해서는 의미가 비슷한 문장들의 집합을 파악해야 한다. 이를 위해서 문장 벡터를 이용하여 유사한 문장들을 클러스러팅 하는 과정이 필요하다.

클러스터링을 하기 위해서는 문장들간의 유사도의 개념들 정의하는 것이 필요하다. 본 연구에서는 비슷한 문장 벡터를 가지면 유사한 문장이라고 가정하였다. 각 문장들은 워드 임베딩으로 표현되는 단어 벡터들의 연산으로 생성되는 값이고, 단어 벡터들은 미리 word2vec으로 알려진 알고리즘을 통해 이미 유의미한 값을 가진 벡터들이다. 즉 워드 임베딩으로 표현된 단어들은 수치적으로 유의미한 값을 갖고 있으며, 문장 벡터들은 공유된 공유된 가중치 값과 문장 벡터들의 연산을 통해 값을 얻으므로 문장 벡터 또한 유사한 의미를 갖고 있다고 가정하였다.



그림. 7 문서 내 문장들의 클러스터링

그림 7 는 각 문서 안에 있는 문장들이 문장 벡터들이 클러스터링 되는 개념을 표현한 것이다. 문서의 종류에 관계없이 전체 문장 벡터들을 클러스터링 하여 값이 유사한 문장 벡터들을 클러스터링 하면 유사한 의미를 문장들을 묶을 수 있다. 수식(11)와 (12)는 실제 클러스터링이 되었을 때 가질 수 있는 자료 구조를 표현하였다. d는 문서의 종류를 의미하고, *sv*와 *c*는 각각 문장 벡터와 클러스터 번호를 의미한다.

|  |  |  |
| --- | --- | --- |
|  |  | (11) |
|  |  | (12) |

본 논문에서는 각 문서에서 포함하고 있는 문맥의 흐름을 분석하기 위하여 K-means 클러스터링 기법을 사용하여 전체 데이터의 문장 벡터들을 군집화 하였다. K-means 클러스터링 기법은 주어진 데이터를 주어진 K개의 군집으로 구분하는 알고리즘이다 클러스터링을 하기 위해 *K* 값이 주어져야 한다. *n*개의 *d*차원의 데이터 X, 가 주어졌다면, *n*개의 데이터를 각 클러스터에 포함된 데이터들의 응집도를 최대로 하는 *n* 보다 작거나 같은 *K*개의 집합으로 분할한다. 만일 가 집합 의 중심점이라 할 때 식(13)을 만족하는 집합 를 찾는다.

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

즉 식(13)와 같이 각 집합의 중심점과 각 집합에 포함되어 있는 데이터 들 간의 거리의 차의 제곱 합을 최소로 하는 집합 *S*를 찾는 것이 K-means의 목표이다. 이 값을 구하기 위해서 다음 두 단계를 거친다. 첫 번째로 각 데이터가 각 클러스터의 중심점과의 거리를 구해서, 해당 데이터가 가장 가까운 클러스터에 해당 데이터를 추가한다. 두 번째로 각 클러스터들의 중심점을 갱신한다. K-means 클러스터링은 값이 구했을 때 결과가 변함이 없을 때 반복해서 알고리즘을 수행한다.

C. 문맥 흐름 생성 학습 모델

클러스터링으로 얻는 정보를 이용하여 문서에 속하는 문맥의 흐름을 학습하는 기법을 설명한다. K-means 클러스터링 기법으로 문장들을 군집화 한 뒤, 각 문장이 속하는 클러스터를 문장 정보에 추가하면, 문서에 클러스터의 흐름을 파악할 수 있다. 즉 언급한 것과 같이 하나의 문서는 일정한 의미의 흐름을 가지며, 그 의미의 흐름을 표현하기 위해서 같은 혹은 비슷한 의미를 가진 문장들 중에서 의미가 적합한 것으로 개별 문장을 생성한다.

클러스터링 기법으로 문장을 군집화 한 뒤 그 결과를 각 문장에 추가했으므로 개별 문장의 흐름뿐 아니라 문서 전체적으로 의미의 흐름이 흘러가는 패턴을 파악할 수 있다. 수식(11), (12)와 같이 기존 (문서 번호, 문장 벡터) 자료 구조에 클러스터의 정보를 추가하면 문서 내 문맥의 흐름을 알 수 있고, 이를 학습하여 문맥에 따라 문장을 생성할 수 있다.

문서 내의 의미의 흐름을 학습하기 위해서 클러스러링의 결과를 RNN을 통해서 학습한다. RNN은 앞에서 언급한 것과 같이 시계열 데이터의 분석하는데 용이한 모델이며, 문서 안의 의미의 흐름은 순차적으로 발생하는 시계열 데이터의 일종이므로 RNN 모델이 적합한 모델이다.



그림. 8 클러스터링 시퀀스를 구하기 위한 RNN 모델

그림 8은 각 문서 내 문맥의 흐름을 학습하기 위한 RNN 구조를 도식화 한 것이다. 그림 8의 는 문서 내 클러스터 정보를 의미한다. RNN 학습을 위하여 start 토큰을 입력으로 하여 학습을 시작하고 마지막으로 end 토큰을 이용한다. 각 클러스터의 번호를 입력으로 하고, 출력은 입력 클러스터의 다음 입력 값으로 하여 모델을 학습한다. 모델의 내부 상태 값은 수식(14)과 (15)으로 표현이 되며, 결과 값은 수식 (16)을 통하여 결정이 된다.

|  |  |  |
| --- | --- | --- |
|  |  | (14) |
|  |  | (15) |
|  |  | (16) |

수식(14)와 (15)의 *W*와 *U* 는 각각 상태 값과 입력 값에 대한 가중치 이고, *V*은 결과 값에 대한 가중치이다. 수식(16)은 *softmax*를 통해서 문서가 가질 수 있는 클러스터 값들 중에서 가 선택 되도록 학습을 시킨다.

**3-3. 문장 생성 모델**

문서 모델이 완성되었으면, 두 번째로 문서 모델을 이용하여 문장 생성에 관련된 모델을 학습 및 생성하는 문장 생성 모델을 제안한다. 문장 생성 모델은 주어진 문맥과 이전 문장을 사전 조건으로 하여 문장을 생성하는 언어 모델을 의미한다.

앞에서 언급한 것과 같이 문서나 글의 문장은 글의 문맥의 흐름 가운데서 선택된 후 생성되는 것이다. 문장이 무의미하게 나열되는 것이 아니라 글은 일련의 문맥이 흐름이 있고, 그 흐름에 적합한 문장이 선택되어 생성되는 것이다. 문장을 생성하기 위해서는 일단 문맥의 흐름을 선택한 뒤, 바로 전에 생성된 문장과의 의미 관계를 고려하여 문장을 생성한다. 이를 위해서 문맥과 이전 문장을 조건으로 문장 생성 과정을 학습한 뒤, 이 정보를 이용하여 새로운 문장을 생성할 수 있는 문장 생성 모델을 제안한다.

문서 생성 모델은 LSTM기반 언어 학습 모델이 사용되며, 입력 정보로 클러스터 정보와 이전 문장 정보 그리고 문장 생성 모델에서 입력으로 사용되는 워드 임베딩 정보가 사용된다. 일반적인 언어 생성 모델이 입력과 출력으로 각각 워드 임베딩 혹은 단어의 시퀀스 정보가 입력되는 데 반하여 본 논문에서 제시하는 모델은 문맥의 정보를 고려한 문장 생성 모델이 되어야 하기 때문에 문맥 정보, 즉 클러스터 정보와 이전 문장의 워드 임베딩 정보가 추가로 입력이 된다.



그림. 9 문맥을 고려한 문장 생성 RNN 구조

입력과 출력 데이터를 정리하면 다음과 같다. 현재 단어와 출력 단어는 워드 임베딩을 통해 사상된 벡터 값을 가진다. 문서의 각 단어는 워드 임베딩을 통해 *d* 차원의 float형 값을 갖는다. 그리고 이전 문장 정보는 시퀀스 투 시퀀스 모델을 통해서 생성된 문장 벡터 값이다. 이 역시 RNN모델이 *d*차원의 상태 값을 가지므로 마지막 상태 값인 문장 벡터도 *d*차원의 값을 가진다. 문맥 정보는 모든 문장 벡터 값들을 클러스터링 하여 생성된 각 클러스터의 중심값을 의미한다. 클러스터 값 역시 상태 값들의 중간 값 이므로 *d* 차원의 값을 가진다. 그림 9는 문장 생성 모델의 구조를 도식화 하였다. *p* 와 *c* 는 각각 이전 문장 정보 벡터와 클러스터 중심 값 벡터를 의미한다. *p* 와 *c* 가 초기 RNN 의 상태 값으로 들어가고 각 워드 임베딩 벡터들은 각 스텝마다 입력으로 들어간다.

|  |  |  |
| --- | --- | --- |
|  |  | (17) |
|  |  | (18) |

수식(17)과 (18)에서 *W* 와 *U* 는 각각 상태 값과 입력 값의 가중치를 나타내며, *p*와 *c*는 서로 2 x *d* 차원이 되도록 접합되어서 초기 상태 값으로 들어간다. 초기 상태 이후 상태 값은 이전 상태 값이 되고 입력 값은 이전 결과 값이 된다.

**제 4장 실 험**

**4-1. 실험 데이터 및 평가 척도**

A. 실험 데이터 분석

본 실험의 데이터는 Document Understaning Conference(DUC)의 DUC 2002 데이터 셋을 사용한다. DUC 2002 데이터 셋은 여러 개의 주제에 대한 여러 다중 문서를 포함하고 있으며 각 문서는 다수의 문장을 포함하고 있다. 약 60여개의 주제를 포함하고 있고, 각 주제별로 약 5~10개의 문서를 포함하고 있으며 각 주제에 대한 요약문을 포함하고 있다. 각 데이터 셋의 세부적인 사항은 아래 표와 같다.

Table 1 학습을 위한 자료 설명

|  |  |
| --- | --- |
| 주제 수 | 59 |
| 전제 문서 수 | 566 |
| 주제 별 평균 문서 수 | 9.6 |
| 전체 문장 수 | 15691 |
| 문서 별 평순 문장 수 | 27.2 |

B. 평가 척도

본 연구에서는 기존의 문장을 학습한 뒤, 문맥에 맞는 문장을 생성하는 모델을 제안한다. 본 실험에서는 모델을 통해 생성된 문장이, 문맥에 맞게 생성되었는지 평가하기 위해서 모델을 통해 주제 별로 생성된 문장과, 데이터 셋에 존재하는 각 주제 별 요약문의 유사도를 비교하였다. 문서들 사이의 유사도가 실험 결과를 통해 나오는 문장의 문맥의 정확성을 판단하기는 어려울 수 있지만, 다른 문서들과 유사도를 비교함으로써 비교적 문맥에 맞는 적합한 문장을 생성했는지 판단의 근거가 될 수 있다.

**유사도(Similarity)** : 두 문서간의 유사도를 측정하기 위하여 문서안의 단어들의 TF-IDF(Term Frequency – Inverse Document Frequency) 가중치를 측정한 뒤 이를 문서 벡터로 사용하여 두 문장 벡터간 cosine similarity 을 측정한다.

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

TF-IDF 는 한 문서 안에서 단어 출현 횟수와 단어가 출현 된 문서의 수의 역수를 곱한 값으로 (수식 19) 각 단어가 문서들 사이에서 얼마나 중요하게 사용되는지 나타내는 수치이다. 이 수치를 벡터화 하여 문장 및 문서의 특성을 나타내고, 또 이 값을 이용하여 문서 간 유사도를 측정한다.

**4-2. 실험 방법 및 결과**

A. 실험 방법

실험 방법을 위해서 본 논문에서 제시한 3개의 RNN 모델이 필요하다. 첫번째 모델은 RNN Encoder-Decoder 모델로 문장 벡터를 학습할 때 사용되며, 두번째 모델은 클러스터된 결과의 시퀀스를 학습하는데 사용된다. 마지막 세번째 모델은 클러스터된 결과와 이전 문장 벡터를 이용하여 문장 생성을 학습하는 모델이다.

첫번째 RNN Encoder-Decoder 모델을 통해서 15691개의 문장의 문장 벡터를 도출하였다. 128차원의 워드 임베딩을 사용하였고 *c*상태와 *h*상태가 있는 RNN 셀을 사용하였기 때문에 총 128 x 2 차원의 문장 벡터가 생성되었다. 문장 벡터로 만들기 위해 *c*와 *h* 상태를 접합하여 216 차원의 문장 벡터로 생성하였다.

두번째 문장 벡터의 클러스터를 위하여 K-mean 클러스터링을 하였고 여러 번의 실험 결과 5000개의 클러스터로 군집화를 하였다. 군집화 결과 각 문장벡터는 클러스터 번호를 갖게 되었고, 각 문서를 기준으로 클러스터 번호의 시퀀스를 학습하였다. 클러스터 번호의 시퀀스를 학습시 RNN 을 사용하였고, 클러스터의 중심 값을 입력 값으로 사용하였다.

세 번째 모델은 각 자기 클러스터와 이전 문장 벡터를 초기값으로 입력 받아 문장 생성을 재 학습 하는 모델이다. 자기 클러스터는 각 클러스터의 중심 값을 사용하였고, 이전 문장 벡터는 이전 문장의 첫 번째 RNN 의 인코더 부분을 사용하여 문장 벡터를 사용하였다. 모델의 초기 상태는 클러스터 중심 값 벡터 216 차원과 이전 문장벡터 216 차원을 사용하였고, 입력 값으로 워드 128 차원을 사용하였다.

B. 실험 결과

Seed 문장을 입력으로 받아, Seed 문장의 문맥을 잇는 문장을 생성하였다. Seed 문장은 미 학습 데이터로 DUC 2002에서 제공하는 요약문 중 하나이다. Seed 문장이 주어지면, Seed문장의 문장 벡터를 생성하고, 이미 학습된 클러스터링 된 결과를 이용하여, Seed 문장의 클러스터 및 클러스터 시퀀스를 계산한다. 그 뒤 클러스터 시퀀스에 맞는 문장을 계산하여 문맥에 맞도록 문장을 생성한다. 표 2에서 확인 시 비교적 정확한 문장은 아니지만 비교적 문맥에 맞는 문장을 생성하였고, 비교적 Seed 문장의 문맥을 잘 이어가는 문장을 생성하였다.

Table 2 Seed 문장을 통해 생성된 문장 예제

|  |
| --- |
| Seed 문장 |
| How the Hubble Space Telescope Finally Got Into Orbit. |
| 생성된 문장. |
| Navy company said the investigation was reported to be the identified .  The 12-ton telescope , a story taken on the issue , the media won the main army group , died in the federal system was announced by the state was criticized to pakistan gilbert . Not never meant in a movie miss called , UNK and the academy 's 13 leader group . Proceeds are more than six hours from the booker prize , " the base , several , and the navy explosion , in the past decade refused , was possible when it was not a full contact by current running to report the governor . The magazine is just after heart attacks , researchers said the drugs , though a UNK rating in system was 1988 's report . Ball has made UNK , then a reporter times by the newspaper of bihar . The two countries mostly weeks agreed to the same war to it to the unit to gilbert rates to it . This year the prize has offered be a UNK literary carried into the dogs as the m building for sponsors by the danger million building group on the sources on the security coast . The daffy , redheaded comedian whose harebrained schemes drove her oscar ceremony for land for robert maxwell 's dogs was given a crew of UNK a chain on a language show a call a month 's district on the campaign , tass . |

Table 2 에서 Seed 문장을 통해 생성된 문장을 확인할 수 있다. 중간에 가서는 Seed 문장과 약간 주제가 다른 문장이 나오지만, 비교적 문법에 맞고 흐름을 따라가는 문장이 생성된 것을 확인할 수 있다.

Table 3 주제별 생성된 문장과 요약문 사이의 유사도 비교

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 요약문 A | 요약문 B | 요약문 C | 요약문 D | 요약문 E |
| 생성 문 A | 0.36376 | 0.35227 | 0.13718 | 0.34936 | 0.21804 |
| 생성 문 B | 0.31807 | 0.37378 | 0.09575 | 0.34571 | 0.23522 |
| 생성 문 C | 0.32865 | 0.36656 | 0.14820 | 0.35060 | 0.18319 |
| 생성 문 D | 0.32348 | 0.35353 | 0.11500 | 0.37075 | 0.22827 |
| 생성 문 E | 0.30511 | 0.33138 | 0.10430 | 0.34816 | 0.19262 |

Table3 에서 데이터 셋에 있는 주제별 요약문과 주제별 생성된 문장간의 유사도를 확인할 수 있다. 생성문과 요약문 A, B, C, D, E 는 데이터 셋에 있는 60개의 주제 중 임의로 선택된 5개 주제들이다. 생성된 문장 A, B, D 는 기존 요약문과 비교했을 때 비교적 기존 요약문과 유사한 문장을 생성하였다.

**제 5장 결 론**

딥러닝 기반의 자연어 처리 연구는 기계 번역, 문서 감성 분류, 문서 요약 등 많은 분야에서 발전된 모습을 보여주고 다양한 문제를 해결하는데 응용이 되고 있다. 뿐만 아니라 기존의 데이터를 분석하는 차원을 넘어서 주어진 데이터를 학습하고 이를 응용하여 새로운 문장 데이터를 생성하는 연구 또한 발전하고 있다.

하지만 기존의 단어의 패턴을 학습하는 일반적인 접근 방법으로는 주제에 맞는 보편적인 글쓰기 방법에 접근하기 어려운 점이 있다. 단편적이고 부분적인 단어 사이의 관계 파악으로는 문맥에 맞는 글쓰기가 어려운 단점이 있다.

이러한 문제를 해결하고 문맥에 맞는 문장을 생성할 수 있는 언어 학습 모델을 제안했다. 문맥이라는 개념을 정의하였고, 글 전체에 존재하는 문맥에 맞추어 문장을 선택, 생성하는 개념을 사용하여 문맥에 맞는 글쓰기 방법을 제안하였다. 이를 위해서 첫째로 문서에 있는 문맥을 파악하기 위해 각 문장의 특성을 나타낼 수 있는 문장 벡터를 생성하였고, 문장 벡터를 클러스터링하여 각 문장들을 의미가 비슷한 것끼리 클러스터링 하였다. 문맥의 흐름을 학습하기 위하여 문서 내 문장들은 클러스터 정보로 치환되어 각 문서별 문맥의 흐름으로 나타내었다. 마지막으로 각 문서별 문맥의 흐름에 맞는 문장을 생성하기 위하여 클러스터 정보와 이전 문장 정보를 이용하여 문장을 생성하도록 언어 모델을 학습시키는 방법은 제시하였다.

논문에서 제안한 문맥에 따른 문장 생성 방법은 다양한 방법으로 확장 및 발전이 가능하다. 첫째 문장을 표현할 때 논문에서 제시한 RNNED(Recurrent Neural Network Encoder Decoder) 이외에 다른 방법으로 사용이 가능하다. 문장의 감성 및 의미를 좀 더 명확하게 나타낼 수 있는 방법 사용이 가능하다. 또한 각 문장간의 유사도를 측정할 때 논문에서 제시한 Euclidean distance 이외에 다른 방법이 있다. 문장간 유사도를 측정하는 것은 도전적인 주제로 많은 연구가 이루어지고 있으므로 다른 방법 사용 또한 가능하다.

향후 연구에서는 문장의 의미와 특성을 좀 더 명확하게 나타내어 문맥의 의미를 좀 더 명확히 할 수 있는 표현 방법에 대한 연구와 문장뿐 아니라 문단과 같이 더 큰 문맥의 의미를 고려하여 글쓰기에 활용할 수 있는 방법을 연구할 것이다.

**참고 문헌**

[1] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[2] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation." arXiv preprint arXiv:1309.4168 (2013).

[3] Tang, Duyu, et al. "Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification." ACL (1). 2014.

[4] Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating text with recurrent neural networks." Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011.

[5] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. 2010.

[6] Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. "LSTM Neural Networks for Language Modeling." Interspeech. 2012.

[7] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

[8] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." arXiv preprint arXiv:1412.6632 (2014).

[9] Serban, Iulian V., et al. "Building end-to-end dialogue systems using generative hierarchical neural network models." Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[10] Choi, YunSeok, SuAh Kim, and Jee-Hyong Lee. "Recurrent Neural Network for Storytelling." Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems, 2016 Joint 8th International Conference on. IEEE, 2016.

[11] Palangi, Hamid, et al. "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval." IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 24.4 (2016): 694-707.

[12] 김다해, and 이지형. "문서 주제에 따른 문장 생성을 위한 LSTM 기반 언어 학습 모델." 한국컴퓨터정보학회 학술발표논문집 24.2 (2016): 17-20.

[13] 국립국어원, 표준국어대사전

[14] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.

[15] Socher, Richard, et al. "Grounded compositional semantics for finding and describing images with sentences." Transactions of the Association for Computational Linguistics 2 (2014): 207-218.

[16] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[17] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

**ABSTRACT**

**Language learning model for context-based sentence generation**

Park Kyounghan

Department of Digital Media

Sungkyunkwan University

This thesis provides language learning model for generating sentence based on context and subject. Along with advanced research about Deep Learning, there is also a lot of progress in research about natural language processing. Natural language processing based on Deep Learning show a state-of-art performance comparing with rule-based approach or method based on probability in before. As a result, they drive research like machine translation, summarization of document or classification based on sentiment and also is used to solve various problems. Not only that, they are being developed to generate new article or sentence based on learning given data.

But these researches only depend on simple rules which exist on between words and cannot generate which have context or subject in them. To overcome this problem, this thesis proposes a language learning model which can generate sentence based on context by analyzing sentences in data.

For that, this thesis proposes three steps method. Firstly, this suggests the method of generating sentence vectors which represent the given sentence to know the context of it. Secondly, this suggests the method of knowing the rules rand relations between the sentences by clustering sentence vectors. Through the second method, we can see the context which is in the given text by re-organizing the sentence with cluster data. Lastly this suggests the method of language learning model to generate sentences based on context using context information and the previous sentence.

Finally, evaluation is done to confirm the suggested method in thesis. Through evaluation, new sentences based on the context being generated are confirmed. The results show that the method which is proposed in this thesis can be used to generate sentences for context.

**Keywords: Language learning model, Text generation, Sentence context, RNN encoder decoder, RNN**

(표지측면)

|  |
| --- |
| **碩**  **士**  **學**  **位**  **請**  **求**  **論**  **文** |
|  |
| **문**  **맥**  **을**  **따**  **른**  **문**  **장**  **생**  **성**  **을**  **위**  **한**  **언**  **어**  **학**  **습**  **모**  **델** |
|  |
| **2**  **0**  **1**  **7** |
|  |
| **朴**  **景**  **韓** |