



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석

연세대학교 공학대학원
컴퓨터공학 전공
조 룬

Attention 기반 LSTM 신경망을

이용한 한국어 화행 분석

지도교수 조 성 배

이 논문을 석사학위 논문으로 제출함




2017년 12월 일

연세대학교 공학대학원

컴퓨터공학 전공

조 룬

조 루의 석사학위논문을 인준함

심사위원 조 성 배 인 
심사위원 박 상 현 인 
심사위원 이 인 권 인 

연세대학교 공학대학원

2017년 12월 일

감사의 글

제 삶의 모든 순간에 함께 하시고 성실과 감사함으로 살아갈 수 있도록 바른길을 깨닫게 하시는 하나님 아버지께 감사드립니다.

빠른 속도로 발전하는 기술 흐름에 제 중심을 바로잡기 위해 대학원에 지원 하였고, 학업 기간 동안 발전된 컴퓨터공학의 지식과 업계의 현황 및 문제점 그리고 이를 극복하고 더 나아가기 위한 지혜와 아이디어들을 얻었습니다. 이제 졸업을 앞두고 되돌아보니 감사드려야 할 분들이 너무도 많습니다.

먼저 인공지능에 대해 깊이 있게 생각할 수 있는 눈을 뜨게 해주시고, 논문을 지도해 주시는 동안 엄격하면서도 따뜻한 격려로 부족한 부분을 알려주시고 논문이 공헌 점을 가질 수 있도록 지도해주신 조성배 교수님께 감사드리며, 더 나은 논문으로 나아갈 수 있도록 심사위원을 맡아주시고 지도해주신 박상현 교수님과 이인권 교수님, 초기 지도를 맡아주신 이경호 교수님, 수학적인 조언을 주신 윤진희 교수님께 감사드립니다. 그리고 학업을 시작할 수 있도록 격려해주신 류재철 이사님과 정병근 형님, 논문작성과 학업 기간 동안 따뜻한 조언과 다양한 도움을 주신 혁형과 재운형 그리고 대학원 동기분들께 감사드리며 또한, 학업과 일을 병행할

수 있도록 배려와 도움을 주신 양유진 대표님과 한우진 본부장님, 이준정 실장님, 장용희님, 강정은님, 이동성님, 김선국님, 전동준님과 동료분들 그리고 특허등록을 제안해주신 최성주님, 송영록님께 감사드립니다.

무엇보다 저를 아끼시고, 저를 위해 묵묵히 기도해주시는 부모님이 계셨기에 여기까지 올 수 있었으며, 마음의 버팀목이 되는 누나와 매형들과 보고픈 조카들, 언제나 저를 믿고 응원해주시는 장모님과 경근형, 따뜻한 마음을 가진 처가 식구들에게 감사드립니다. 그리고 항상 저를 위해 기도해주신 할렐루야 교회 영아부와 세레팀, 3 교구 가족분들과 믿음의 멘토인 성현누나와 명안형, 명우, 현진이에게 감사함을 전합니다.

하나님이 주신 가장 큰 은혜인 아내에게 감사의 말을 전합니다. 힘든 상황에서도 언제나 나를 믿어 주었으며, 학업과 일을 병행하여 가정과 승범이와 승우에게 늘 부족한 아빠였지만 언제나 빈틈없이, 더욱 튼튼한 믿음의 가정으로 이끌어줘서 감사하고 언제나 나를 위한 배려만 해줘서 늘 미안하고 고맙게 생각하고 있습니다. 우리 승범이와 승우, 그리고 무엇보다 당신의 행복과 건강을 위해 그리고 믿음의 가정이 되도록 기도하고 노력하며 살아갈 것입니다.

이외에도 도움을 주신 한분 한분께 깊이 감사드립니다.

2017 년 12 월 14 일 조륜

목차

목차	i
그림 목차	iii
표 목차	iv
약기호표	v
국문요약	vi
 제 1 장 서론	 1
1.1. 연구의 배경	1
1.2. 연구의 내용	4
1.3. 논문의 구성	6
 제 2 장 관련 연구	 7
2.1. 기계학습 기반 화행 분류	7
2.2. 순환 신경망 기반의 자연어처리에 관한 연구	9
2.3. Attention 기반 인공신경망을 이용한 연구	15
 제 3 장 Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석 방법	 18
3.1. 순환 신경망을 이용한 한국어 문장의 벡터 표현 방법	18
3.2. Attention 기반 LSTM 신경망을 적용한 방법	21
 제4장 실험 및 결과	 25
4.1. 실험 데이터 수집	25

4.2. 실험 설계	27
4.3. 실험 결과 및 분석	29
 제5장 결론 및 향후 연구	 36
5.1. 연구의 한계점 및 향후 연구	36
 참고문헌	 38
 ABSTRACT	 42

그림 목차

<그림 1> 질의응답 시스템 구성도	2
<그림 2> 순환 신경망 다이어그램	9
<그림 3> 전개된 순환 신경망 다이어그램	10
<그림 4> Long Short Term Memory 다이어그램	11
<그림 5> 양방향 순환 신경망 다이어그램	13
<그림 6> 시퀀스 투 시퀀스의 인코더 디코더(출처: [18])	15
<그림 7> Neural Machine Translation Attention Mechanism(출처: [19])	16
<그림 8> Show Attend and Tell Memory(출처: [20])	17
<그림 9> LSTM Context Vector	20
<그림 10> attention 벡터를 context 벡터로 사용한 방법	22
<그림 11> attention 벡터와 마지막 시퀀스를 연결한 방법	23
<그림 12> attention 벡터와 마지막 시퀀스 요소 간 곱을 사용한 방법	24
<그림 13> 학습 시스템 구성도	28
<그림 14> 화행의 혼동 행렬	32
<그림 15> Attention을 이용한 화행 별 형태소 가중치	35

표 목차

<표 1> 화행의 종류와 발화 문장 예	3
<표 2> 데이터 셋 구성	26
<표 3> 실험 환경	29
<표 4> 모델간 성능 비교	30
<표 5> attention context vector 비교	30
<표 6> 모델간 대응표본 T-검정 유의확률	31
<표 7> 활성 함수와 Optimizer 비교	31
<표 8> 화행의 혼동 행렬	32
<표 9> Precision, Recall, f1-score	33
<표 10> 발화 문장 결과표	34

약기호표

RNN: Recurrent Neural Network

GRU: Gated recurrent unit

LSTM: Long Short Term Memory

CNN: Convolutional Neural Network

SVM: Support Vector Machine

CRF: Conditional random field

HMM: Hidden Markov Model

NMT: Neural Machine Translation

국 문 요약

Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석

연세대학교 공학대학원

컴퓨터공학 전공

조 루

화행(speech act)은 발화를 통해 전달되는 화자의 의도를 가리킨다. 대화형 질의응답 시스템에서는 사용자의 의도에 맞는 응답을 제공하기 위해 사용자의 질의를 분석하는 것이 중요하다. 화행 분석을 위한 기존 연구들은 발화의 정보를 분석하기 위해 과도한 자질 엔지니어링을 하였으나 최근 자연어 처리 분야 연구에서 우수한 성능을 보이는 인공신경망을 이용한 연구가 활발히 진행되고 있다.

그러나 인공신경망을 이용한 자연어처리 연구 대부분이 영어 및 라틴어 기반 언어로 진행되어 한국어에 관한 연구사례는 비교적 적으며

인공신경망 기계번역에서 우수한 성능을 보인 Attention 기반 한국어 화행 분석에 관한 선행 연구사례가 없었다.

이에 본 논문에서는 한국어 화행 분석을 위해 SNS 에 작성된 게임 분야 한국어 문장 607,000 개를 수집하였고 요청 화행, 항의 화행, 질문 화행, 진술 화행 4 가지에 대해 총 48,000 개의 학습데이터를 만들어 Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석 모델을 제안하였다.

실험을 통하여 한국어에 적합한 임베딩 파라미터와 Attention 조합 방법에 따른 성능을 비교 평가하였으며 학습된 형태소의 가중치를 추출하여 각 화행의 자질이 되는 형태소에 더 높은 가중치가 학습되는 것을 확인하였다.

핵심어: 화행, 의도, 인공신경망, 순환신경망, 딥러닝, LSTM, Attention

제 1 장 서론

본 장에서는 연구의 배경 및 연구의 목적이 되는 화행에 대한 정의와 화행 분석에 관하여 설명하고, 연구의 범위 및 논문 구성에 관해 기술한다.

1.1. 연구의 배경

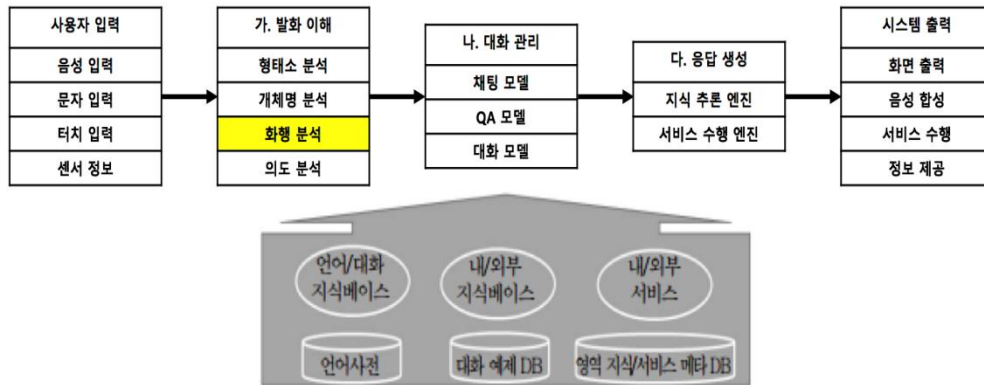
최근 지능형 개인비서, 챗봇과 같은 자연언어처리 기반의 연구가 활발히 이루어지면서 질의응답이 가능한 다양한 시스템이 등장하고 있다. 국내에서는 IBM 왓슨(Watson)¹을 한국어 환경에 최적화한 인공지능 서비스 에이브릴(Aibrill)²이 오픈하여 키워드 기반 검색을 넘어 사용자가 자연어로 질문하면 응답을 찾아주는 질의응답 시스템 (Question Answering System, QA System)으로 다양한 분야에서 연구 및 활용이 되고 있다.

대화형 질의응답 시스템에서는 사용자가 입력한 발화로부터 화행 분석을 통해 화자의 의도를 파악하고 해당 도메인에 필요한 정보를 추출한 뒤, 시스템상에서 정의된 전략에 따라 응답을 생성한다[1].

1 <https://www.ibm.com/watson/>

2 <https://www.aibril.com/web/main/getMain.do>

<그림 1>은 질의응답 시스템의 프로세스 동작 방식을 보여주고 있으며 초기 단계에 화행 분석이 수행되는 것을 알 수 있다.



<그림 1> 질의응답 시스템 구성도³

화행(speech act)이란 발화 속에 포함되어 있는 의도 된 언어적 행위로, 자연어를 처리하는 많은 시스템에서 화자의 의도 파악을 위해 중요한 역할을 한다[2]. 또한, 화행 분석은 시스템에서 응답을 위해 처리하는 절차 중 초기 단계에 해당하기 때문에 중요하고 높은 성능을 보여야 한다[3].

<표 1>은 게임 분야 SNS 게시글에서 수집한 문장으로 간접 화행의 예시를 보여주고 있다. 간접화행이란, 문장의 유형이 가진 본질적인 화행과 다른 화행을 나타내는 것으로 진술 문으로 질문이나 항의 화행을,

3 nipa 정보통신산업진흥원(http://itfind.or.kr/itfind/getFile.htm?identifier=02-001-160412-000003&classCode=B_ITA_01_42)

질문문으로 향의 화행이나 요청 화행을 나타낸다.

<표 1> 화행의 종류와 발화 문장 예

화행 종류	발화 문장
질문 화행	어떤 메뉴를 선택해야 하나요. 속도를 올리고 싶어요.
향의 화행	구매 했는데 안들어 옵니다. 한 두번도 아니고 매번 뭘니까?
요청 화행	접속이 안될때가 있어요. 왜 안되죠? 해결 부탁 드려요.
진술 화행	이전 버전과 많은 차이점은 모르겠어요. 하지만 속도는 더욱 빨라져서 좋아요.

<표 1>의 문장과 같이 화자는 간접 화행으로 자신의 의도를 다양하게 표현할 수 있으므로 어휘, 품사 자질 정보만으로는 화행을 완벽하게 분석하는 것이 어렵다[4].

이처럼 화행 분석은 자연어를 통하여 의도를 파악하고 응답을 찾아주는 지능형 시스템에서 기초적이고 중요한 역할을 담당하고 있으며 어휘 및 품사 정보만으로는 분석하는 것이 어렵기에 다양한 기계학습⁴ 방법으로 연구되고 있다[1][5].

4 https://en.wikipedia.org/wiki/Machine_learning

1.2. 연구의 내용

최근 다양한 분야에서 활용되고 있는 딥 러닝이 기존 기계학습 방법론들을 뛰어넘고 있으며 여러 자연어처리 분야에서도 우수한 성능을 보이고 있다[6].

그러나 딥 러닝을 이용한 자연어처리 연구 대부분은 영어를 비롯한 라틴어 기반 언어만을 대상으로 진행되어 한국어에 관한 연구사례는 비교적 적으며[7], 인공신경망 기반 기계번역⁵에서 우수한 성능을 보인 Attention 기반 한국어 화행 분석을 실험한 선행 연구사례가 없었다. 이에 본 논문에서는 Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석 모델을 제안한다.

화행 분석을 위한 데이터는 게임 분야 SNS 게시글을 기본 말뭉치로 사용하였다. 화행 분류 체계는 각 분야의 데이터마다 다양하게 수립될 수 있기에 본 논문에서는 (Austin et al., 1962)의 언어 행위론에 근거해 기본 화행으로 제시한 명령, 요구, 축원, 겸허, 질문, 기원, 요청, 감탄을 기본 체계로 하여[8], 수집된 데이터에서 발생 빈도가 높은 요청, 질문,

5 https://en.wikipedia.org/wiki/Neural_machine_translation

항의 그리고 나머지 화행은 진술로 하여 4 가지 화행에 대해 12,000 개씩
총 48,000 개의 학습데이터를 생성하였다.

본 연구에서 이용한 인공신경망 모델로는 자연어처리 분야에서 많이
사용되는 순환 신경망과 LSTM 신경망 그리고 Attention 기반 LSTM
신경망을 이용했으며 한국어에 적합한 파라미터 및 성능을 실험하였다.

1.3. 논문의 구성

본 논문의 구성은 다음과 같다. 다음 2장에서는 화행 분석과 관련된 기존 연구와 인공지능망을 이용한 자연언어처리 및 Attention 기반 연구에 대해 살펴본다. 3장에서는 한국어 화행 분석을 위한 Attention 기반 LSTM 신경망 구조를 중점적으로 살펴보고 분석한다. 그리고 4장에서는 본 논문에서 제안하는 Attention 기반 LSTM 신경망의 결합 방법 따른 성능을 비교 평가한다. 마지막으로 5장에서는 연구의 결론 및 향후 과제에 대해 정리한다.

제 2 장 관련 연구

본 장에서는 화행 분석을 위한 기존 기계학습 방법과 최근 활발히 연구가 진행되고 있는 인공신경망을 이용한 자연언어처리 연구에 대해 살펴본다.

2.1. 기계학습 기반 화행 분류

화행 분석을 수행하는 기존 연구에서는 대화 말뭉치에서 직접 얻을 수 있는 어휘, 품사 자질 등을 이용하여 기계학습 모델을 연구하였다. (김세종 et al., 2008)의 연구에서는 분류할 발화 이전의 발화에서 얻을 수 있는 정보를 이용하여 SVM⁶ 과 HMM⁷ 기반의 분류 모델을 제안하였으며[9], (이현정 et al., 2008)의 연구에서는 형태소 단위의 품사 및 어휘 정보를 특정 도메인에서 자질로 이용하여 CRF⁸ 기반의 분류 모델을 제안하였으며[5], (Silva et al., 2011)의 연구에서는 화행 분류를

6 https://en.wikipedia.org/wiki/Support_machine

7 https://en.wikipedia.org/wiki/Hidden_Markov_model

8 https://en.wikipedia.org/wiki/Conditional_random_field

위해 SVM과 Naïve Bayes⁹를 결합하는 방법을 제안하였다[10].

화행 분류에 사용되는 말뭉치는 화행 분류 체계를 세분화할 경우 말뭉치 크기보다 화행 종류가 많아져서 데이터 희소 문제가 발생할 수 있는데, (김동준 et al., 2015)의 연구에서는 화행 분류에서의 데이터 희소 문제를 완화하기 위해 단어 임베딩을 사용하는 방법을 실험하였다[11]. 대량의 원시 말뭉치로 임베딩 모델을 구축하고 문장 내에서 선택이 안 된 단어가 있을 때 선택된 단어들 간의 임베딩 공간의 유사도를 계산하여 해당 단어로 사용하고 선택되지 않은 단어도 SVM으로 학습시켜 적은 말뭉치로도 효율적인 학습 방법을 제안하였다.

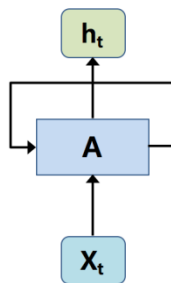
제안된 기계학습 방법론들은 기존 방식들보다 성능 및 효율의 향상이 있었지만, 과도한 자질 엔지니어링이 필요하다는 단점이 존재하였다.

9 https://en.wikipedia.org/wiki/Naive_Bayes_classifier

2.2. 순환 신경망 기반의 자연어처리에 관한 연구

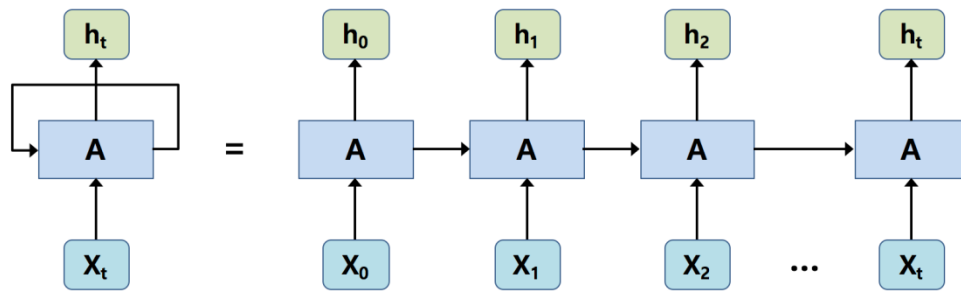
최근 자연어 처리 분야에서는 인공지능망 기반 연구들이 다양하게 진행되고 있으며 우수한 결과를 보이고 있다. 인공지능망 중 은닉층이 여러 층으로 쌓인 구조를 딥 러닝(Deep Learning)이라 부른다. 딥 러닝을 이용하면 입력 형태소 및 문장을 추상화시켜 벡터로 임베딩할 수 있다 [12].

딥 러닝의 여러 가지 구조 중 순환 신경망(Recurrent Neural Network, 이하 RNN)[13] 구조는 그림과 같이 내부에 루프가 있으며, 루프는 네트워크의 정보가 지속하는 것을 도와주는 역할을 한다.



<그림 2> 순환 신경망 다이어그램

<그림 2>에서 RNN 셀은 입력 x_t 를 받아 결과값 h_t 를 출력하며 신경망의 정보는 다음 단계로 전달되도록 한다. 네트워크의 정보를 다음 네트워크로 전달하는 특성을 가진 RNN은 순차적인 데이터를 처리하는데 특화된 구조이며 펼치면 <그림 3>과 같은 네트워크 구조이다.

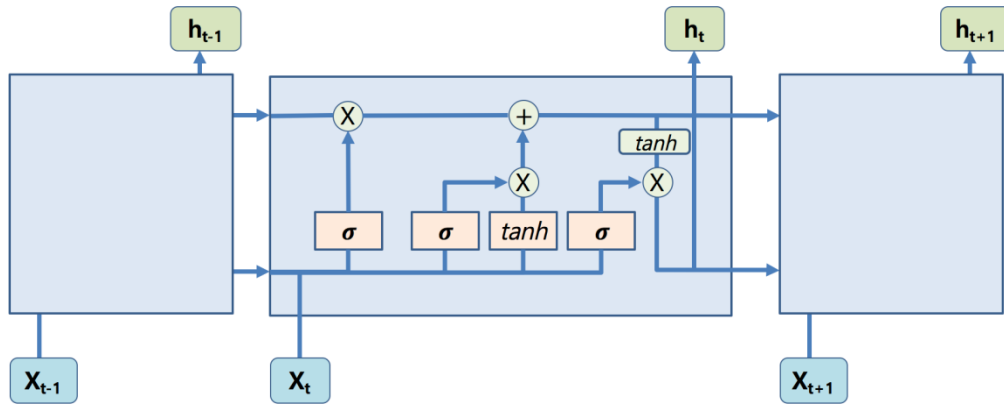


<그림 3> 전개된 순환 신경망 다이어그램

RNN은 다른 인공신경과 달리 전체 시스템이 같은 인자를 공유하며 데이터의 길이에 상관없이 같은 연산을 각 데이터에 반영하고, 이전 데이터의 연산 결과가 현재 데이터에 반영되어 연속적인 입력 데이터를 같은 크기의 벡터로 표현할 수 있다.

RNN의 이러한 특징을 이용하여 자연어 처리 분야에서는 문장, 문단 및 문서 전체를 하나의 벡터로 표현하려는 연구가 활발하게 진행되고 있다. 문장 및 문서의 의미를 벡터로 표현하게 되면 감성 분석, 기계 번역, 정보 검색, 그리고 화행 분석 등 자연어 처리의 다양한 분야에서 유용하게 활용할 수 있다.

지금까지 살펴본 RNN은 연속적인 입력을 처리하는데 특화된 구조이지만, 입력 데이터가 너무 길어지면 초기 데이터의 정보가 네트워크의 마지막까지 전달 안 되는 그래디언트 사라짐 문제가 자주 발생하게 된다. 이를 보완하기 위해 <그림 4>의 구조를 가진 LSTM 신경망이 제안되었다[14].



<그림 4> Long Short Term Memory 다이어그램

LSTM 모델은 입력 데이터가 들어왔을 때 이전 네트워크의 상태를 얼마나 적용할 것인지 함께 학습하는 구조이다. 은닉 상태 외에도 과거의 입력 정보를 저장하는 메모리인 셀 상태와 입력 및 은닉 상태를 다음 상태와 셀에 얼마나 적용할지 조절하는 게이트들을 추가하여 긴 입력으로 구성된 데이터도 초기의 입력 정보가 오래 보존될 수 있다. 이를 수식으로 표현하면 다음과 같다.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

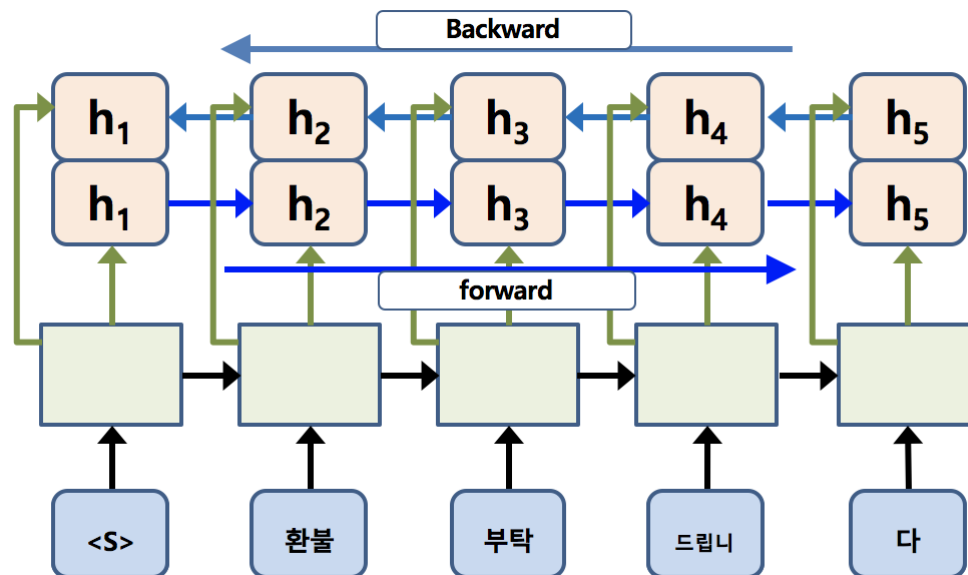
$$h_t = o_t \odot \tanh(C_t)$$

LSTM의 구조에서 가장 중요한 부분은 셀 상태 C_t 이다. 이 셀 상태는 선형 상호작용만 일으키며 셀 상태에 정보를 추가하거나 삭제할 수 있다. 이러한 과정을 게이트 구조들을 조절하며 필요한 정보를 선택적으로 통과하게 한다. 이러한 게이트는 시그모이드와 요소 간 곱셈으로 계산된다. 시그모이드는 0~1 사이의 실수를 출력하는데 이 값이 필요한 정보를 얼마나 통과시키는지 나타낸다.

LSTM은 셀 상태를 조절 하기 위해 세 종류의 게이트를 갖는다. LSTM의 첫 단계인 잊음 게이트는 셀 상태에서 어떤 정보를 잊을지 결정한다. 위의 수식에서 f_t 층이며 이 층은 h_{t-1} 과 x_t 값을 받아 0~1 사이의 실수를 출력한다. 다음 단계로 셀 상태에 새로운 정보를 저장할지 결정하는데, 이 단계는 두 부분으로, 먼저 입력 게이트 i_t 층에서는 어떤 값을 추가할지를 결정한다. 그리고 \tanh 층은 새로운 후보 값 C_t 를 만들어 셀 상태에 더해질 수 있도록 한다. 그리고 앞의 두 값을 합쳐 셀 상태 갱신을 위한 값을 만든다. 마지막으로, h_t 의 값을 결정하는 영역으로 시그모이드 출력 게이트 o_t 층을 이용해 셀 상태 C_t 에서 어떤 부분들을 h_t 로 출력할지 결정한다. 그리고 값이 -1~1사이가 되도록 C_t 를 \tanh 에 넣고 선택한 부분만 출력하도록 o_t 층과 요소 간 곱셈으로 곱해 h_t 를 출력한다. 이렇게 학습된 LSTM은 초기 입력 정보를 더 오래 보존할 수 있다.

<그림 5>는 기본 LSTM의 성능을 더욱 개선한 모델로 네트워크의

입력을 정방향과 역방향으로 동시에 학습을 진행하는 양방향 순환 신경망으로 다양한 자연어 처리 연구에 이용되고 있다[15].



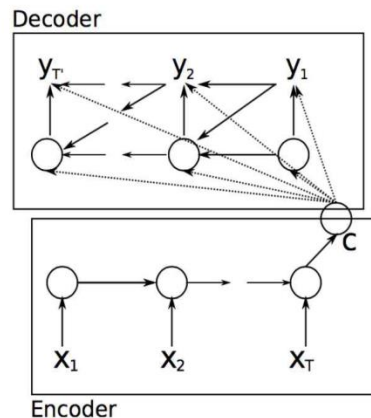
<그림 5> 양방향 순환 신경망 다이어그램

다양한 실험을 통해 순환 신경망 기반 자연어처리 모델은 계속 발전되고 있으며, 또한 기존 연구 대부분은 영어 또는 라틴어로 수행되었으나 최근에는 한국어를 이용한 연구 사례도 점차 증가하고 있다. (김민정 et al., 2016)의 연구에서는 LSTM을 이용하여 한국어 대화체 문장의 감정과 화행, 서술자를 동시에 분석하는 통합 대화 분석모델을 제안했으며[16] (이치욱 et al., 2017)의 연구는 한국어 이슈 관리 시스템에서 전체 모듈 개수 32개, 이슈 리포트 13,474개 데이터 중에서 데이터 개수가 상위 15개 모듈, 이슈 리포트 12,625개를 대상으로 제목,

본문 정보를 수집하여 CNN-LSTM 모델을 이용해 SVM, CNN, LSTM보다 분류 성능이 개선되는 효과를 보였다[17].

2.3. Attention 기반 인공지능망을 이용한 연구

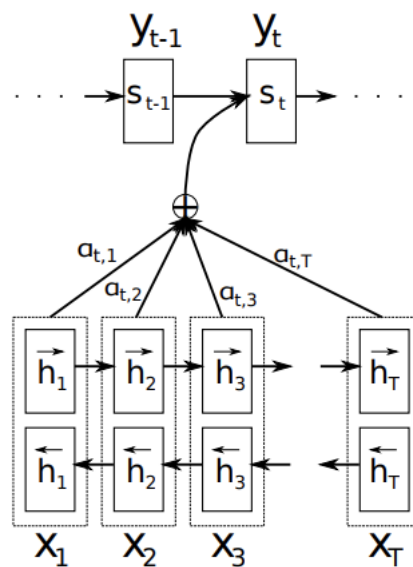
시퀀스 투 시퀀스는 동일 의미의 영어 문장-프랑스어 문장 쌍을 이루는 데이터에 대해 유용한 학습 방법으로 기계 번역 문제에 딥 러닝 기반 학습을 적용한 기계 번역 연구를 통해 처음 소개되었다[18][19]. 이 학습 방법은 순환 신경망의 한 종류이며 <그림 6>는 순환 신경망이 여러 개 연결된 인코더와 디코더의 구조를 보여준다. 데이터 입력을 받는 소스 시퀀스가 출력 데이터인 타겟 시퀀스로 변환되는 구조를 기본으로 한다.



<그림 6> 시퀀스 투 시퀀스의 인코더 디코더(출처: [18])

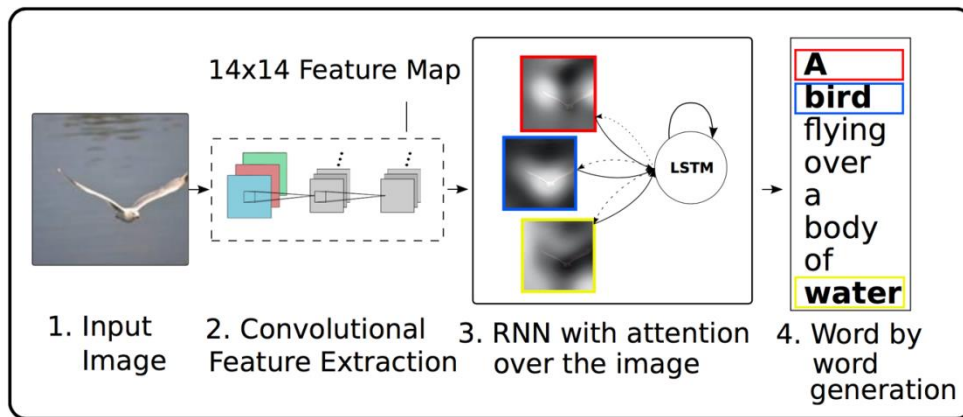
입력 데이터 길이가 길어질수록 시퀀스 투 시퀀스 학습은 인코더가 생성한 context 벡터가 입력 데이터의 정보를 제대로 담기 어려워지는 문제가 있다. 이를 해결하기 위해 디코딩 과정에서 context 벡터를 동적으로 생성하는 Attention 기반 시퀀스 투 시퀀스 모델이

연구되었다[19]. 기본적인 시퀀스 투 시퀀스는 디코더가 context 벡터로만 모든 출력을 생성했다면, Attention 기반 시퀀스 투 시퀀스는 디코더가 각 시퀀스마다 서로 다른 context 벡터를 활용한다. 이때 문맥 벡터들은 인코더의 각 시퀀스에서 생성된 모든 은닉 상태의 가중치 합으로 나타내진다.



<그림 7> Neural Machine Translation Attention Mechanism(출처: [19])

(Kelvin et al., 2016)의 연구는 영상에 대한 설명을 자연어로 생성하는 연구이며[20] Attention을 이용해 기존 연구보다 좋은 성능을 보였다. <그림 6>은 학습 방식을 설명하고 있으며 2번 항목에서 특징 맵을 추출하고 3번 항목에서는 추출된 특징 맵과 LSTM으로 생성한 단어에 Attention을 적용하여 다음 특징 맵의 추출과 단어를 생성한다.



<그림 8> Show Attend and Tell Memory(출처: [20])

제 3 장 Attention 기반 LSTM 신경망을 이용한 한국어

화행 분석 방법

본 장에서는 논문에서 제안하는 Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석 방법 등을 중점적으로 살펴본다. 3.1 에서는 한국어 문장의 벡터 표현을 이용한 화행 분석 방법 및 이를 구성하는 파라미터들을 정의하고 3.2 에서는 Attention을 이용한 화행 분석 방법을 제시한다.

3.1. 순환 신경망을 이용한 한국어 문장의 벡터 표현 방법

순환 신경망은 연속적인 데이터를 처리하는데 적합한 구조이며 발화의 단어를 차례대로 입력받으며 각 단어가 임베딩 계층을 거치면서 정해진 차원의 벡터로 위치하게 되고, 이 벡터는 신경망의 은닉층에 입력으로 사용된다.

한국어의 경우 교착어에 속하는 특성으로 인해 영어의 단어 임베딩을 생성하는 과정에서 사용되는 어절을 공백으로 구분하는 방법은 적합하지 않다. 다양한 조사, 어미로 인해 다른 어절이 같은 형태소를 포함하는 경우가 많으므로, 이를 고려한 단어 임베딩을 생성해야 한다[21]. 본 논문에서는 형태소를 기본 단위로 하는 한국어 단어 임베딩을 은닉층에

입력으로 사용하였고 효과적인 임베딩 크기 확인을 위해 [21]에서 제안하는 100, 200, 300 크기에 대해 실험을 하였다.

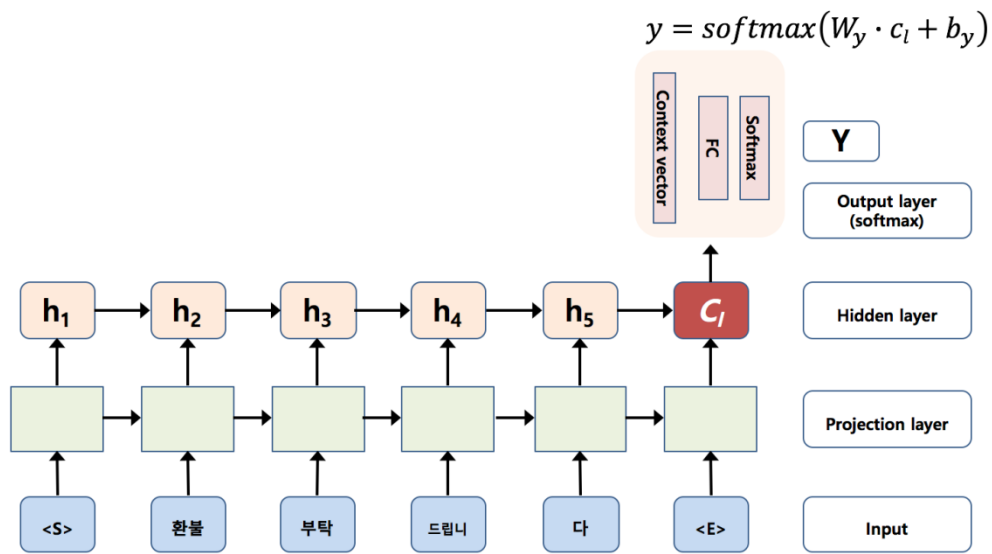
네트워크의 출력은 입력으로 사용된 발화의 화행이며 네트워크에서의 출력값과 정답의 차이에 대해 크로스 엔트로피값을 역전파 알고리즘을 이용해 네트워크의 파라미터를 학습시켰다. 본 연구에서는 순환 신경망 은닉층의 마지막 시퀀스 출력을 context 벡터의 일부로 사용했다.

$$c_l = h_T$$

$$y = \text{softmax}(W_y c_l + b_y)$$

c_l 는 문장을 요약한 문맥 벡터이며, T 는 문장의 길이를 뜻하고, h_t 는 마지막 LSTM의 은닉층이다. y 는 시스템의 출력층이며 softmax¹⁰를 사용해 각 화행 클래스의 확률을 얻는다. <그림 9>은 LSTM 신경망을 이용해 한국어 문장의 context 벡터를 구하는 구조이다.

10 https://en.wikipedia.org/wiki/Softmax_function



<그림 9> LSTM Context Vector

3.2. Attention 기반 LSTM 신경망을 적용한 방법

Attention 기반 LSTM 신경망은 각 시퀀스의 은닉층 출력의 중요도인 가중치를 각 은닉층 출력에 가중 합을 계산하여 attention 벡터를 구한다. attention 벡터를 구하기 위한 가중치는 학습을 통해 얻으며 수식은 다음과 같다.

$$s_t = V_a^T \tanh(W_a h_t + b_a)$$

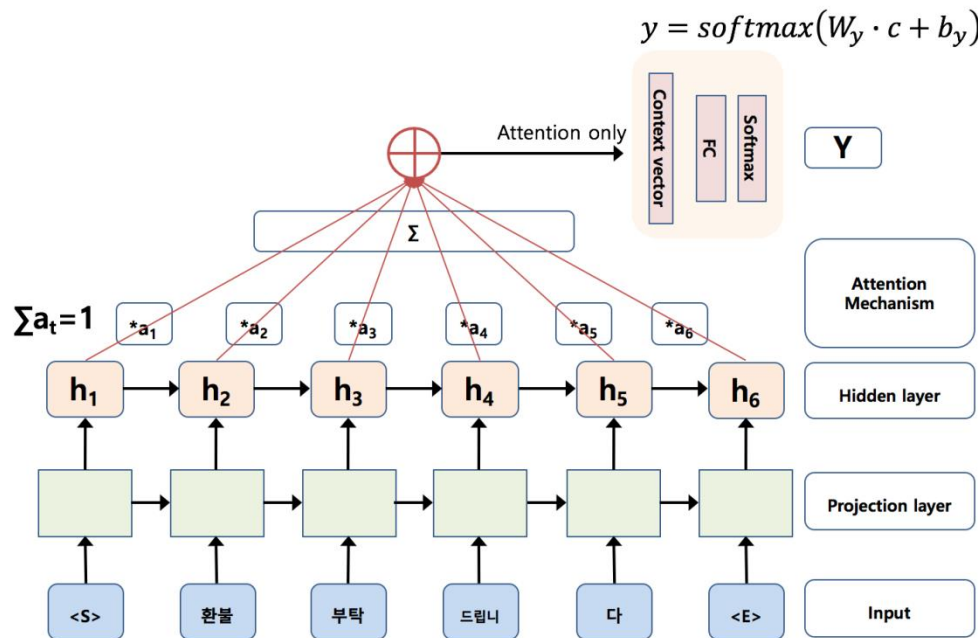
$$a_t = \frac{\exp(s_t)}{\sum_{k=1}^{T_x} \exp(s_k)}$$

$$c = \sum_{t=1}^{T_x} a_t h_t$$

수식에서 V_a 는 attention 을 구하기 위한 가중 벡터이고, h_t 는 수식에서 구한 각 시퀀스의 은닉층의 출력을 나타낸 벡터이며, W_a 는 attention 을 구하기 위해 학습되는 가중 행렬이고, b_a 는 Bias 이다. a_t 는 attention 벡터를 구할 때 각 h_t 에 대해 가중치 값이 된다. a_t 는 각 h_t 에 가중치로 곱하여 가중 합으로 attention 벡터를 구하고 문장 내에서 결과에 영향을 크게 미치는 h_t 를 강조하게 된다.

context 벡터를 구하는 방법으로 LSTM 신경망에서는 은닉층의 마지막 시퀀스 출력만을 사용했지만, Attention 기반 LSTM 신경망에서는 attention 벡터를 함께 사용하였다.

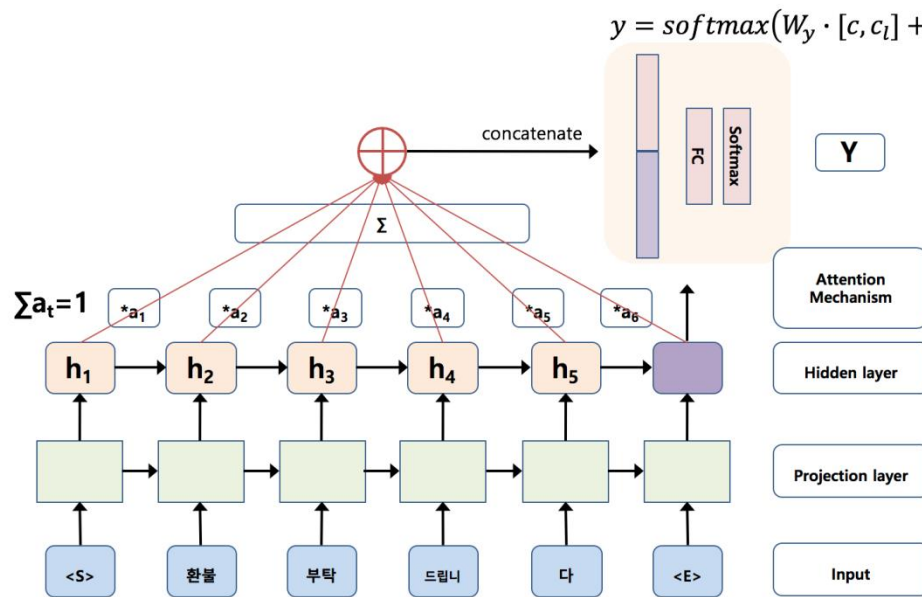
본 논문에서는 context 벡터를 구하는 방법 세 가지를 제안하고 4.2에서 각 방법을 실험하였으며 성능을 비교 평가하였다.



<그림 10> attention 벡터를 context 벡터로 사용한 방법

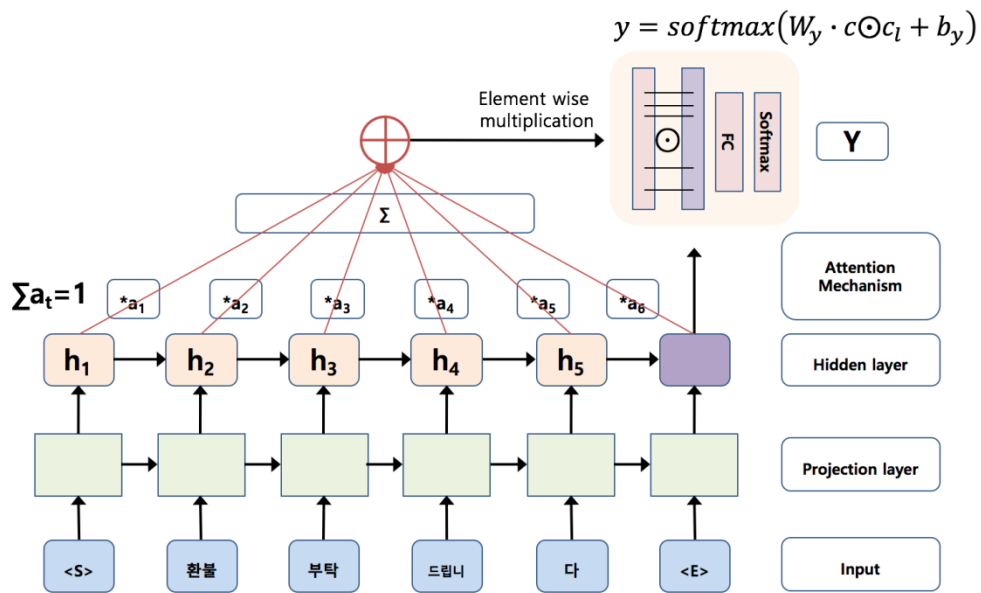
<그림 10> 은 context 벡터를 구하는 첫 번째 방법으로 각 시퀀스의 가중치 정보를 가진 attention 벡터만을 context 벡터로 사용해 학습하는

모델로 각각의 입력 시퀀스에서 중요도가 높은 형태소에 많은 가중치를 가진 기본 모델이다.



<그림 11> attention 벡터와 마지막 시퀀스를 연결한 방법

<그림 11>은 은닉층의 마지막 시퀀스 출력 벡터를 attention 벡터에 연결하여 context 벡터를 생성하는 방법이다. context 벡터에 더 많은 정보를 담기 위해 attention 의 가중치 정보와 LSTM 신경망의 마지막 시퀀스 정보를 연결한 모델이다.



<그림 12> attention 벡터와 마지막 시퀀스 요소 간 곱을 사용한 방법

<그림 12>은 은닉층의 마지막 시퀀스 출력 벡터와 attention 벡터의 차원이 동일하므로 두 벡터를 요소 간 곱을 하여 context 벡터에 attention 벡터와 마지막 시퀀스 출력 벡터의 정보를 섞어주는 모델이다.

제 4 장 실험 및 결과

본 장에서는 화행 분석을 위한 한국어 데이터를 소개하고 제안하는 모델의 성능을 비교 평가한다.

4.1. 실험 데이터 수집

화행 분류 체계는 각 분야의 데이터마다 다양하게 수립될 수 있기에 본 연구에서는 게임 분야 질의응답 상황에서 사용할 수 있는 도메인으로 실험 데이터를 수집했다. 22 개의 게임 분야 SNS 및 리뷰 게시글을 2016 년 6 월부터 5 개월 동안 각 SNS 에서 제공하는 API¹¹를 이용해 한국어 문장 607,000 개를 수집하였으며 요청 화행, 향의 화행, 질문 화행, 진술 화행 4 가지로 분류해 12,000 개씩 총 48,000 개의 학습 및 평가 셋을 구축하였다. 다만 각 화행의 라벨링 작업에서 두 가지 화행이 섞여 있는 경우에는 높은 우선순위의 화행을 라벨링 하였으며 우선순위를 향의 화행,

11 <https://developers.facebook.com/docs/graph-api/using-graph-api>

질문 화행, 요청 화행, 진술 화행 순으로 하여 다수의 화행이 포함되어도 높은 우선순위의 화행을 라벨링 하였다.

효율적인 임베딩¹²을 위해 수집된 607,000 개의 문장에서 평가 셋과 도메인 무관한 데이터를 제외하고, 형태소 추출 후 숫자나 특수기호를 제거한 출현 빈도가 높은 상위 30,000 개의 단어를 사전으로 사용하였으며 상위 30,000 개가 아닌 단어들은 <UNK>(unknown) 토큰으로 변형시켜 처리하였다.

데이터 셋의 구성은 순서나 길이의 영향을 받지 않도록 무작위로 섞은 후 <표 2>와 같이 학습 데이터 36,000 개, 검증 데이터 4,000 개, 테스트 데이터 8,000 개로 구성하였다.

<표 2> 데이터 셋 구성

화행 분류	학습 데이터	검증 데이터	테스트 데이터	전체
질문	9,000	1,000	2,000	48,000
항의	9,000	1,000	2,000	
요청	9,000	1,000	2,000	
진술	9,000	1,000	2,000	

12 https://en.wikipedia.org/wiki/Word_embedding

4.2. 실험 설계

본 연구에서는 기본이 되는 RNN 과 LSTM 신경망의 화행 분석 성능을 먼저 평가하고 Attention 기반 LSTM 신경망의 성능을 비교하였으며 3.2 에서 제안한 attention 벡터를 이용해 context 벡터 조합하는 세 가지 방법에 대해 성능을 비교 평가하였다

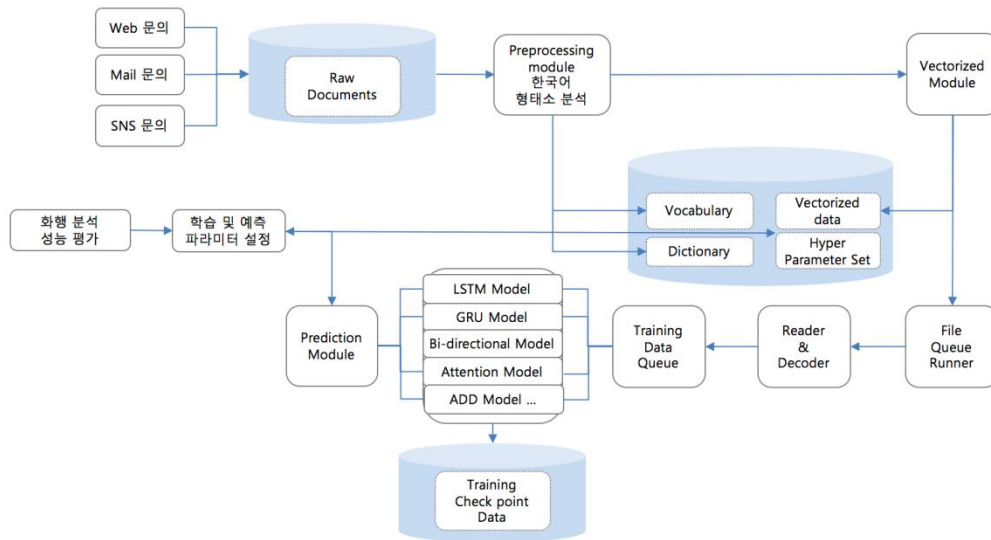
먼저 한국어에 적합한 단어 임베딩 크기의 파라미터를 설정하기 위해 [21]에서 제안하는 100~300 차원의 크기로 분류의 성능을 확인하였으며 시퀀스 길이는 100 으로 하여 단문의 경우에는 여백을 <PAD> 토큰으로 변형하여 시퀀스 길이에 대해 처리를 하였다. 그리고 RNN 계열 모델 학습 시에 그라디언트 폭발 문제를 예방하기 위해 파라미터의 값이 5 이상이 될 때 그라디언트 클리핑 처리를 하였으며 옵티마이저는 Adam¹³과 Rmsprop¹⁴ 두 가지를 비교 실험하였다.

마지막으로 attention 벡터에 대한 가중치 스코어를 문장의 형태소와 함께 확인할 수 있도록 구현하여 attention 층이 학습이 잘 되었는지 확인하고, 화행 분류 시 어느 형태소의 가중치가 높은지 확인하였다.

¹³ <https://arxiv.org/abs/1412.6980>

¹⁴ <http://climin.readthedocs.io/en/latest/rmsprop.html>

<그림 13>은 반복되는 실험을 학습 및 평가하기 위한 프레임워크 구조의 학습 시스템 구성도이다.



<그림 13> 학습 시스템 구성도

본 논문에서는 정확도(Accuracy)를 성능 평가 척도로 사용하였으며 다음 수식으로 정확도를 측정하였다.

$$Accuracy(A) = \frac{C}{T}$$

위 수식에서 T 는 전체 테스트 문장의 개수이고, C 는 예측 결과가 정답을 맞힌 문장의 개수이며 정답 일치 여부를 판단할 때, 예측 출력값이 준비된 정답과 정확히 일치한 경우에만 정답을 맞혔다고 간주하였다.

4.3. 실험 결과 및 분석

본 연구에서 성능 정확성을 위해 분류 성능 측정의 신뢰도를 높이기 위해서 10 겹 교차 검증¹⁵과 을 적용하였으며 <표 3>는 실험 환경이다.

<표 3> 실험 환경

	구분	명세
서버 사양	CPU	Intel i7-7700K
	RAM	32G
	GPU	Nvidia-1080 Ti 11GB
	머신러닝 라이브러리	Tensorflow 1.0

<표 4>는 본 실험에서 소개하는 신경망 모델 및 임베딩 사이즈에 대한 성능을 나타낸 표이다. LSTM-attn-only 모델은 attention 벡터를 context 벡터로 사용한 모델이며 RNN 보다 1.4%, LSTM 보다 0.3% 근소하게 높은 성능이 나왔으며 [21]에서 한국어 단어 임베딩에 적합한 사이즈로 제안된 100, 200, 300 크기는 모두 비슷한 성능을 보였다.

¹⁵ [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

<표 4> 모델간 성능 비교

model	optimizer	word embedding	dropout	acc
RNN	Adam	100	0.5	0.855
		200		0.856
		300		0.856
LSTM	Adam	100	0.5	0.866
		200		0.867
		300		0.866
LSTM-attn-only	Adam	100	0.5	0.869
		200		0.870
		300		0.870

<표 5>는 3.2 에서 제안한 attention 벡터를 조합하는 세 가지 방법에 대한 성능을 보여주고 있으며 LSTM-attn-mul 의 성능이 가장 높게 나왔다. <표 4> 에서 가장 좋은 성능을 보인 LSTM-attn-only 보다 0.4% 더 높은 성능을 보였다.

<표 5> attention context vector 비교

model	acc
RNN	0.856
LSTM	0.867
LSTM-attn-only	0.870
LSTM-attn-concat	0.872
LSTM-attn-mul	0.874

실험한 모델 간 성능 차이가 근소하여 10 겹 교차 검증 테스트의 결과로 대응표본 T-검정을 수행하였다. <표 6>은 비교한 모델간

유의확률을 나타낸 표이며 각 유의확률은 유의수준 0.05 보다 낮았고 통계적 유의수준 하에서 각 모델 간 성능 차이가 있는 것을 확인하였다.

<표 6> 모델간 대응표본 T-검정 유의확률

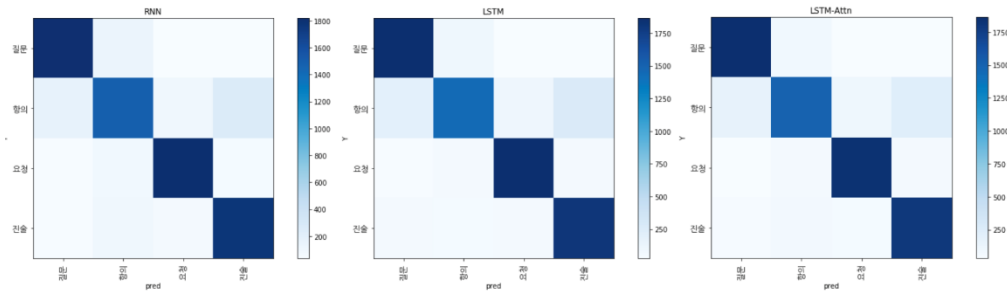
Model 1	Model 2	Paired T-Test p-value
LSTM	LSTM-att-only	0.027
LSTM-attn-only	LSTM-attn-concat	0.024
LSTM-attn-concat	LSTM-attn-mul	0.026
LSTM	LSTM-attn-mul	7.177e-07

성능이 더 높은 함수의 조합을 찾기 위해 attention 구현 내부의 활성화 함수를 tanh 에서 relu 로 변경하고, optimizer 은 Adam 과 Rmsprop 을 비교 실험하였다. 본 실험에서는 초기 조합으로 사용했던 tanh 와 Adam 이 높은 성능을 보였다.

<표 7> 활성화 함수와 Optimizer 비교

model	Optimizer	Activation	acc
LSTM-attn-mul	Adam	tanh	0.874
		relu	0.872
	Rmsprop	tanh	0.870
		relu	0.869

<그림 14>는 각 화행의 정답률을 확인하기 위한 혼동행렬¹⁶이며 첫 번째는 RNN, 두 번째는 LSTM, 세 번째는 LSTM-attn 에 대한 혼동 행렬이다. 항의 화행 색이 상대적으로 흐리게 나타난 것은 정답률이 낮은 것을 뜻하며 <표 8>에서도 항의 화행의 정답 숫자가 낮은 것을 알 수 있다.



<그림 14> 화행의 혼동 행렬

<표 8> 화행의 혼동 행렬

model	정답 \ 예측	질문 화행	항의 화행	요청 화행	진술 화행
RNN	질문 화행	1809	135	37	39
	항의 화행	165	1494	93	268
	요청 화행	49	92	1817	62
	진술 화행	49	94	77	1770
LSTM	질문 화행	1864	96	28	32
	항의 화행	191	1453	103	273
	요청 화행	42	61	1860	57
	진술 화행	60	53	70	1807
LSTM attn	질문 화행	1862	87	36	35
	항의 화행	173	1512	102	233
	요청 화행	38	75	1843	64
	진술 화행	50	83	58	1799

16 https://en.wikipedia.org/wiki/Confusion_matrix

본 실험의 데이터 셋은 4.1 에서 언급했듯이 다수의 화행이 섞인 경우 항의 화행의 순위를 가장 높게 라벨링 하여 항의 화행은 다른 화행의 포함한 데이터가 많기에 성공률이 비교적 낮다. <표 9> 에서 precision, recall¹⁷, f1-score¹⁸를 평가하였고 항의 화행의 f1-score 가 가장 낮게 나온 것을 확인하였다.

<표 9> Precision, Recall, f1-score

model	Target	precision	recall	f1-score
RNN	질문 화행	0.87	0.90	0.88
	항의 화행	0.82	0.74	0.78
	요청 화행	0.90	0.90	0.90
	진술 화행	0.83	0.89	0.86
LSTM	질문 화행	0.86	0.92	0.89
	항의 화행	0.87	0.72	0.79
	요청 화행	0.90	0.92	0.91
	진술 화행	0.83	0.91	0.87
LSTM-attn	질문 화행	0.89	0.90	0.90
	항의 화행	0.85	0.76	0.80
	요청 화행	0.91	0.92	0.91
	진술 화행	0.84	0.87	0.87

하나의 화행으로 구성된 데이터 대부분은 예측에 성공했으며, 실패 데이터는 두 가지 이상의 화행이 섞인 경우가 많았다. <표 10>은 정성적인 평가를 위해 성공과 실패한 문장을 확인해 보았으며 두 가지 화행이 포함된 문장을 평가하였다.

17 https://en.wikipedia.org/wiki/Precision_and_recall

18 https://en.wikipedia.org/wiki/F1_score

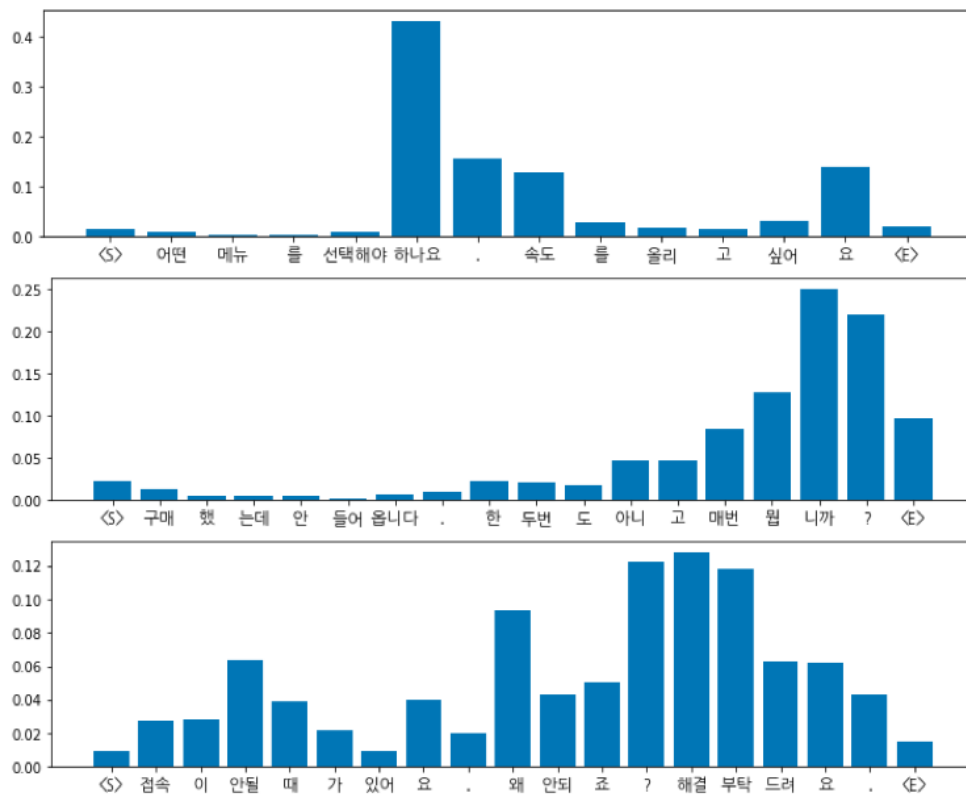
<표 10> 발화 문장 결과표

No	발화 문장	정답	예측	결과
1	어떤 메뉴를 선택해야 하나요. 속도를 올리고 싶어요.	질문	질문	Success
2	구매 했는데 안들어 옵니다. 한 두 번도 아니고 매번 뭘니까?	항의	항의	Success
3	접속이 안될때가 있어요. 왜 안되죠? 해결 부탁 드려요.	요청	요청	Success
4	이전 버전과 많은 차이점은 모르겠어요. 하지만 속도는 더욱 빨라져서 좋아요.	진술	진술	Success
5	다이하 복구되었는지 궁금한데 결과 알려주세요.	질문	요청	Fail
6	환불 해 달라는데 뭐가 이리 복잡합니까?	항의	질문	Fail
7	아니 이렇게 얘기하면 해결안해준다는 거예요?	항의	질문	Fail

예측에 성공한 1, 2, 3, 4 항목은 두 가지 화행이 포함되어 있지만, 예측에 성공하였고 5, 6, 7 항목은 예측에 실패한 경우이다. 예측에 성공한 데이터는 두 가지 화행이 포함되어도 다수의 문장이나 긴 문장으로 구성되었고 두 가지 화행이 하나의 짧은 문장으로 구성된 경우 실패가 많았다.

<그림 15>는 예측에 성공한 1, 2, 3 항목에서 attention 벡터에 사용된 가중치를 나타낸 그래프이다. 1 번 항목 질문 화행에서는 “하나요” 형태소 가중치가 높게 나왔고, 2 번 항의 화행에서는 “뭘니까?”의 가중치가 높았으며 3 번 요청 화행에서는 “해결 부탁드립니다” 가 높았다.

각 화행의 특성이 되는 형태소 자질에 가중치가 높았으며 이를 통해 attention의 가중치가 각 화행에 따라 학습되고 있는 것을 확인하였다.



<그림 15> Attention을 이용한 화행 별 형태소 가중치

제 5 장 결론 및 향후 연구

본 장에서는 실험 결과에 대한 평가 및 연구의 한계점과 향후 연구에 관하여 기술한다.

5.1. 연구의 한계점 및 향후 연구

본 논문은 Attention 기반 LSTM 신경망을 이용한 한국어 화행 분석에 대해 실험하였다. 한국어 화행 데이터에 대해 분류를 수행하고 attention 벡터의 조합 방법 세 가지를 제안하였으며 Attention 을 이용해 화행의 자질이 되는 형태소에 가중치가 학습되는 것을 확인하였다.

attention 벡터를 조합하는 세 가지 방법 중에는 기본 attention 벡터만을 사용하는 LSTM-attn-only 모델이나 은닉층의 마지막 시퀀스 출력 벡터를 attention 벡터에 연결하는 LSTM-attn-concat 모델보다 마지막 시퀀스 출력 벡터를 attention 벡터와 요소별 곱 연산을 하는 LSTM-attn-mul 모델이 근소하게 높은 성능을 보였다.

하지만 라벨링 된 데이터 수가 딥 러닝에서 사용하기에는 적었고 SNS 에서 수집된 데이터 셋 대부분이 단문이었기에 긴 문장에서

성능향상을 보이는 LSTM 과 Attention 기반 모델에서 성능향상의 폭이 낮았으며 글을 수집한 SNS 원본 문장에 도메인 무관한 광고성 문구, 오타, 신조어, 띄어쓰기 오류가 많이 있어 데이터 전처리 및 형태소 분석 단계에서 많은 손실이 있었다.

본 연구는 단일 화행 분석만을 실험하였지만 두 가지 이상의 화행이 포함된 문장에 대해 다수의 화행을 함께 라벨링하고 학습을 수행하여 다수의 화행을 통합 분석하는 실험과 다양한 한국어 분석 도메인에 Attention 기반 인공지능망을 적용한 연구를 제안한다.

참고 문헌

- [1] Song, N., Bae, K., & Ko, Y. (2016). 분류 우선순위 적용과 후보정 규칙을 이용한 효과적인 한국어 화행 분류 Rules. Journal of KIISE, 43(1), 80-86.
- [2] 김경선, 서정연 (2003). 자질 선택 기법을 이용한 한국어 화행 결정. 정보과학회논문지 : 소프트웨어 및 응용, 30(3·4), 278-284.
- [3] 유동현 (2016). 다양한 어휘 자질의 워드 임베딩을 이용한 화행 분석 추출. 1-60.
- [4] C . Doran, et al., "Comparing Several Aspects of Human -Computer and Human-Human Dialogues," Current and New Directions in Discourse and Dialogue, pp. 133-159, 2003.
- [5] 이현정, 김학수, 서정연 (2008). 일정 관리 영역에서의 화행 분석을 위한 효과적인 2 단계 모델. 인지과학, 19(3), 297-310.
- [6] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2017). Recent Trends in Deep Learning Based Natural Language Processing. arXiv preprint arXiv:1708.02709.

- [7] 최상혁 (2017). 음절 기반 한국어 단어 임베딩 모델 및 학습 기법, 1-50.
- [8] Austin, J.L. (1962). How to do things with words. Oxford: OUP.
- [9] 김세중, 이용훈, 이종혁 (2008). 이전 문장 자질과 다음 발화의 후보 화행을 이용한 한국어 화행 분석. 정보과학회논문지 : 소프트웨어 및 응용, 35(6), 374-385.
- [10] Silva, L. Coheru, A. Mendes, A. Wichert. 2011. From symbolic to sub-symbolic information in question classification. Artificial Intelligence Reviews, 32(1):137-154.
- [11] 김동준, 이연수, 장정선, 임해창 (2015). 한국어 대화 화행 분류를 위한 어휘 자질의 임베딩. 한국정보과학회 학술발표 논문집, 575-577.
- [12] G.E. Hinton, R.R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. Science, 313(5786):504-507.
- [13] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. Cognitive modeling, 5(3), 1.

- [14] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*. 9:1681-1726.
- [15] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." *Signal Processing, IEEE Transactions on* 45.11 (1997): 2673-2681.2. Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan.
- [16] 김민경, 김학수 (2016). Long Short-Term Memory 를 이용한 통합 대화 분석, 1-3.
- [17] 이치욱(2017). CNN_LSTM 모델을 이용한 이슈 관리 시스템의 모듈 분류 연구. 1-59.
- [18] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104-3112).
- [19] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

[20] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. (2016). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, 1-22.

[21] 최상혁, 설진석, 이상구 (2016). 한국어에 적합한 단어 임베딩 모델 및 파라미터 튜닝에 관한 연구, 1-5.

[ABSTRACT]

Analysis of Korean Speech-act using Attention-based LSTM

Cho, Lyun

Major in Computer Engineering

The Graduate School of Engineering

Yonsei University

Supervised by Prof. Cho, Sung Bae

Speech act indicates the intention of speakers through utterance. In conversational Question Answering System, it's important to analyze the questions from the users to provide the right response to meet the intention of the users. Existing researches for the analysis of speech act have required lots of studying and engineering to analyze the information of utterance. but in the latest natural language process or research area, active researching is being done using AI neural network showing the extraordinary performance.

But most of the part of the natural language using AI neural network is based on the Latin language and the Korean language based

ones are very less and there's been nothing for researches of Korean speech act based on Attention showing outstanding performances in AI neural network machine interpretation.

Therefore, in this paper, game related sentences posted on SNS up to 607,000 ones have been collected to analyze the speech act. 48,000 trained data have been created for Request, Complain, Question, Statement Speech Act. Korean Speech Act Analysis model has been proposed using neural network based on Attention LSTM.

Through the experiment, embedding parameters suitable for Korean language and performance comparison assessment by the Attention combination method have been done and the abstraction of the weight value from the trained morphological elements, I could confirm that it gives the more weights to the morphological elements of the feature of the speech act.

Keywords: Speech act, Intents, Machine Learning, Deep Learning, RNN, LSTM, Attention