기술 용어에 대한 한국어 정의 문장 자동 생성을 위한 순환 신경망 모델 활용 연구*

Research on the Utilization of Recurrent Neural Networks for Automatic Generation of Korean Definitional Sentences of Technical Terms

최 가 람 (Garam Choi)**, 김 한 국 (Han-Gook Kim)***
김 광 훈 (Kwang-Hoon Kim)****, 김 유 일 (You-eil Kim)*****
최 성 필 (Sung-Pil Choi)******

목 차

초 록

본 논문에서는 지속적으로 커져가는 산업・시장에 대해 관련 연구자들이 이를 효율적으로 분석할 수 있는 반자동 지원 체제 개발을 위한 기술 용어와 기술 개념에 대한 정의문 및 설명문을 자동으로 생성하는 한국어 문장 생성 모델을 제시한다. 한국어 정의 문장 생성을 위하여 딥러닝 기술 중 데이터의 전/후 관계를 포함한 시퀀스 레이블링이 가능한 LSTM을 활용한다. LSTM을 근간으로 한 두 가지 모델은 기술명을 입력할 시 그에 대한 정의문 및 설명문을 생성한다. 다양하게 수집된 대규모 학습 말뭉치를 이용해 실험한 결과, 본 논문에서 구현한 2가지 모델 중 CNN 음절 임베딩을 활용한 어절 단위 LSTM 모델이 용어에 대한 정의문 및 설명문을 생성하는데 더 나은 결과를 도출시킨다는 사실을 확인하였다. 본 논문의 연구 결과를 바탕으로 동일한 주제를 다루는 문장 집합을 생성할 수 있는 확장 모델을 개발할 수 있으며 더 나아가서는 기술에 대한 문헌을 자동으로 작성하는 인공지능 모델을 구현할 수 있으리라 사료된다.

ABSTRACT

In order to develop a semiautomatic support system that allows researchers concerned to efficiently analyze the technical trends for the ever-growing industry and market. This paper introduces a couple of Korean sentence generation models that can automatically generate definitional statements as well as descriptions of technical terms and concepts. The proposed models are based on a deep learning model called LSTM (Long Sort-Term Memory) capable of effectively labeling textual sequences by taking into account the contextual relations of each item in the sequences. Our models take technical terms as inputs and can generate a broad range of heterogeneous textual descriptions that explain the concept of the terms. In the experiments using large-scale training collections, we confirmed that more accurate and reasonable sentences can be generated by CHAR-CNN-LSTM model that is a word-based LSTM exploiting character embeddings based on convolutional neural networks (CNN). The results of this study can be a force for developing an extension model that can generate a set of sentences covering the same subjects, and furthermore, we can implement an artificial intelligence model that automatically creates technical literature.

키워드: 문장 생성, 텍스트 생성, 자연어 생성, 보고서 자동 생성, 딥 러닝
Sentence Generation, Text Generation, Natural Language Generation(NLG), Automatic Report Generation, Deep Learning

---

* 본 연구는 2017년도 한국과학기술정보연구원(KISTI) 주요사업 과제로 수행하였음.
** 경기대학교 일반대학원 문헌정보학과 석사과정(garam1310@kyonggi.ac.kr) (제1저자)
*** 한국과학기술정보연구원 산업정보분석실 책임연구원, 과학기술연합대학원대학교 과학기술정보과학과 전임교수(hgkim712@kisti.re.kr) (공동저자)
**** 한국과학기술정보연구원 산업정보분석실 선임연구원(kh.kim@kisti.re.kr) (공동저자)
***** 한국과학기술정보연구원 산업정보분석실 책임연구원(yekim@kisti.re.kr) (공동저자)
****** 경기대학교 문헌정보학과 조교수(spchoi@kgu.ac.kr) (교신저자)
논문접수일자: 2017년 10월 16일 최초심사일자: 2017년 10월 25일 게재확정일자: 2017년 11월 13일
한국문헌정보학회지, 51(4): 99-120, 2017. [http://dx.doi.org/10.4275/KSLIS.2017.51.4.099]

1. 서 론

2014년 글로벌 시장조사기관 Gartner의 시장 보고서 “Market Share Analysis: Business Intelligence and Analytics Software”에 따르면 기업의 의사 결정을 위한 BI와 분석 소프트웨어 시장에 대해 연간 8%의 성장을 예측했다. 한국의 비즈니스 컨설팅 시장 또한 2014년 기준 약 2조 5,800억 원의 규모를 형성하고 있으며, 2015년은 이보다 4.2% 성장한 약 2조 7,000억 원의 시장규모로 증가될 것을 예측했다(Woodward, Sood and Hare 2016).

점차 큰 규모의 시장이 구성됨에 따라 산업・시장 분석은 지속적으로 그 중요성을 더해가고 있다. 급격하게 증가하는 시장규모로 인해 각 분야의 관련 정보 또한 빠른 속도로 증가하고 있으며 이는 관련 연구자들이 연구 분야의 학술적 성과 및 유관 기술을 이해하고 분석하는 데 많은 어려움을 유발한다. 또한 연구자들은 산업・시장 분석을 위해 자료 검색, 보고서 작성 등에 많은 시간을 투자해야 하므로 이는 큰 부담요소로 작용 될 수 있다. 일반적으로 산업 시장 분석은 해당 산업의 근간을 구성하는 요소 기술에 대한 분석, 시장 현황 및 규모 파악, 미래 시장 전망 등을 포괄하는 복합적이고 정교한 작업이다. 또한 이를 위해 해당 정보가 포함된 다양하고 폭넓은 문헌 검토 및 분석이 필수적이다. 복합적인 분석 작업을 도우며 효율적인 보고서 작성을 지원할 수 있는 반자동 지원 체제의 개발 및 적용은 시급한 문제라 볼 수 있다.

산업 시장 분석 보고서는 기본적으로 기술에 대한 정의, 기술 설명을 포함한 제품 개요, 시장 동향, 향후 전망 등으로 구성되어 있다. 이러한 구성 요소 중 특정 산업의 근간이 되는 기술에 대한 정의와 설명은 전체 보고서의 핵심 사항이 될 수 있는 중요한 항목이다.

최근 기계학습(Machine Learning) 분야의 딥 러닝(Deep Learning)을 기반으로 한 자연어 처리(Natural Language Processing, NLP)는 Deng과 Yu(2014), Bian과 Gao 및 Liu(2014), Zheng과 Chen 및 Xu(2013) 등의 다양한 연구를 통해 진행되고 있다. 자연어 처리의 일종인 자연어 생성(Natural Language Generation, NLG)을 위해서는 텍스트(Sutskever, Martens and Hinton 2011) 및 음악(Boulanger-Lewandowski, Bengio, and Vincent 2012) 생성이 가능한 순환 신경망 네트워크(Recurrent Neural Networks, RNN)를 활용할 수 있다. 또한 이를 응용한 LSTM(Long-Short Term Memory)은 순환 신경망 계층 보다 더 나은 정보 처리 및 정보 저장이 가능한 기술(Hochreiter and Schmidhuber 1997)이다.

LSTM은 과거 시간의 은닉 상태(Hidden State)를 통해 현재 은닉 상태를 예측하고, 여러 개의 은닉 상태를 연결하는 방법을 통해 입력 값의 전/후 관계를 포함한 데이터의 전반적인 특징을 학습할 수 있다. 입력 값의 특징을 학습한 각각의 LSTM 셀 시퀀스는 최종적으로 입력에 대한 결과가 된다. 또한 이미지 및 특징 추출에 유용한 CNN(Convolutional Neural Networks)(Krizhevsky, Sutskever and Hinton 2012)은 필터를 통해 입력 값의 특징을 추출한 후 이를 통해 비슷한 사진 등을 구분할 수 있는 이미지 분류 분야에서 많은 활용이 되어져왔다. 유사한 방식으로 텍스트 특징 추출 또한 가능한 CNN(Kalchbrenner, Grefenstette and Blunsom

2014)은 최근 자연어 처리 분야에서 활발히 활용되고 있다.

본 연구에서는 앞서 언급한 산업 시장 분석 보고서 작성 지원을 위해 일차적인 세부 목표를 선정해 이에 따른 연구를 진행하였다. 보고서의 주제와 핵심 내용을 담고 있는 기술 용어와 기술 개념에 대한 정의문 및 설명문을 자동으로 생성할 수 있는 딥러닝 기반의 특화 언어 생성 모델을 제안한다. 정의문 생성을 위해 기존재하는 딥러닝 모델들을 활용하여 이를 한국어에 적용한 정보 기술 분야의 전문용어 및 정의문 자동 생성 순환 신경망 모델이다. 순환 신경망의 일종인 LSTM 네트워크를 근간으로 음절 및 어절 단위의 문장 생성 모델 2가지를 제안하고 실험을 통해 그 성능을 비교, 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 자연어 문장을 생성하기 위해 다양한 형태로 개발되었던 언어 모델 및 유관 연구에 대하여 설명하고, 3장에서는 개발된 모델들의 근간이 되는 LSTM 기술의 기본 개념 및 원리를 기술한다. 4장에서는 본 연구에서 제안하는 기술 정의문 생성 모델을 세부적으로 설명하고 5장에서는 한국어 기반의 정보기술 분야 대용량 데이터를 학습한 결과인 기술 정의문의 세부 분석을 기술한다. 마지막으로 6장에서는 본 연구의 결론 및 향후 연구 방향을 도출한다.

2. 관련 연구

기존의 많은 자연어 생성 연구는 규칙 기반의 기술이 주를 이루었다. 규칙을 기반으로 한 마크업 언어(Markup Language) 생성기(Bauer, Hoedoro and Schneider 2015), 음성 언어 생성기(Mairesse 2005), 보고서 생성기(Bontcheva and Wilks 2004) 등이 개발되었으며, 통계적 기법을 활용한 자연어 생성기(Langkilde-Geary 2002) 또한 연구되었다. 그러나 이러한 규칙 기반의 생성 방법론은 정보 전달에 대한 계획, 어휘 및 품사를 포함한 문장 구조 분석, 정답을 출력하기 위한 부가 시스템을 필요로 할 뿐만 아니라 사전에 존재하는 표현들에 기반을 둔 자연어 생성만이 가능한 기술이다.

이러한 부가적인 시스템 없이 충분한 학습 데이터만 존재한다면 자동적으로 데이터를 학습하고 그에 따른 자연어 생성이 가능한 기계학습 분야의 심층 학습 기술(Bian, Gao and Liu 2014)이 등장하였다. 심층 학습을 기반으로 한 기존 연구들은 순환 신경망 네트워크를 활용하여 텍스트 생성 연구를 진행하였다. 하지만 순환 신경망은 학습이 어렵다는 단점이 존재하고 이러한 단점을 극복한 LSTM 네트워크가 등장하였다. LSTM을 활용하여 연구된 모델은 다음과 같다. LSTM 셀의 기본 구조를 변형하여 3가지 게이트(입력, 잊기, 출력)에 통제 가능한 요소를 추가하여 성능을 개선한 언어 생성 모델(Sundermeyer, Schlüter and Ney 2012), 기본적으로 순방향으로 진행되는 LSTM 계층에 역방향 계층을 추가하여 양방향 학습이 가능하도록 한 Bidirectional LSTM 활용 연구(Graves, Jaitly and Mohamed 2013), Bidirectional LSTM을 이용하여 문장을 학습하고(Encoder) 학습된 문장에 대해 특정 언어로 번역하는(Decoder) Sequence to sequence 모델 기반의 번역기(Bahdanau, Cho and Bengio 2014), 문서 요약 연구(Nallapati et al. 2016) 또한 진행되었다.

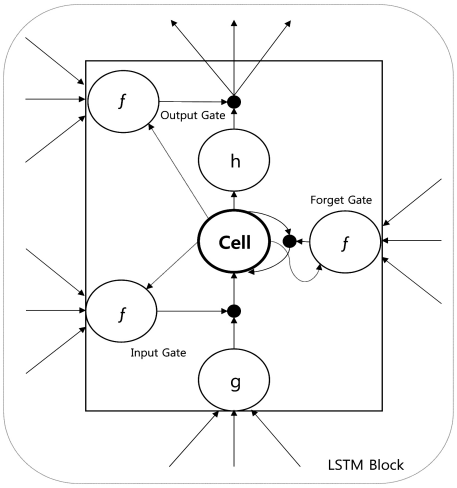
최근에는 문맥 자체의 학습이 가능한 변형된 오토인코더(Variational Autoencoders, VAE)를 활용한 연구(Bowman et al. 2015)도 등장하기 시작하였다.

위와 같이 다양한 자연어 생성 연구가 진행되고 있으나 한국어 기반의 특정 주제를 지니는 기술 정의문 생성 연구는 존재하지 않아 보인다. 이에 따라 다양한 모델을 적용하여 한국어의 언어적 특성을 고려한 자연어 생성 결과의 의문을 해소하고자 본 논문에서는 가장 기초적인 LSTM 순방향 학습 기반의 생성 모델과 이를 확장한 모델을 구성하여 기술 분야 정의 문장 생성 연구를 진행하였다. 모델의 근간이 되는 LSTM의 기본 개념 및 원리는 3장에서 자세히 설명한다.

3. Long Short-term Memory (LSTM)

LSTM은 순환 신경망에 기반을 두는 응용 네트워크이다. 순환 신경망은 자연어, 유전자, 음성 인식 등과 같이 연속적인 형태를 가지는 데이터 처리에 효과적인 신경망이다. 순환 신경망은 이전 단계의 은닉 상태와 현재 단계의 입력 값을 계산해 다음 단계의 은닉 상태를 예측하는 구조를 가지며 주로 생성 및 레이블링 분야에서 활용된다. 그러나 순환 신경망 계층의 입력 값들은 수많은 계층들을 거치면서 그 값이 유지되지 못하고 급격히 감소하거나 폭발적으로 증가하는 현상을 보인다. 이러한 현상은 '기울기 값이 사라지는 문제(Vanishing Gradient Problem)'라고 불리며 문제의 해결 방안으로 LSTM(Hochreiter 1991)이 등장하였다.

LSTM은 〈그림 1〉과 같은 메모리 블록을 여러 개 연결한 형태이다. 하나의 메모리 블록은 입력(Input), 출력(Output), 잊기(Forget) 게이트를 가진다. 이러한 게이트를 사용하여 현재, 이전 단계, 특정 상황에 따른 정보를 골라 사용할 수 있다. 각 게이트들은 곱셈 연산을 통해 서로 연결된다. LSTM 계층에서 은닉 상태는



〈그림 1〉 LSTM(Long Short-term Memory) 블록 구조

식 (6)으로 계산 과정은 다음과 같다.

$$i = \sigma(x_t U^i + s_{t-1} W^i + b^i) \quad (1)$$

$$f = \sigma(x_t U^f + s_{t-1} W^f + b^f) \quad (2)$$

$$o = \sigma(x_t U^o + s_{t-1} W^o + b^o) \quad (3)$$

$$g = \tanh(x_t U^g + s_{t-1} W^g + b^g) \quad (4)$$

$$c_t = c_{t-1} \circ f + g \circ i \quad (5)$$

$$s_t = \tanh(c_t) \circ o \quad (6)$$

위 수식에서 σ는 sigmoid와 같은 비선형 활성화 함수이며, x_t는 현재 시점(t)에서의 입력 값, s_{t-1}은 이전 시점(t-1)에서의 은닉 상태이다. 또한 s_t의 경우 현재 시점(t)의 은닉 상태이고 $\circ$는 요소별 연산(Element - wise Product)을 의미한다. U, W 와 b는 각각 게이트 별 가중치와 바이어스를 뜻한다. 식 (1), (2), (3)은 각각 입력, 잊기, 출력 게이트를 구하는 과정으로 각 수식에서 파라미터 행렬을 제외한 나머지 부분은 동일한 형태이다. 각 게이트는 시그모이드 함수(Sigmoid Function)가 씌워진 형태로 행렬들의 합을 0과 1 사이로 제한한다. 행렬들의 합이 0에 가까울수록 더 적은 양의 정보가 사용되며, 1에 가까울수록 더 많은 양의 정보가 사용된다.

입력 게이트는 새로운 은닉 상태를 계산하기 위해 입력 값을 얼마나 사용할지 결정한다. 잊기 게이트는 이전 시점의 은닉 상태에서 현재 연산에 필요 없는 값을 제외하는 역할을 수행하며, 출력 게이트는 현재 계산된 내부 상태 값 중에서 어떤 정보를 다음 단계로 넘겨줄지를 결정한다.

g는 식 (4)와 같이 현재 입력 값과 이전 시점의 은닉 상태를 기반으로 계산된 출력 값이며, c_t는 LSTM의 내부 메모리로 식 (5)에서와 같이 이전에 저장된 메모리 c_{t-1}과 잊기 게이트 f의 곱, 그리고 g와 입력 게이트 i의 곱을 합친 형태로 계산된다. 순환 신경망 계층은 추가적인 연산 없이 g 자체를 새로운 은닉 상태로 사용했으나, LSTM은 입력 게이트와 잊기 게이트를 추가하여 새로운 은닉 상태를 구성한다. c_t가 계산되면 c_t와 출력 게이트 o의 곱으로 최종적인 은닉 상태 s_t가 출력된다. s_t를 출력하는 과정에서 필요 없는 정보가 내부 메모리에 포함되는 것을 방지하기 위해 출력 게이트를 사용한다. 출력 또한 시그모이드 함수를 통해 그 양이 결정된다.

4. 기술 용어에 대한 한국어 정의 문장 생성 모델

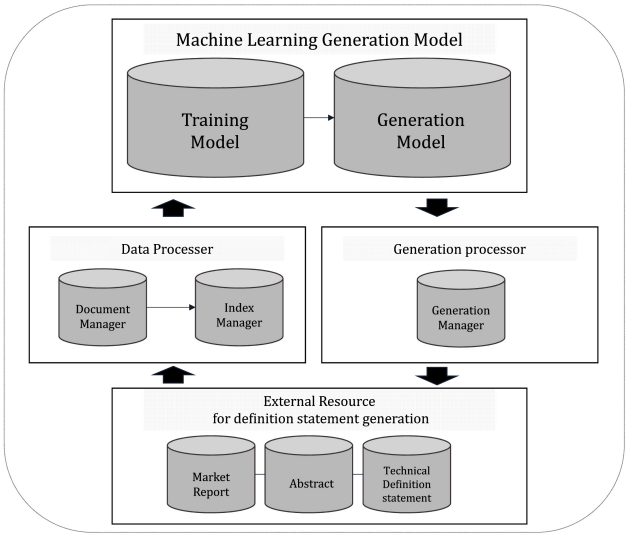
기술 용어에 대한 정의문 및 설명문을 생성하기 위해 본 논문에서 활용한 모델은 총 2가지다. 이에 근간이 되는 기술은 LSTM으로 LSTM은 과거 시간 단계의 은닉 상태를 통해 현재 은닉 상태를 예측한다. 예측된 여러 개의 은닉 상태를 연결하는 방법을 통해 입력 값의 전/후 관계를 포함하고, 데이터의 전반적인 특징을 지닌 결과 값을 생성한다. LSTM의 특징은 3장에서 자세히 살펴보았다. 이러한 기술을 활용하여 음절/어절 단위의 정의 문장을 생성하는 첫 번째 모델과 두 번째, 특징 및 자질 추출에 유용한 CNN 음절 임베딩을 기반으로 한 LSTM 어절 단위 정의 문장을 생성하는 모델에 대하여 자세히 살펴본다.

4.1 기술 용어에 대한 한국어 정의 문장 생성 모델의 전체 구성

연구에서 구현된 두 가지 모델의 전체적인 문장 생성 과정은 〈그림 2〉와 같다. 모델은 크게 3가지 과정과 5가지 세부 모듈로 구성되어 있다. 먼저 수집 데이터를 시스템에서 학습할 수 있도록 처리하는 과정(Data Processer), 실질적인 학습 및 생성이 이루어지는 생성 과정(Machine Learning Generation Model), 생성 모델의 생성을 돕는 생성 처리 과정(Generation Processer)이 존재한다.

5가지 세부 모듈 중 문서 관리자(Document Manager)에서는 학습, 검증, 테스트 집합으로 이루어진 세 가지 학습 데이터를 읽고 음절 혹은 어절 단위 형태로 변환하는 전처리 과정을 수행한다. 문서 관리자에 입력되는 학습 데이터 값은 〈표 1〉과 같다.

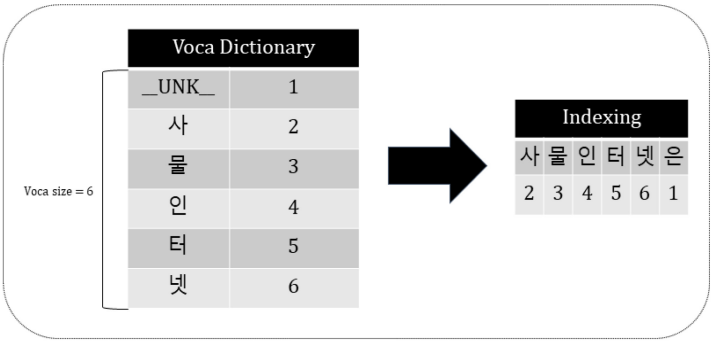
색인 관리자(Index Manager)는 음절 및 어절 단위로 변환된 학습 데이터에 대한 어휘 사전을 구성한다. 어휘 사전을 기반으로 학습 집합을 수치화하는 과정인 인덱싱과, 인덱싱 된 학습 집합을 적절한 크기에 따라 분리해 주는 작업을 진행한다. 어휘 사전은 지정된 단어 수에 맞게 전체 학습 집합에서 빈도수가 높은 음절 혹은 어절 추출을 통해 구성된다. 어휘 사전 수는 시스템에서 지정 가능한 파라미터 값이다. 인덱싱 과정은 〈그림 3〉과 같다. 〈그림 3〉은 음



〈그림 2〉 기술 용어에 대한 한국어 정의 문장 생성 모델 전체 구성도

〈표 1〉 기술 용어에 대한 한국어 정의문 생성 모델의 입력 값 예시

사물인터넷은 인간이 정보를 수집하고 지시하는 것이 아닌 사물 스스로가 정보를 주고 받으며 해석 및 실행하는 M2M(Machine to Machine)시스템으로 미래 초연결 사회의 기본이 되는 기술이다. /r/n



〈그림 3〉 색인 관리자(Index Manager)의 어휘 사전을 이용한 인덱싱 과정

절 단위의 어휘 사전 예시이다. 어절 단위 인덱싱 또한 사전의 단위가 어절일 뿐 동일한 과정을 거친다.

트레이닝 모델(Training Model)과 생성 모델(Generation Model)은 4.2와 4.3에서 자세히 살펴본다. 트레이닝 모델은 색인 관리자에 의해 구성된 학습 집합을 실제로 학습하는 과정을 수행하며 학습된 모델을 통해 새로운 데이터를 생성하는 역할은 생성 모델에서 진행된다.

생성 관리자(Generation Manager)의 경우 생성 모델의 출력 값을 입력 받아 생성 방법론에 따라 구성된 여러 메소드를 통해 실제 기술 문장을 생성해 주는 역할을 한다.

본 연구에서 고안된 생성 방법론은 다음과 같다. 먼저 시드(Seed)를 어떤 형식으로 입력할 것인지에 대한 방법론이다. 시드는 기술 문장 생성을 위해 문장의 시작과 중간 등에 인위적으로 넣어 줄 음절 및 어절을 뜻한다. 시드의 입력 방안으로는 첫 번째 시드, 중간 시드, 마지막 시드를 넣는 총 3가지 방법을 고안하였다. 첫 번째 시드는 기술 명칭, 중간 시드와 마지막 시드는 기술 정의문에 반드시 들어가야 할 음절 및 어절로 구성하였다. 그에 따른 예시는 〈표 2〉와 같다.

〈표 2〉 기술 문장 생성을 위한 시드(Seed) 입력 방법론 예시

정의문: AMI 시스템은 언제 어디서나 자유롭게 이용할 수 있게 하는 정보 서비스 인프라이다. Seed: AMI 시스템 Inter seed: 자유롭게 Last seed: 인프라

중간 시드를 넣는다는 것은 어느 정도 파악된 규칙에 기반을 두어 정의 문장을 생성한다는 의미이다. 그러나 정의문 중간에 추가적인 시드를 입력하는 방법론은 생성된 문장이 의미적으로 일괄된 형태를 구성하지 못하는 결과를 보였다. 이에 따라 중간, 마지막 시드 입력 방법론을 제거하였다.

두 번째로는 생성 모델에서 다음 음절 및 어절을 예측할 때 예측 값에 대한 확률을 구하는 방법론에 대한 것이다. 다음 단어를 예측할 경우 예측 확률에 소프트 맥스 함수 확률 값 중 가장 큰 값을 사용할 것인지, 최대 확률 값에 기반을 둔 임의 확률 값을 사용할 것인지에 대한

두 가지 방법론이다. 일반적인 소프트 맥스의 확률 값은 다음 수식 (7)과 같다. 소프트 맥스의 $\sigma(z)_j$의 확률 값들은 범위 [0, 1] 사이에 존재하고 확률 값들을 모두 더하면 1이 된다.

$$\sigma(z)_j = \frac{e^{zj}}{\sum_{k=1}^{K} e^{zk}} \quad (j = 1,...,K) \qquad (7)$$

소프트 맥스 최대 확률 값에 기반 하여 임의 확률 값을 구하는 방법론을 자세히 살펴보면 아래 수식과 같다.

$$\sigma(z)_j = [0.1,0.5,0.3,0.0,0.1\cdots] \qquad (8)$$

$$y = random_i(\sigma(z)_j) = 1 \ , \ i = 1 \qquad (9)$$

예를 들어 현재 입력 값 z의 소프트 맥스 확률 값이 다음 수식 (8)과 같고 그 확률 값에 기반 하여 임의로 y를 예측할 경우 확률 값이 가장 높은 인덱스 1번이 예측 값으로 선정될 확률이 높아지는 것을 의미한다. 이때 j는 어휘 사전의 크기와 같다. 단순히 소프트 맥스의 높은 확률 값에 따른다면 1번이 예측되는 것이고 임의의 값을 따른 다면 1번이 예측될 확률이 높아지는 것이다. 임의의 값을 따르는 것은 수치에 대한 정규화 과정으로 볼 수 있다.

세 번째는 문장 생성 시 생성 과정을 얼마나 반복할지에 대한 방법론이다. 음절 및 어절을 각 시스템에서 지정된 파라미터 수에 따라 생성할 것인지 〈EOS〉(End of Sentence) 심볼이 나올 때까지 그 과정을 반복할 것인지를 의미한다. 본 시스템에서의 〈EOS〉 심볼은 학습 데이터 문장의 마지막에 위치하는 '.' 형태의 마침표가 된다.

〈표 3〉과 같이 문장의 길이를 지정하는 방법론은 종결 어미로 끝나는 완벽한 형태의 문장을 생성하지 못한다. 본 연구의 목표는 하나의 기술 정의 문장을 생성하는 것이므로 정의 문장 형식에 맞지 않는 문장 길이 지정 방법론을 제거하였다.

생성 관리자에서 실제 문장 생성에 사용된 방법은 〈표 4〉로 정리할 수 있다. 최종적으로 문장 생성에 사용되는 방법론은 총 2가지로 각각 Eos sample, Random eos sample이라고 명명하였다. Eos sample의 경우 문장 생성 시 음절 및 어절 단위의 시드(기술명)를 입력받고 메소드 내부에서 그 입력 값에 대한 마지막 은닉 상태와, 소프트 맥스의 최대 확률 값을 통해 지속적으로 다음 음절 및 어절을 예측하는 방법이다. 문장 생성은 〈EOS〉 심볼인 '.' 형태의 마침표가 나올 때 종료된다. 마찬가지로 Random Eos sample의 경우 문장 생성 시 음절 및 어절 단위의 시드(기술명)를 입력받고 그 입력 값에 대한 마지막 은닉 상태와, 소프트 맥스 최대 확률 값에 기반 한 임의 확률 값을 통해 지속적으로 다음 음절 및 어절을 예측하는 형태이다. 문

〈표 3〉 기술 문장 생성 과정 반복 방법론 예시

Sentence length 50: 입도 분석기(Particle Size Analyzer)는 분말 시료나 Eos: 입도 분석기(Particle Size Analyzer)는 분말 시료나 현탁액에 포함된 입자의 크기 분포를 측정하는 장치이다.

〈표 4〉 기술 정의 문장 생성 시 사용되는 방법론

문장 생성 시 사용되는 방법론	
(1) 시드(기술명) 입력 (2) 소프트 맥스의 최대 확률 값을 통해 다음 음절 및 어절을 예측 (3) 소프트 맥스 최대 확률 값에 기반 한 임의의 확률 값을 통해 다음 음절 및 어절을 예측 (4) 〈EOS〉 심볼이 나올 때 까지 음절 및 어절을 출력	
문장 생성 방법론 명칭	문장 생성 방법론을 적용한 실제 생성 과정
Eos sample	(1) -〉 [(2) -〉 (4)] 시드(기술명) 입력 -〉 소프트 맥스의 최대 확률 값을 통해 다음 음절 및 어절을 예측 -〉 〈EOS〉 심볼이 나올 때 까지 출력
Random eos sample	(1) -〉 [(3) -〉 (4)] 시드(기술명) 입력 -〉 소프트 맥스 최대 확률 값에 기반 한 임의의 확률 값을 통해 다음 음절 및 어절을 예측 -〉 〈EOS〉 심볼이 나올 때 까지 출력

〈표 5〉 생성 관리자(Generation Manager)의 생성 문장 및 기타 정보 출력 예시

```
Learning (1) epoch Learning rate (2)
(3) Seed:
(4) Sample:
Epoch (1) Train_loss (5), Perplexity (6)
Epoch (1) Valid_loss (7), Perplexity (8)
…
Test Perplexity: (9)
```

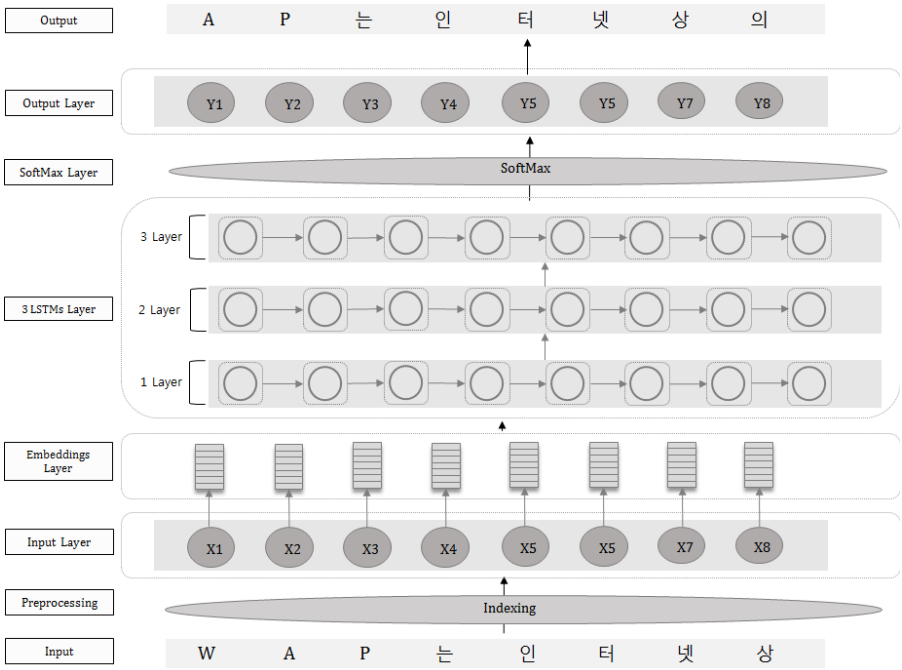
장 생성을 마무리하는 방법은 Eos sample과 동일하다.

생성 관리자에서는 생성 메소드에 따라 생성된 문장과 기타 정보를 실시간으로 확인할 수 있도록 생성 문장, 파라미터 정보, 성능 값들을 파일로 출력하는 작업을 수행한다. 〈표 5〉의 (1)의 경우 학습이 몇 번 진행되었는지 확인할 수 있는 에포크 수이고, (2)는 데이터를 얼마나 학습할지에 대한 학습률을 의미한다. (3)은 정의문으로 생성될 기술명이고 그에 따라 생성된 기술 정의문은 (4)에 나타난다. (5), (7)은 각각 학습 및 검증 집합에 대한 손실 값이고 (6), (8)에는 언어 모델의 손실 값인 혼잡도(Perplexity)가 출력된다. 최종적으로 학습된 모델의 테스트 집합 혼잡도는 (9)에서 확인할 수 있다.

4.2 LSTM 음절/어절 단위 기술 정의 문장 생성 모델(CHAR/WORD-LSTM)

LSTM 기반의 음절/어절 단위 기술 정의 문장 생성 모델은 음절 및 어절 단위로 일반 및 기술 정의문을 순차적으로 입력받아 입력 값인 현재 어휘와, 정답 값인 다음 어휘를 함께 학습하는 구조이다. 학습이 끝난 모델은 입력된 음절 및 어절에 대한 예측 값들을 이어 붙여 최종적으로 하나의 문장 형태의 결과 값을 출력한다. 〈그림 4〉는 음절 단위에 대한 예시이다.

모델의 입력 값은 〈그림 4〉에서와 같은 전처리 과정을 거친다. 전처리 과정은 4.1에서 자세히 설명하였으며 음절 및 어절 단위에 따라 데이터



〈그림 4〉 LSTM 음절/어절 단위 기술 정의 문장 생성 모델

를 구성하고 각 단위 데이터는 어휘 사전에 기반하여 인덱싱 되는 과정을 의미한다. 인덱싱 된 입력 값은 입력 계층과 임베딩 계층(Embedding Layers)을 거쳐 어휘 벡터(Vocabulary Vector)를 구성한다. 어휘에 대한 벡터 크기는 $[x_i \times v_{size}]$이다. 여기서 x는 입력된 음절 및 어절 값, v는 어휘 사전을 의미한다. 어휘 벡터는 데이터가 학습됨에 따라 계속 변화되고 점점 어휘 간의 유의미한 관계 표현 값을 지니게 된다.

임베딩 된 입력 값은 LSTM 계층으로 입력된다. 은닉 상태의 크기는 모델 구성 시 지정한 파라미터 값이다. 지정된 깊이를 지닌 은닉 상태 h_t는 이전 시간 단계의 은닉 상태 h_{t-1}과 그에 대한 가중치 W_h, 입력 값 x_t와 그에 대한 가중치 W_x, 바이어스 b_x에 의해 계산된다. 전체적인 과정은 수식 (10)과 같으며 g는 비선형 활성화 함수이다.

$$h_t = g(x_t W_x + h_{t-1} W_h + b_x) \quad (10)$$

다음과 같이 구성된 은닉 상태를 이어 하나의 LSTM 계층을 구성하고 이러한 LSTM 계층을 차례대로 여러 개 쌓아 계층 간의 은닉 상태를 전달한다. 이때 LSTM의 출력 크기는 $[x_i \times h_{size}]$가 된다.

LSTM 출력 값은 선형 계층을 통해 다시 $[x_i \times v_{size}]$ 형태가 된다. 선형 계층을 거치게 되면 데이터 x_i에 대해 어휘 사전 v_{size}에 존재하는 각 어휘에 대한 예측 확률 값을 얻어 낼 수 있다. 즉, 현재 입력 값 x_t 다음 어떤 음절 및 어절이 올지에 대한 예측 값을 측정하는 것이다. 이러한 예측 확률 값을 바탕으로 가

장 높은 확률 값을 지니는 값과, 그 높은 확률 값에 기반 한 임의의 값을 이용하여 최종적인 문장을 생성한다. 수식 (11)은 입력 값 x_t에 대해 어휘 사전에 존재하는 각 어휘에 대한 확률 값 중 가장 큰 값을 통해 x_{t+1}을 예측한다는 의미이다.

$$y_t = p(x_{t+1}|\max(p(v_{x_t})) \quad (11)$$

$$S = [y_1, y_2, \ldots, y_t] \quad (12)$$

결국 최종 문장 S는 수식 (12)와 같이 입력 값 x_t에 대한 예측 값 y_t를 여러 개 나열한 값이 된다. 이 단계에서의 최종 문장은 숫자로 이루어진 인덱스 값이기 때문에 어휘 사전을 통하여 다시 음절 및 어절로 변환되는 과정을 거친다.

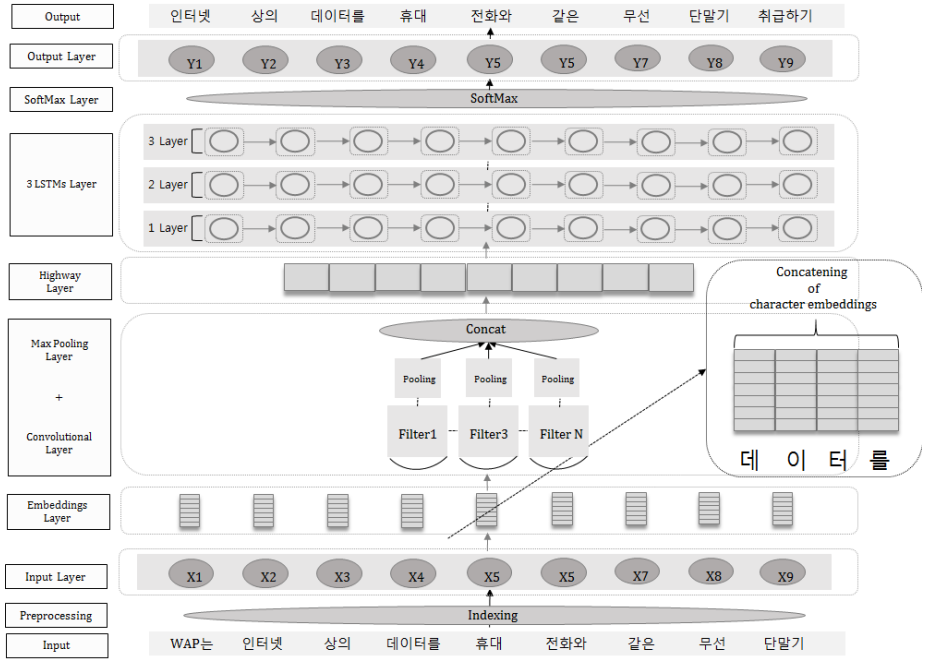
4.3 CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델 (CHAR-CNN-LSTM)

두 번째 모델은 CNN 음절 임베딩을 거친 입력 값을 통해 LSTM 어절 단위 기술 정의 문장을 생성하는 모델이다. 첫 번째 모델과의 차이점은 LSTM 계층 이전 컨볼루션 계층과 하이웨이 계층이 존재한다는 점이다. 컨볼루션 계층은 앞서 언급했던 것과 같이 데이터 특징 및 자질 추출에 유용한 네트워크이다. 이러한 컨볼루션 계층과 LSTM 계층을 정보의 손실 없이 효율적으로 이어줄 수 있는 계층이 하이웨이 네트워크(Srivastava, Greff and Schmidhuber 2015)이다. 또한 첫 번째 모델은 입력 값의 단위에 따라 음절과 어절 두 어휘에 대한 생성 값을 낼 수 있지만 두 번째 모델은 음절 단위의 문장을 입력받아 최종적으로 어절 단위의 출력 값을 낸다는 것에 그 차이가 있다.

두 번째 모델의 입력 데이터는 앞서 설명한 모델과 동일한 인덱싱 단계를 거친다. 이 단계에서 첫 번째 모델과는 다르게 음절과 어절 두 가지 어휘 사전이 모두 구성된다. 두 개의 어휘 사전을 바탕으로 각각 음절 인덱싱, 어절 인덱싱이 이루어진다. 단어가 해당되는 숫자로 단순 변환되는 어절 인덱싱과는 달리 음절 인덱싱은 어절의 최대 길이에 따라 이루어져야 한다. 예를 들어 학습 집합에 존재하는 어절의 최대 길이가 5일 경우 그 최대 길이에 맞춰 나머지 음절은 패딩 값으로 채워지는 형태이다.

이러한 음절 입력 값은 임베딩 계층을 거쳐 음절 벡터를 구성한다. 어절에 대한 각 음절 벡터는 〈그림 5〉에서와 같이 서로 연결(Concatenate)되어 하나의 어절 임베딩 벡터를 구성한다. 여러 음절 단위의 임베딩이 합쳐진 어절 임베딩 벡터의 크기는 $[x_i \times char_{\max} \times E_{size}]$가 된다. x는 입력된 어절 값, $char_{\max}$는 어절의 최대 길이, E_{size}는 음절 임베딩 차원 수이다. 음절 임베딩의 차원 수는 모델 구성 시 지정한 파라미터 값이다.

위와 같이 음절 임베딩을 통해 구성된 어절 벡터는 컨볼루션 계층의 입력 값이 된다. 컨볼루션 계층의 필터의 깊이와 넓이인 윈도우 사이즈(Window Size)는 모델 구성 시 지정한 파라미터 값이다. 만약 지정된 n개의 필터의 크기를 $w_n = [2,3,4,5,6,7\cdots]$로, 필터의 깊이를 $h_n = [50,50,100,100,100\cdots]$로 각각 지정했다면 반드시 윈도우 사이즈의 개수와 필터 깊이의 개수가 동일한지 확인해야 한다. 또한 각 필터의 크기는 $[\min\{100, 50\times w_n\}]$이 된다. 크기와 깊이가 지정된 n개의 필터는 입력된 어절 임베



〈그림 5〉 CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

딩 벡터를 필터 별로 컨볼루션 하게 된다. 컨볼루션 된 임베딩 벡터는 맥스 풀링 계층을 거치는데, 맥스 풀링 계층까지 거친 i개의 어절 임베딩 벡터는 하나로 연결되어 $[x_i \times \sum_{n=1}^{N} w_n h_n]$ 형태가 된다. x는 입력된 어절, $\sum_{n=1}^{N} w_n h_n$은 각 필터 크기의 전체 합을 의미한다. 컨볼루션 계층의 출력 값은 다시 하이웨이 네트워크의 입력 값이 된다. 하이웨이 계층은 각 필터 별로 가지고 있는 데이터의 지역적인 특징들이 하나로 합쳐지는 과정이다. 하이웨이 계층의 출력 값은 LSTM 계층에 들어가기 전 $[x_i \times 1 \times h_{size}]$ 형태로 변형된다. h_{size}는 LSTM 은닉 상태의 크기로 모델 구성 시 지정한 파라미터 값이다. 두 번째 모델은 첫 번째 모델에서 설명한 LSTM 계층의 예측 방법론과 동일한 형태로 문장을 생성한다.

5. 실 험

5.1 실험 데이터 및 실험 방법

본 논문에서 구현한 언어 생성 모델의 성능 최적화를 위해 학습 데이터에 기반 한 파라미터 실험을 진행하였다. 한국어 언어 모델의 경우 생성 모델을 평가할 수 있는 직접적인 데이터 및 평가 셋이 존재하지 않는다. 그렇기 때문에 자체적으로 수집한 한국어 기술 데이터를 사용하여 학습, 검증, 테스트 집합을 구성하고 모델의 학습 및 평가를 진행하였다. 한국어 범용 및 기술 분야 용어 자원 수집 현황은 〈표 6〉과 같다. 수집된 한국어 기반의 기술 분야 용어 자원만을 모델 최적화 학습 데이터로 사용하기에는 그 양이 적어 상대적으로 쉽게 수집 가능한 범

용 분야의 한국어 자원(Kowiki 2017)을 추가로 수집하였다. 언어 생성 모델에서 손실 값을 통해 성능을 측정하는 방법인 혼잡도 측정과 기술 정의 문장을 직접 확인하고 그에 대한 의미적 타당성을 평가할 수 있는 정성 평가를 진행하였다. 의미적인 타당성 평가를 위해서 수집된 학습 데이터를 기반으로 추가 평가 셋을 구성하였고 이를 통해 평가를 진행하였다.

〈표 6〉 수집된 한국어 범용 및 기술 분야 용어 자원 수집 건 수

데이터 분야	자원명	수집 건 수
범용 분야	Kowiki Abstract (Kowiki 2017)	382,672
기술 분야	KMR 기술 보고서	399
	과학기술분야 논문초록	10,114
	IT/기술 용어 사전	39,467
	기타 IT/기술 분야 정의문	34,536

개발된 모델의 성능 평가를 위해 사용된 학습 데이터의 크기는 〈표 7〉과 같다. 데이터는 2가지 종류로 첫 번째 데이터의 경우 범용적인 일반 정의문과 기술 정의문 및 설명문을 포함한 가공 데이터이다. 두 번째 데이터의 경우 범용 정의문을 제외한 기술 정의문 및 설명문에 대한 데이터이다.

〈표 7〉 수집 및 가공된 범용 및 기술 분야 데이터 세부 사항

데이터 명	데이터 크기	
범용 정의문을 포함한 기술 정의문 데이터	문장 수	300,000
	단어 수	4,239,113
범용 정의문을 불 포함한 기술 정의문 데이터	문장 수	201,102
	단어 수	3,232,988

언어 모델의 성능을 평가하는 가장 기본적인 방법은 손실 값을 활용한 혼잡도 측정 방법이다. 혼잡도는 주어진 음절 및 어절 다음에 오는 단어를 얼마나 잘못 예측했는지 그 평균값을 측정하는 것이다. 혼잡도를 구하는 방식은 수식 (13)과 같다.

$$\exp(\sum_{x} p(x)\log_e \frac{1}{p(x)}) \quad (13)$$

혼잡도 측정은 시스템 자체에 대해 객관적 평가가 가능하지만 생성 문장에 대한 의미적 타당성 평가는 불가능하다. 실제 입력된 기술명에 대해 본 모델들이 정의문을 의미적으로 유의미하게 생성했는지를 판단하기 위해서는 또 다른 평가 방법이 필요하다. 이에 따라 기술 정의 문장의 의미적인 적절성을 평가할 수 있는 추가 평가 셋을 구축하였다. 평가 셋은 총 8가지로 학습 데이터에 존재하는 기술명과 그에 대한 정의문 쌍으로 구성되어 있다. 이에 따른 평가 셋은 〈표 8〉과 같다.

본 논문에서는 기술명 사물인터넷, 증강현실, 라이파이, 지능형 교통 시스템에 대한 생성문장 결과를 싣고 그 결과를 세부적으로 분석하였다. 나머지 4가지 기술명은 상대적으로 학습 데이터에 기술명칭에 대한 정의문 및 설명문의 등장횟수가 적어 이에 대한 적절한 학습이 이루어지지 못해 본 논문의 생성문장 결과 분석 과정에서 제외되었다. 이를 기반으로 실제 생성된 기술 문장이 그 정의와 의미적으로 유사한지를 직접 대조하는 방식으로 성능을 평가하였다.

〈표 8〉 기술 정의 문장 생성 모델의 의미적 타당성 측정을 위한 평가 셋

개념	정의문
사물인터넷	**사물인터넷**은 인위적인 지시가 없어도 **사물**에 내장된 센서를 통해 수집한 데이터를 사용자 간에 교환하고, 이 데이터를 기반으로 2차적인 지능적 서비스를 수행한다.
	사물인터넷은 인간이 정보를 수집하고 지시하는 것이 아닌 **사물** 스스로가 정보를 주고받으며 해석 및 실행하는 M2M(Machine to Machine)시스템으로 미래 초연결 사회의 기본이 되는 기술이다.
증강현실	**증강현실** 가상 현실(Virtual Reality)의 한 분야로 실제 환경에 **가상** 사물이나 정보를 합성하여 원래의 환경에 존재하는 사물처럼 보이도록 하는 컴퓨터 그래픽 기법이다.
	증강현실 사용자가 눈으로 보는 현실 세계에 **가상** 물체를 겹쳐 보여주는 기술.
라이파이	**라이파이** 발광 다이오드(LED)에서 나오는 빛의 파장을 이용하여 정보를 전달하는 가시광 통신(VLC: Visible Light Communication) 기술의 보조 방식.
	라이파이 라이파이(Li-Fi)는 조명이 있는 곳이면 어디서나 사용할 수 있으며 인체에 무해하고 짧은 도달 거리, 저비용, 고속 통신, 안정성, 보안성 등 다양한 장점을 갖고 있으며 허가 불필요 대역으로 주파수 사용 대가가 무료다.
지능형 교통 시스템	**지능형 교통 시스템**(ITS)은 기존 **교통체계**를 기반으로 **자동차·도로** 분야에 정보통신 및 제어 등의 지능형 기술을 접목하여 교통운영·관리의 효율성을 높이고 이용자의 편의와 안정성을 제공하는 미래형 **교통 체계**이다.
	지능형 교통 시스템 교통·전자·통신·제어 등 첨단기술을 **교통·차량**·화물 등 교통 체계의 구성 요소에 적용하여 실시간 **교통 정보를** 수집 및 관리, 제공함으로써 **교통** 시설의 이용 효율을 극대화하고, **교통** 이용 편의와 **교통**안전을 높이며, 에너지를 절감하게 한다.
비콘	**비콘**은 블루투스 통신 기술을 활용해 **근거리** 내에 감지되는 스마트 기기에 정보를 전송할 수 있는 **무선 통신** 장치로서, 최근에는 애플 아이비콘(iBeacon)처럼 배터리 소모가 적은 저전력 **블루투스**(BLE, Bluetooth Low Energy) 기반의 비콘이 주류로 부상하고 있다.
착용 기술	정보통신(ICT) 기기를 사용자 손목, 팔, 머리 등 몸에 지니고 다닐 수 있는 기기로 만드는 기술.
에이디에스비	ADS-B 시스템은 **항공기의 감시 정보**(항공기 식별 부호, 위치, 속도, 방향 등)를 1초 단위로 지상의 ATC(Air Traffic Control) 시스템과 다른 항공기에 방송(broadcast)한다.
위성 기반 보정 시스템	GNSS의 위치 오차를 보정한 정보를 위성을 통해 사용자에게 전달하는 광역(wide-area)의 위성 항법 보정 시스템.

5.2 실험 결과

5.2.1 최적화 함수(Optimizer), 드롭아웃(Dropout), LSTM 계층 수 별 혼잡도 측정

LSTM을 활용한 음절/어절 단위 기술 정의문 생성 모델 중 음절 단위에 대해 파라미터 별 혼잡도 실험을 진행하였다. 혼잡도 측정에 사용된 파라미터는 각각 최적화 함수, 드롭아웃, LSTM 계층 수이다. 실험에 사용되는 파라미터 세 가지를 제외한 나머지 파라미터들은 동일하게 지정하였으며 사용된 전체 파라미터는 다음 〈표 9〉와 같다. 파라미터 실험은 300,000 문장의 '범용 정의문을 포함한 기술 정의문 데이터'로 진행되었다.

최적화 함수, 드롭아웃, LSTM 계층 수에 따른 파라미터 별 혼잡도 측정 결과는 〈표 10〉과 같다. 최적화 함수의 경우 일반적인 GradientDescent 함수를, 드롭아웃은 20을 적용했을 경우 가장 높은 성능을 보였다. LSTM을 5계층 사용할 시

〈표 9〉 LSTM 음절 단위 기술 정의문 생성 모델에서 사용된 파라미터 범위 값

파라미터명	범위 값
Optimizer	[AdadeltaOptimizer, AdamOptimizer, GradientDescent]
Init_scale	0.1
Learning rate	1.0
Max_gradient_norm	5
Num_layer	[3, 5]
Batch_size	24
Num_steps	70
Hidden_size	1024
Max_lr_epoch	24
Epoch	255
Drop Out	[0.6, 0.8, 1.0]
Lr_decay	1 / 1.15
Voca_size	2500
Patience	15

〈표 10〉 LSTM 음절 단위 기술 정의문 생성 모델 파라미터 별 혼잡도 측정 결과

학습 데이터: 범용 정의문을 포함한 기술 정의문 데이터			
최적화 함수	**AdadeltaOptimizer**	**AdamOptimizer**	**GradientDescent**
	8.745	8.314	8.214
드롭 아웃	**50**	**80**	**100**
	6.749	6.723	8.214
LSTM 계층 수	**3**	**5**	
	8.214	151.896	

3계층을 사용한 경우에 비해 상대적으로 혼잡도가 매우 높은 것을 볼 수 있다. 이는 더 깊은 계층으로 인해 입력 값과 예측 값이 제대로 전달되지 않은 경우로 판단된다. 또한 실험 중 적용된 얼리 스타핑(Early Stopping) 함수의 결과 또한 영향을 미친 것으로 보인다. 얼리 스타핑은 단일 에포크 후 검증 집합의 혼잡도를 측정할 시 검증 집합의 혼잡도가 파라미터 Patience 수만큼 개선되지 않을 경우 학습을 중지하는 방법이다. 상대적으로 더 많은 계층을 가진 모델은 적은 계층의 모델보다 학습 최적화가 느리다. 높은 혼잡도는 얼리 스타핑이 적용되어 내부 파라미터가 최적화되지 못한 채 학습이 끝난 모습으로 추측할 수 있다.

또한 LSTM 음절/어절 단위 기술 정의문 생성 모델 중 어절 단위와 CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델의 혼잡도를 비교하였다. 혼잡도 실험은 200,000 문장의 '범용 정의문을 불 포함한 기술 정의문 데이터'로 진행되었다. 혼잡도 측정 결과는 〈표 11〉과 같다.

실험결과 WORD-LSTM 모델의 경우 CHAR-CNN-LSTM 모델보다 높은 혼잡도를 보였다. 그리고 앞서 설명한 음절 단위 시스템에 비해

〈표 11〉 생성 시스템 별 혼잡도 측정 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터	
WORD-LSTM	CHAR-CNN-LSTM
94.853	94.234

* WORD-LSTM: LSTM 어절 단위 기술 정의 문장 생성 모델
* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

혼잡도가 매우 높은 것을 확인할 수 있다. 이는 어휘 사전 크기와 관련이 있다. 어휘 변형으로 인한 경우의 수가 음절보다 어절이 더욱 많이 존재하기 때문에 사전 크기를 크게 잡아야 하고 이는 손실 값 측정 시 예측 값의 범위를 증가시켜 더 큰 혼잡도를 발생시켰다.

5.2.2 생성된 기술 정의문에 대한 의미적 타당성 평가

5.2.2의 아래 표들은 개발된 모델을 통해 생성된 정의 문장에 대한 결과이다. 문장 생성 시 최대 확률 값에 대한 생성 방법론과 최대 확률 값을 기반으로 한 임의 생성 방법론을 혼용하였으며, 기술 명칭을 입력할 시 이에 따라 예측되는 값들을 통해 문장을 생성해 내는 방식이다. 또한 정의문 형태의 문장을 생성해야하기 때문에 마침표를 EOS 심볼로 지정하였고, EOS 심볼이 나올 때까지 반복적으로 문장을 생성하였다. 〈그림 6〉은 앞서 설명한 LSTM 계층의 문장 생성 과정을 도식화한 예시이다. 그림에서 Method1과 2는 각각 최대 확률 값을 통한 생성 방법론과 최대 확률 값을 기반으로 한 임의

Input = "사물인터넷"
Preprocessing1 = [x_1="사", x_2="물", x_3="인", x_4="터", x_5="넷"]
Preprocessing2 = [x_1="1", x_2="2", x_3="3", x_4="4", x_5="5"]
Generation_Method1:
  $O_t(t=6) = \arg max P(x_6|x_1 \ldots x_5) = 6$, $index6$ = "("
  while :
    $input_t$ = [x_t ="사물인터넷(IOT)… 기술이다"]
    $O_t(t=i) = \arg max P(x_t|x_1 \cdots x_{t-1}) = 0$, $index0$ = "."
    $if(O_t == 0)$:
      print ("사물인터넷(IOT)… 기술이다.")
      break
Generation_Method2:
  $O_t(t=6) = random(\arg max(P(x_6|x_1 \ldots x_5)) = 7$, $index7$ = "은"
  while :
    $input_t$ = [x_t ="사물인터넷은… 기술이다"]
    $O_t(t=i) = random(\arg max P(x_t|x_1 \ldots x_{t-1})) = 0$, $index0$ = "."
    $if(O_t == 0)$:
      print ("사물인터넷은… 기술이다.")
      break

〈그림 6〉 LSTM 계층에서 생성되는 정의문장 과정 예시

생성 방법론을 의미한다.

〈표 12〉는 두 가지 모델에서 생성된 기술 정의 문장이다. 기술명으로 '사물인터넷'을 입력하고 그에 따라 생성된 문장 결과를 평가 셋에 기반 하여 의미적으로 타당한 문장들을 필터링 하였다. 가장 먼저 두 모델에서 1번 문장은 사물인터넷의 정의를 의미론적으로 가장 잘 생성한 문장으로 평가하였다. 〈표 8〉 평가 셋에 존재하는 정의 문장에 대한 핵심 키워드인 '사물'이 포함되어 있으며 각각 '사물을 인식', '사물을 실시간 제공' 등의 문장을 통해 의미적으로도 사물인터넷의 정의를 유사하게 생성했음을 확인할 수 있다. 또한 생성된 정의문에는 기술명의 영문명 및 약어들이 함께 생성되었다.

사물인터넷이라는 기술명과 그에 대한 영문명 'Internet of Things' 그리고 약어 'IOT', 'iot'와 표에는 존재하지 않지만 약어의 혼용인 'IoT: Internet of Things' 또한 생성되었다. 이러한 약어 및 영문명은 CHAR-LSTM 모델에서 상대적으로 더 잘 생성되는 것을 확인할 수 있었다. 약어 및 영문명은 학습 데이터 내부에서 기술명과 멀지 않은 거리에 존재하며 이는 다음 음절 및 어절을 레이블로 사용하는 LSTM 메커니즘을 잘 반영한 결과로 보인다. 〈표 12〉에서 소문자화 된 영문과 문장 내 괄호 띄어쓰기는 CHAR-CNN-LSTM 모델의 전처리 과정에 따른 결과이다.

〈표 13〉은 기술명 '증강현실'에 대한 정의 문장 생성 결과이다. 두 모델의 1번 문장의 경우 '증강현실 기술을 이용하여', '사용하여'라는 단어들로 시작된다. 또한 문장 내에 가상의 공간을 구성한다는 의미를 담고 있다. 두 모델의 2번 문장의 경우 '가상'이라는 키워드를 포함하고 있으며 3번 문장은 가상현실과 유사한 환경임을 의미론적으로 표현하고 있다.

〈표 14〉는 '라이파이'에 대한 정의 문장 생성 결과이다. CHAR-CNN-LSTM 모델의 1번 문장의 경우 라이파이 의미를 담은 기술 문장을 생성한 것처럼 보인다. 또한 CHAR-LSTM 모델의 1번 문장의 경우 라이파이 정의문에 존재하는 '발광 다이오드(LED)' 키워드가 포함된 모습을 보여준다. 이는 앞서 설명한 기술명에 대한 영문명 및 약어 생성 과정과 유사하다. 또한 CHAR-LSTM 모델의 2번 문장은 라이파이의 영문명인 'Li-Fi'를 생성했다. 그러나 전체적으로 라이파이의 영문명으로 'Wi-Fi'가 생

〈표 12〉 기술명 '사물인터넷'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터		
	CHAR-LSTM	CHAR-CNN-LSTM
1	사물인터넷(IoT) 기술, 방송네트워크, 그리고 **사물을 인식**하는 핵심 기술로 주목받고 있다.	사물인터넷(iot)을 통해 각종 정보와 환경 활동 **사물을 실시간으로 제공**하는 웹 기반의 홈 환경.
2	사물인터넷(IoT)을 적용한 TV 서비스를 제공할 수 있는 시스템이다.	사물인터넷(iot)을 통해 실시간으로 음성과 정보를 얻을 수 있게 하는 기술.
3	사물인터넷(Internet of Things)컴퓨터가 대표적인 예이다.	사물인터넷(iot)은 크게 음성・영상・웹 관련 프로그램 등 **다양한 형태로** 제공된다.

* CHAR-LSTM: LSTM 음절 단위 기술 정의 문장 생성 모델
* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

〈표 13〉 기술명 '증강현실'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터		
	CHAR-LSTM	CHAR-CNN-LSTM
1	**증강현실 기술을 이용하여** 다양한 가상의 공간을 구현한다.	**증강현실 기술을 사용하여** 화상을 실제 단말에 같은 제약 없이 배치하는 방법이다
2	증강현실 및 CoD protection으로서 물리적으로 떨어져 있는 모바일 기기로 바로 보이게 하는 **가상**화 기술이다.	증강현실 기술을 이용하여 **가상**의 시장 인식, **가상** 네트워크의 통신, **가상** 사설 등 다양한 분야에 복잡한 보안 서비스를 제공한다.
3	증강현실에서 우수한 가상현실 인공지능 기술을 혼합, 실시간 환경에서 원인으로 컴퓨터를 비롯하여 민간 분야와 여러 가지 특성을 갖고 있다.	증강현실 같은 가상 현실 (vr)와 가상 환경과 같은 사람이 존재하는 연결 방식에서 네트워크 기능을 이용하여 웹 서비스를 사용해 모든 프로그램을 실행할 수 있는 환경.

* CHAR-LSTM: LSTM 음절 단위 기술 정의 문장 생성 모델
* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

〈표 14〉 기술명 '라이파이'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터		
	CHAR-LSTM	CHAR-CNN-LSTM
1	라이파이 **발광 다이오드(LED)** 칩 속과 구별되는 것과 달리 장된 전극을 기준으로 반대 방향으로 바깥 위에 빛을 내는 것으로, 조명 대역폭(듐ε)은 10억㎜² 이상이 큰 회로를 사용하고 있지만, 이것들은 수컴 배에서 수천 배의 전원 범위를 지속해서 큰 지연 시간을 보이고 있다.	라이파이 및 유무선 통신망을 통해 언제 어디서나 네트워크에 접속할 수 있는 통신망.
2	라이파이(**Li-Fi** Field) 반경(WiFi 이상)의 경우 무선 디바이스	라이파이, 무선(lan)기술을 의미한다.
3	라이파이(**Wi-Fi**) 기기 내와 같이 상수 등의 신뢰도가 높은 단말기에서 해당 기기를 소지한 후에 사용할 수 있는 서비스이다.	라이파이 또는 다른 네트워크로 서비스를 제공할 수 있게 하는 기술

* CHAR-LSTM: LSTM 음절 단위 기술 정의 문장 생성 모델
* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

성되는 모습을 확인할 수 있었다. 이는 학습 데이터 내부에 존재하는 와이파이 내용의 영향으로 보인다. 학습 데이터 속 라이파이는 6개의 문장에서 총 7회 등장하였으나, 와이파이는 82개의 문장에서 총 107회 등장하였다. 이는 두 기술명에 존재하는 '파이'라는 공통 어절에 의한 결과로 보인다.

〈표 15〉는 '지능형 교통 시스템' 기술명에 대한 정의 문장 생성 결과이다. 보는 바와 같이 생성 문장은 전체적으로 지능형 교통 시스템의 기술적 의미를 잘 표현하고 있다. 첫 번째 문장의 경우 지능형 교통 시스템의 정의인 '자동차・도로'와 유사한 '차량과 도로' 키워드들을 포함하고 있으며 형식적인 부분뿐만 아니라 의미적으로도 비교적 완벽한 형태의 정의문임을 확인할 수 있다. 또한 두 번째 문장의 경우 '이용자 정보를 바탕으로 하여 각종 정보와 교통 정보를 수집, 처리하고 저장한 데이터를'라는 구절을 통하여 지능형 교통 시스템의 기능을 유사하게 설명하였다. 또한 '사용자 중심의 자동차 관리를 가능하게 한다.'라는 문장은 지능형 교통 시스템을 잘 설명한 구절이라 볼 수 있다.

본 논문에서 구현된 CHAR-LSTM 모델의 문장 생성 결과를 전반적으로 살펴볼 경우 문장의

〈표 15〉 기술명 '지능형 교통 시스템'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터	
	CHAR-LSTM
1	지능형 교통 시스템(ITS)에 내장된 센서 및 센서를 통해 **차량과 도로** 상황을 인식하는 시스템으로 관리자, 결제 시설 및 서비스 이용 시간 등 정보를 수집하여 필요 없는 지역 도로반경 약 30m 거리에서 30MHz까지 측정하고 조정하는 장치이다.
2	지능형 교통 시스템(ITS) 정보 기술을 활용하는 **이용자 정보를 바탕으로 하여 각종 정보와 교통 정보를 수집, 처리하고 저장한 데이터를** 가공기관에 리더로 활용하여 다양한 형태의 정보를 수집하고, 이를 효율적으로 관리함으로써 복합적인 도시 구조와 각종 상황을 고려해 **사용자 중심의 자동차 관리를 가능하게 한다**.
3	지능형 교통 시스템(ITS: Information Technology System)을 설계, 지침으로 제공하는 것을 목표로 하고 있다.

* CHAR-LSTM: LSTM 음절 단위 기술 정의 문장 생성 모델

형식이 자연스럽고 매끄러우며, 기술명의 영문명 및 약어 생성 부분이 상대적으로 잘 이루어지는 것을 확인할 수 있었다. 그러나 CHAR-LSTM 모델의 경우 기술 정의의 의미를 담은 문장 생성이 비교적 잘 이루어지지 않는 모습을 보였다. 한편 CHAR-CNN-LSTM 모델은 CHAR-LSTM 모델에 비해 관련 약어 및 영문명 생성이 잘 이루어지지 않았다. 이는 어절 단위로 진행된 생성 방법론의 영향으로 보인다. 어절 단위는 상대적으로 큰 단어 사전을 필요로 하고, 단어 사전 구성은 학습 데이터의 단어 빈도수를 기반으로 구성된다. 이에 따라 빈도수가 적은 영어 단어를 생성하는 것은 비교적 어렵다고 판단된다. 그러나 의미 있는 기술 정의문을 생성할 수 있을 뿐만 아니라 학습 데이터에 기반 한 변형된 기술 정의문을 생성하였다. 이에 따라 본 논문의 두 모델 중 기술 정의문 생성에 좀 더 적합한 시스템은 CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델로 판단된다.

추가적으로 〈표 16〉과 〈표 17〉은 CHAR-CNN-LSTM 모델에서 생성된 문장 결과이다. 앞서 나온 기술명보다는 좀 더 일반적으로 사용될 수 있는 용어 '데이터베이스'와 '알고리즘'에 대한 결과이다. 모든 문장에서 비교적 자주 사용되며 특정 용어에 대한 정확한 정의문이 데이터에 존재하지 않기 때문에 이에 따라 정의문보다는 문맥적, 의미적으로 평이한 문장의 형태를 보이고 있다. 이러한 결과를 통해 추가적인 데이터 수집 과정에서는 특정 기술 명칭에 대한 '정의문'을 대량으로 수집하는 것에 초점을 두어야 한다는 사실을 확인할 수 있었다.

〈표 16〉 기술명 '데이터베이스'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터	
	CHAR-CNN-LSTM
1	데이터베이스(db)에서 기억 장치를 이용하여 데이터를 전송하고 활용하는 데 사용된다.
2	데이터베이스 관리 시스템(dbms)을 통하여 각종 사용자의 데이터베이스(db)정보 등을 수집 / 제공하는 장치.
3	데이터베이스 관리 시스템(dbms)을 도입, 운영하고, 그 뒤 검색 정보를 활용할 수 있다.

* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

〈표 17〉 기술명 '알고리즘'에 대한 정의문장 생성 결과

학습 데이터: 범용 정의문을 불 포함한 기술 정의문 데이터	
	CHAR-CNN-LSTM
1	알고리즘 데이터 및 네트워크 서비스에 대한 이해를 분석하는 행위
2	알고리즘 성능 개선 활용 시 새로운 알고리즘 선택 방법을 제안한다.
3	알고리즘 특정 처리 목적을 수행하기 위해 시스템에 실행되는 데이터를 단계별로, 컴퓨터에 의한 정보 또는 그에 의한 기술적인 접근을 방지하기 위해 준비된 작성 방법을 의미한다.

* CHAR-CNN-LSTM: CNN 음절 임베딩에 기반 한 LSTM 어절 단위 기술 정의 문장 생성 모델

6. 결론 및 제언

본 논문에서는 한국어 기반의 범용 정의문과 기술 정의문 데이터를 수집, 가공하였으며 이를 바탕으로 연구를 통해 구현된 한국어 기술 정의문 생성 모델 두 가지를 세부적으로 설명하였다. 또한 정의 문장 생성을 위해 고안된 방법론을 적용한 결과인 생성 문장을 비교 분석하였으며, 모델 최적화를 위한 파라미터 실험과 생성 문장에 대한 세부 분석을 통해 문장 생성에 적합한 모델을 제안하였다.

진행된 연구를 통해 향후 연구의 발전 방향을 크게 두 가지로 나누어 보았다. 먼저 완벽한 형태의 기술 정의문 생성을 위해 기술 정의 문장으로만 이루어진 학습 데이터 구축의 필요성이다. 현재 수집된 데이터는 기술 정의 문장과 그에 대한 포괄적인 설명문이 포함된 문장들이다. 논문의 초록과 같이 기술에 대해 포괄적인 설명을 지니는 데이터는 생성 문장의 어휘를 풍부하게 하는데 도움이 될 수 있으나 문장이 특정 기술의 정의를 생성하는 것에 대해서는 큰 도움이 되지 못한다. 이에 따라 영문 기술에 대한 정의문 번역 및 한국어 기술 정의문 변형 등의 방법을 통해 대용량의 한국어 기술 정의 문장 학습 데이터를 수집할 수 있는 방안을 마련할 것이다. 두 번째로는 정의 문장 생성 시 문장 전체의 문맥 의미를 학습할 수 있는 시스템의 필요성이다. 현재는 음절 및 어절 단위로 문장을 분리하여 학습을 진행하였으나, 일반적인 자연어 생성과는 다르게 특정 주제 분야에 대해 일괄된 맥락을 가진 문장을 생성해야 하므로 이를 위해서는 문장 자체에 대한 학습이 필요해 보인다. 앞서 언급했던 변형된 오토인코더 기술은 이론적으로 문맥에 대한 광역적인 학습이 가능하다. 또한 최근 생성 모델 중 가장 높은 성능을 보이는 RHN(Recurrent Highway Networks)(Zilly 2016) 기술 등을 활용하여 추가적인 한국어 정의 문장 생성 연구를 진행할 예정이다.

본 연구는 한국어를 기반으로 특정 주제 및 의도에 맞는 정의 문장 생성 초기 연구를 진행했다는 점에 그 의의를 지닌다. 그러나 부족한 학습 데이터 및 모델 최적화로 인해 생성된 정의문이 의미적으로 완벽하게 그 기술을 서술하지 못하는 형태를 보인다. 추가적으로 국외에서 연구되고 있는 다양한 언어 모델들을 한국어에 적용하고 지속적으로 데이터를 확장시킨다면 정의문 생성에서 더 나아가 기술에 대한 분석 보고서 작성의 효율성을 확보할 수 있는 지원 시스템 개발이 가능할 것이라고 사료된다.

참 고 문 헌

[1] Bahdanau, D., Cho, K., and Bengio, Y. 2014. "Neural Machine Translation by Jointly Learning to Align and Translate." In *conference ICLR 2015*.

[2] Bauer, A., Hoedoro, N. and Schneider, A. 2015. "Rule-based Approach to Text Generation in Natural Language-Automated Text Markup Language (ATML3)." In *Challenge+ DC@ RuleML 2015*.

[3] Bian, J., Gao, B. and Liu, T. Y. 2014. "Knowledge-powered Deep Learning for Word Embedding." In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, September 15-19, 2014, Nancy: 132-148.

[4] Bontcheva, K. and Wilks, Y. 2004. "Automatic Report Generation from Ontologies: the MIAKT Approach." In *International Conference on Application of Natural Language to Information Systems*, 324-335.

[5] Boulanger-Lewandowski, N., Bengio, Y. and Vincent, P. 2012. "Modeling Temporal Dependencies in High-dimensional Sequences: Application to Polyphonic Music Generation and Transcription." In *Proceedings of the Twenty-nine International Conference on Machine Learning ICML*.

[6] Bowman, S. et al. 2016. "Generating Sentences from a Continuous Space." In *SIGNLL Conference on Computational Natural Language Learning (CONLL), 2016*.

[7] Deng, L. and Yu, D. 2014. "Deep Learning: Methods and Applications." *Foundations and Trends® in Signal Processing*, 7(3-4): 197-387.

[8] Graves, A., Jaitly, N. and Mohamed, A. R. 2013. "Hybrid Speech Recognition with Deep Bidirectional LSTM." In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, 273-278.

[9] Hochreiter, S. 1991. *Untersuchungen zu Dynamischen Neuronalen Netzen*. Ph.D. diss., Institut fur Informatik, Technische Universitat, Munchen.

[10] Hochreiter, S. and Schmidhuber, J. 1997. "Long Short-Term Memory. Neural Computation." *Neural Computation*, 9(8): 1735-1780.

[11] Kalchbrenner, N., Grefenstette, E. and Blunsom, P. 2014. "A Convolutional Neural Network for Modelling Sentences." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 655-665.

[12] Krizhevsky, A., Sutskever, I. and Hinton, G. E. 2012. "Imagenet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems*,

60(6): 1097-1105.

[13] Langkilde-Geary, I. 2002. "An Empirical Verification of Coverage and Correctness for a General-purpose Sentence Generator." In *Proceedings of the 12th International Natural Language Generation Workshop*, 17-24.

[14] Nallapati, R. et al. 2016. "Abstractive Text Summarization using Sequence-to-sequence Rnns and Beyond." In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, August 7-12, 2016, Berlin: 280-290.

[15] Mairesse, F. 2005. *Natural Language Generation: APT on Dialogue Models and Dialogue Systems*. [online] [cited 2017. 6. 30.] 〈http://farm2.user.srcf.net/research/papers/ART-NLG.pdf〉

[16] Srivastava, R. K., Greff, K. and Schmidhuber, J. 2015. "Tranining Very Deep Networks." In *Advances in Neural Information Processing Systems, (2015a)*: 2377-2385.

[17] Sundermeyer, M., Schlüter, R. and Ney, H. 2012. "LSTM Neural Networks for Language Modeling." In *Thirteenth Annual Conference of the International Speech Communication Association.*

[18] Sutskever, I., Martens, J. and Hinton, G. E. 2011. "Generating Text with Recurrent Neural Networks." In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 1017-1024.

[19] Wikipedia. 2017. *kowiki-latest-abstract.xml*. [online] [cited 2017. 6. 27.] 〈https://dumps.wikimedia.org/kowiki/latest/〉

[20] Woodward, A., Sood, B. and Hare, J. 2016. *Market Share Analysis: Business Intelligence and Analytics Software*, 2015. [online] [cited 2017. 6. 2.] 〈https://www.gartner.com/doc/3365832/market-share-analysis-business-intelligence〉

[21] Zheng, X., Chen, H. and Xu, T. 2013. "Deep Learning for Chinese Word Segmentation and POS Tagging." In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 647-657.

[22] Zilly, J. G. et al. 2016. "Recurrent Highway Networks." In *Proceedings of the 34 th International Conference on Machine Learning*.