말소리와 음성과학 제7권 제4호 (2015.12.31)
ISSN 2005-8063, pp. 3~8
ksss.jams.or.kr
http://dx.doi.org/10.13064/KSSS.2015.7.4.003

Deep Neural Network 언어모델을 위한 Continuous Word Vector 기반의 입력 차원 감소

Input Dimension Reduction based on Continuous Word Vector for Deep Neural Network Language Model

김 광 호[1] · 이 동 현[2] · 임 민 규[3] · 김 지 환[4]
Kim, Kwang-Ho · Lee, Donghyun · Lim, Minkyu · Kim, Ji-Hwan

ABSTRACT

In this paper, we investigate an input dimension reduction method using continuous word vector in deep neural network language model. In the proposed method, continuous word vectors were generated by using Google's Word2Vec from a large training corpus to satisfy distributional hypothesis. 1-of-$|V|$ coding discrete word vectors were replaced with their corresponding continuous word vectors. In our implementation, the input dimension was successfully reduced from 20,000 to 600 when a tri-gram language model is used with a vocabulary of 20,000 words. The total amount of time in training was reduced from 30 days to 14 days for Wall Street Journal training corpus (corpus length: 37M words).

Keywords: deep neural network, language model, continuous word vector, input dimension reduction

1. 서론

모바일 환경에서 사용자의 입력을 편리하게 하면서, 효율적인 접근 방법으로 음성인식 기술을 이용한 음성 검색 서비스의 사용이 증가하고 있는 추세이다. 이와 같이 음성 검색 서비스에서는 음성인식에 기반한 음성 입력 인터페이스가 사용자 경험의 핵심 역할을 담당한다. 음성인식은 사람이 일상 생활 속에서 마우스나 키보드 등을 사용하지 않고 목소리를 통해 원하는 기기 및 정보 서비스의 이용을 제어 할 수 있는 기술이다. 즉 일반적으로 마이크나 전화를 통하여 얻어진 음향학적 신호를 단어나 단어 집합 또는 문장으로 변환하는 과정을 말한다. 인식된 결과는 명령이나 제어, 데이터 입력 등의 응용 분야에서 최종 결과로 사용될 수 있으며, 음성이해와 같은 분야에는 언어 처리과정의 입력으로 사용될 수 있다.

사람이 발성한 음성신호는 수식적으로 음향학적 벡터 열 $O=(o_1,o_2,\ldots,o_T)$로 표현된다. 각 벡터는 짧은 시간의 음성구간 (약 10~20ms)에 대한 주파수 영역별 에너지로 나타낸다. 주어진 음성신호가 특정 단어 열, $W=(w_1,w_2,\ldots,w_N)$을 발성한 결과라고 했을 때 음성인식 시스템의 목표는 음성신호로부터 추출된 음향학적 벡터 열(O)에 대해서 가장 높은 확률을 가지는 단어 열($\hat{W}$)을 제시하는 것이다. 하지만 같은 사람이 같은 단어 열을 발성한다고 해도 음성신호로부터 변환되는 음향학적 벡터 열은 다르게 나올 수 있다. 이러한 이유로 발견 가능한 음향학적 벡터(O)의 수는 무한대가 된다. 이러한 음성인식의 목표를 Bayes rule를 이용하여 정리하면, 다음의 수식 (1)로 표현된다.

수식 (1)에서 $P(O)$는 음향학적 특징 벡터(O)의 발생 확률로서, W와 독립이므로 최대 확률을 가지는 단어 열을 구할 때 고려치 않아도 된다. 결국 음성인식 시스템은 $P(O|W)$와 $P(W)$의 곱이 가장 높은 단어 열을 제시하게 된다. $P(W)$는 단어 생성확률로서 언어 모델로부터 생성되며, $P(O|W)$는 특정 단어 열이 음향학적 벡터를 생성해 내는 확률로서 음향 모델로부터 생성된다. 언어모델은 음성인식뿐만 아니라, 통계적

1) 서강대학교, kimkwangho@sogang.ac.kr, 제1저자
2) 서강대학교, redizard@sogang.ac.kr, 제2저자
3) 서강대학교, lmkhi@sogang.ac.kr, 제3저자
4) 서강대학교, kimjihwan@sogang.ac.kr, 교신저자

접수일자: 2015년 8월 25일
수정일자: 2015년 9월 21일
게재결정: 2015년 11월 23일

기계 번역, 자연어 처리, 철자 오류 검증 등과 같은 다양한 애플리케이션에서 사용된다. 기존의 언어모델은 단어 기반의 *n*-gram 빈도수 정보를 이용하여 단어간의 관계를 확률 모델로 표현하였다.

$$\hat{W} = \arg max_W P(W|O) = \arg max_W \frac{P(O|W)P(W)}{P(O)} \approx \arg max_W P(O|W)P(W) \quad (1)$$

기존 *n*-gram 언어모델의 unseen word sequence 문제를 연속 공간상에서 해결하기 위해서, Deep Neural Network (DNN) 연구가 진행되었다. 언어모델링 분야에서는, Feed-Forward Neural Network (FFNN) 연구가 진행되었다. DNN 기반 언어모델에서는 연속 공간 상에서 1-of-$|V|$ coding 기반의 벡터로 표현된 단어 벡터로 사용하고 있다. 여기서 $|V|$은 어휘사전의 어휘 개수를 나타낸다. 연속 공간상으로 dimension reduction을 통해 단어 벡터로 표현하는 경우, 벡터 사이의 관계를 의미적 또는 구조적 정보를 내재하도록 할 수 있다. 학습자료에 나오지 않았던 unseen word sequence에 대한 확률을 연속 공간상에서 neural network로부터 확률 추정 함수를 통해 smoothing 해주는 효과를 얻을 수 있다. 예를 들어, 'The cat is walking in the room'이 학습 자료에 나타났고, 'A dog is running in the room' 문장이 테스트로만 입력되는 경우를 생각해 보자. 'the'와 'a'은 특정명사를 한정하는 문법적 역할을 하며, 'walking'과 'running'은 진행의 문법적 역할을 한다. 그리고 'cat'과 'dog'는 비슷한 의미적 내용을 포함하고 있다. 이를 어떤 함수로 모델링하는 경우에, 모델의 입력으로 사용되는 'the'와 'a', 'walking'과 'running', 그리고 'cat'과 'dog'에 해당하는 각 단어 벡터가 문법적 역할별로 또는 의미적 의미별로 연속 벡터 공간상에서 서로 가깝게 위치하게 된다면, 학습 자료에 나오지 않았던 unseen word sequence의 언어적 확률 모델을 표현하는 값은 seen word sequence으로 생성된 모델로부터 효과적으로 추정해 줄 수 있는 확률을 얻을 수 있다 [1][2].

DNN을 이용한 언어모델은, 단어와 같이 index만을 표현하는 이산 공간에서의 벡터가 입력으로 사용되고, 차원이 높은 벡터가 입력으로 사용되기에 DNN 적용시 계산량이 많이 요구되는 문제점이 있어, 학습에 많은 시간이 소요되었다. 결과적으로 현재는 많은 양의 학습 자료를 사용하지 못하고 있어, 많은 양의 학습자료를 이용한 *n*-gram 보다 성능이 좋지 못한 결과를 제시하였다. 이를 개선하기 위하여, 본 연구에서는 입력층의 입력의 표현을 1-of-$|V|$ coding 형태에서 연속 단어 벡터를 이용하여 입력 차원을 감소 방법을 제시하고, 성능 검증을 진행하고자 한다. 입력 차원의 감소 방법은 distributional hypothesis에 근거한 연속 벡터 공간상에서의 단어 벡터 학습 방법의 배경적 근거를 설명하고, 이를 잘 표현하는 하나의 방법인 Word2Vec을 이용하여 실험 결과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 대표적인 DNN기반 언어모델에 대해 설명한다. 3장에서는 연속 벡터 공간상에서의 단어 벡터 모델을 이용한 입력 벡터의 차원 감소 연구에 대해 기술한다. 4장에서는 연속 단어 벡터 모델 검증에 대해 설명한다. 5장에서는 본 연구의 결론에 대해 설명한다.

2. Deep Neural Network 기반 언어모델

본 장에서는 기존의 DNN 기반 언어모델에서 단어 벡터를 활용하는 대표적인 FFNN 기반 언어모델에 대해 상세히 설명한다.

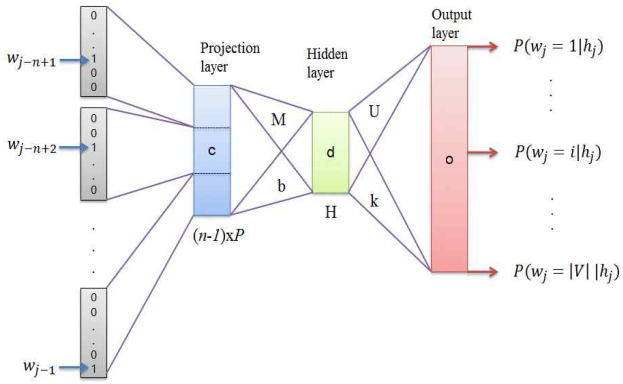

그림 1. Feed-Forward Neural Network 언어모델 구조도
Figure 1. Structure of FFNN-based language model

[1]에서는 FFNN 구조를 제시하였으며, 이를 이용하여 언어모델을 생성하고 음성인식에 접목함으로써, 기존의 *n*-gram 모델 수준의 성능을 제시하였다. FFNN의 입력과 출력을 살펴보면 다음과 같다. 입력으로는 *n*-gram의 (*n*-1) words로 구성된 단어 히스토리로 되어 있으며, 단어 히스토리의 각 단어는 1-of-$|V|$ 형태로 표현된다. V는 인식 가능한 단어 집합이고, $|V|$는 단어 집합의 개수를 나타낸다. 따라서 V내의 모든 단어들은 length $|V|$개의 값으로 표현되는 벡터 형태로 표현이 가능하다. 이 때, 단어의 index에 해당하는 값만 1로 나타나고, 다른 모든 값은 0으로 표현된다. 단어 벡터의 차원을 줄이기 위하여, projection matrix가 사용된다. 이 projection matrix 는 단어 히스토리의 시간적 위치에 상관없이, 단어에 따라 벡터로 표현되어 있다. 출력은 단어 히스토리가 주어진 경우에, 현재 단어의 확률로 표현되어 있으며, 현재 단어는 V에 있는 인식 가능한 단어들로 구성되어 있다. 이 때 FFNN 관련 수식은 다음의 식 (2), (3), (4), (5)로 표현된다. <그림 1>은 FFNN 기반 언어모델의 구조도를 보여주고 있다. 그리고 <표 1>은 FFNN 수식 기호의 정의를 나타낸다.

$$d_j = \tanh\left(\sum_{l=1}^{(n-1)\times P} M_{jl} \cdot c_l + b_j \right) \quad \forall j = 1,...,H \tag{2}$$

$$o_i = \sum_{i=1}^{H} U_{ij} \cdot d_j + k_i \quad \forall i = 1,...,|V| \tag{3}$$

$$p_i = \frac{\exp(o_i)}{\sum_{r=1}^{|V|} \exp(o_r)} = P(w_j = i|h_j) \tag{4}$$

$$E = -\sum_{i=1}^{|V|} t_i \cdot \log(p_i) + \beta(\sum_{jl} m_{jl}^2 + \sum_{ij} v_{ij}^2) \tag{5}$$

표 1. FFNN 수식의 기호 정의
Table 1. Notations in FFNN-related equations

수식 기호	정의
$\|V\|$	Vocabulary size
w_j	Current word
h_j=(w_{j-n+1},…, w_{j-2},w_{j-1})	(n-1)개의 word history
P	Feature dimension
i	i-th word in vocabulary
H	Hidden units in hidden layer
c	Linear activations in the projection layer
M	Weight matrix between the projection and hidden layer
U	Weight matrix between the hidden and output layer
d_j, b_j	j-th hidden value in hidden layer and j-th hidden bias
o_i, k_i	i-th output value in output layer and i-th output bias
E	Error function based on cross entropy

[3]에서는 대용량 학습자료를 이용한 FFNN 기반 언어모델의 학습 소요 시간 감소를 위한 연구를 진행하였다. 이를 위해, shortlist, regrouping, block mode 방식을 제시하였다. Shortlist는 출력층의 출력 어휘를 어휘 사전 내에서 학습 자료에서 자주 나오는 어휘만을 이용하는 방법으로 출력층의 크기를 줄이는 방법이다. Regrouping은 동일한 word history에 대한 현재 단어의 확률을 구할 때, 현재 단어들을 그룹으로 지정하여, 여러 번 학습되는 것을 한 번 학습으로 처리하는 방법이다. Block mode 방식은 벡터와 벡터 사이 연산을 matrix와 벡터 사이 연산으로 표현함으로써, 일정 개수의 입력 벡터를 모아서 matrix로 표현함으로써, 일정 벡터의 block으로 지정하여 처리하는 방법이다. 이와 같은 방법은 모두 학습 시간을 줄이기 위한 방법이다. 이 때의 성능 수준을 살펴보면, 프랑스 방송뉴스에 대해, 약 65K 단어에 대한 실험에서, 4-gram에 대해 Word Error Rate (WER) 14.24%, FFNN 언어모델 이용 시 WER 13.61%로 나타나, 0.63%의 미미한 성능 향상을 결과가 되었다. Shortlist를 이용하여 출력층의 출력 개수를 줄일 수는 있었지만, 결과적으로 성능 향상은 미미하였으며, 학습 소요시간의 개선이 어느 정도인지가 명확하게 제시하지 못하였다.

[4]에서는 FFNN LM에서 은닉층을 여러 개 (1-4개)를 이용한 언어모델 성능 실험을 진행하였다. 특히, 은닉층의 개수 및 은닉 unit수, feature의 개수에 따른 성능 변화를 제시하였다. 영어 WSJ 도메인으로, 약 20K 단어에 대해 결과를 제시하였다. 4-gram의 WER 22.3%에서 FFNN LM 이용시 20.8%로 나타났으며, 1.5%의 미미한 성능 향상 결과를 제시하였다. 학습자료는 약 900K 문장으로 구성된 2천 3백 50만 단어를 사용하였으며, 학습 소요 시간은 은닉 층 3개, 은닉 unit 500개, feature 30개에 대해 약 72시간으로 나타나고 있다. 이 연구에서 사용한 학습자료가 증가하는 경우에는 학습 소요 시간 또한 증가하는 문제점이 있어, 이를 해결하기 위한 방안 제시가 부족하였다.

3. 연속 벡터 공간상에서의 단어 벡터 모델을 이용한 입력 벡터의 차원 감소 연구

본 장에서는 연속 공간상에서의 단어 벡터 모델 및 속성에 대해 기술한다. 단어 벡터로 표현하게 되면, 벡터간의 유사도를 통하여 벡터 사이의 관계를 의미적 또는 구조적 정보를 포함하도록 할 수 있다. 이와 같이 단어 벡터로 표현함으로써, 학습자료에 나오지 않았던 unseen word sequence에 대한 확률을 연속 공간상에서 neural network로부터 확률 추정 함수를 통해 smoothing 해주는 효과를 얻을 수 있기 때문이다.

단어란 자립적으로 쓸 수 있는 말이나 이에 준하는 말, 또는 그 말의 뒤에 붙어서 문법적 기능을 나타내는 말을 의미하며, 언어학적인 심볼로 표현된다. 이와 같은 언어학적인 단어는 표면적인 표현과 내재하는 의미를 가지고 있다. 기존에는 이런 단어들은 단어간의 내재하는 의미 연관 관계를 표현하지 못하는 이산 공간상에서 표현되었다. 즉, 표면적으로 다른 단어들은 이산 공간상에서는 완전히 다른 단어로 취급되었다. 그러나 사람들은 언어학적인 선험적 경험을 통해서, 특정 단어간의 연관 관계가 있음을 알고 있다. 그래서, 단어를 연속 벡터 공간상에서 단어 벡터로 표현할 수 있다면, 단어간의 유사도를 이용한 연관 관계를 표현 할 수 있으며, 단어간의 관계를 수학적으로 모델링 할 수 있는 연속 단어 벡터 모델이 제시 되었다 [5]. 단어를 벡터로 표현하게 되면, 정량적으로 단어를 표현 할 수 있으며, 정량적인 값인 벡터의 크기 (magnitude) 값과 방향 (direction)을 이용하여 연속 공간 상에서 하나의 점 (point)으로 표현 된다. 단어를 정량적으로 표현 할 수 있다면, geometric 공간상에서의 proximity (공간상의 가까움을 나타내는 척도)를 통해 단어 간의 연관 관계를 나타내는 유사도 (similarity)를 측정할 수 있다.

연속 공간상에서의 단어 벡터 모델은, 위에서 언급한 내용대로 단어를 벡터로 표현한다는 것이며, 단어간의 관계로 대표되는 단어 유사도는 *k*-dimensional space에서 proximity로 표현된다. 여기에서, *k*는 정수 범위로써 2에서 임의의 정수 *k*의 범위를 나타낸다. 단어 벡터는 임의 정수 *k*로 인하여, *k*-dimensional space로 나타나며, 이와 같은 고 차원 공간상에서는 사람이 직접 관찰 및 분석하는 것이 어렵다. 만약에 *k*를 2로 한정되어 표현한다면, 2-dimensional 공간상에 단어 벡터로 다음의 <그림 3>과 같이 악기에 해당하는 5개 단어에 대해, 연속 공간상에서의 단어 벡터로 표현된다.

이와 같은 *k*-dimensional 공간상에서의 단어 벡터에서, 단어간에 공간상 proximity는 각 단어의 의미가 얼마나 유사한지를 표현한다. 즉 서로간의 유사한 단어들은 개념적으로 또는 공간적으로 가깝게 위치한다고 생각하며, 반면에 유사하지 않는 단어들은 개념적으로 또는 공간적으로 멀리 떨어져 위치한다고 생각하게 된다. <그림 2>에서, 'violin'은 'piano'보다 'cello'에 더 가깝게 나타나며, 이는 'violin'의 meaning이 'piano'의 meaning보다 'cello'의 meaning에 더 유사함을 나타낸다. 이를 통해 유추할 수 있는 결론은, 공간상의 proximity는 유사도를 나타낸다는 점이다. 그리고 proximity를 표현하기 위해서는, 단어 벡터에서 단어들은 공간상에 다른 point를 점유하고 있어야 하며, 이를 공간상의 단어 벡터들은 point들로 표현된다는 의미로 해석된다. 이 두 가지 내용 중 더 중요한 사항은, 단어가 연속 벡터 공간상에서의 하나의 point로써 표현된다는 것이 아니라, 단어와 단어 사이의 유사도로 표현되는 proximity를 어떻게 잘 표현하는 방법이 더 중요하다는 점이다. 그 이유는, 단어에 대한 벡터 공간에서의 point로 표현하는 방법에서, 단어에 대한 시간 또는 context에 따라 변하지 않는 절대적인 point 지점은 없기 때문이다. 단어의 의미는 다양한 코퍼스 자료에서 context에 따라 달라질 수 있으며, 그래서 유사한 context내에서 나타나는 유사한 단어들은 비슷한 의미을 가진다는 점을, 연속 공간사에서 단어 벡터로 잘 표현하고자 한다.

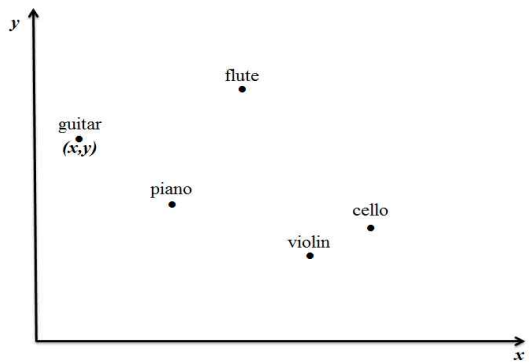

그림 2. 2-dimensional word vector space 예제
(words: guitar, piano, flute, violin, cello)
Figure 2. Example of 2-dimensional word vector space
(words: guitar, piano, flute, violin, cello)

언어학자들은 '단어의 의미는 고립된 단어에서 찾을 수 있는 정보가 아니라, context상에서의 단어가 의미을 가지고 있다'고 생각했다 [6]. 이와 비슷한 의견으로는 'Distributional Hypothesis: words with similar distributional properties have similar meanings.'이 있다 [7]. 이 생각은 단어의 distributional property에 해당하는 통계적 정보를 활용하여 표현 할 수 있음을 나타낸다. 즉, 통계적 정보라는 것은 사람의 지식이나 제약이 일부러 가해지지 않은 학습 코퍼스 자료를 이용하기 때문에, 단어 간의 관계를 이 학습 코퍼스 자료로부터 자동적으로 생성할 수 있음을 나타낸다. 단어 벡터 생성에 있어서 사람이 직접 단어 벡터를 생성하는 것이 아닌, 학습 코퍼스 자료로부터 자동으로 기계 학습 기법을 적용하여 생성 가능하다는 것이 중요하다.

Distributional hypothesis에 대한 또 다른 해석을 살펴보자. [6]에서는, 'Words which are similar in meaning occur in similar contexts.'라고 설명한다. '비슷한 의미를 가지는 단어들은 비슷한 context에서 다시 출현된다'라는 것을 나타낸다. [7][8] 에서는, 'Words with similar meanings will occur with similar neighbors if enough text material is available.' 이라고 설명하고 있다. 이 내용은 '충분한 양의 학습 자료가 제공되는 경우에, 비슷한 의미를 가진 단어들은 비슷한 neighbor 단어들을 가지고 있다'라는 것을 나타낸다. Distributional hypothesis를 만족하는 단어 벡터를 생성하는 대표적인 모델은 'Word2Vec' 이며, 본 연구에서는 이를 활용하여 입력 층의 구조를 개선하였다 [9]. Word2Vec의 대한 자체 성능 평가는 [10]에서 자세히 설명하고 있다.

표 2. 입력층 구조 개선을 통한 학습속도 개선
Table 2. Training speed improvement using word vector in input layer

Input layer	No. of hidden units	학습 소요 시간	비고
1-of-|V| coding	600	약 30일	비교 평가
Word vector	600	약 14일	

입력 층 구조 개선을 통한 학습속도 개선 결과는 다음의 <표 2>와 같다. Continuous word vector를 이용하여 학습한 경우, 동일한 hidden unit으로 했을 경우에는, 약 2배 정도의 학습속도 개선을 보여주었다.

4. 연속 공간상에서의 단어 벡터 검증

Proximity를 이용한 단어 벡터 검증 방법으로는 2가지 방법

이 있다. 첫 번째 방법은, 단어 벡터 visualization에 해당하는 방법으로써, 고차원 연속 공간상의 벡터를 2차원 공간상에 투영함으로써, 사람이 직접 눈으로 확인하는 방법이다. 두 번째 방법으로는, 사람이 생각하는 단어간의 유사도와 단어 벡터로 표현된 단어들로 생성된 cosine similarity와의 통계적 비교 분석 방법이 있다. 본 절에서는 두 번째 방법에 대해 실험을 진행하여 결과를 살펴본다. 실험을 위한 단어 벡터는, Word2Vec [9] 소프트웨어를 이용하여 약 3,940만 단어로 구성된 영어 텍스트 자료로부터 생성한 각 단어별로 100차원의 단어 벡터를 이용하여 실험을 진행하였다. Proximity의 정량적 measure로써, cosine similarity를 이용하여 측정하였다.

단어 벡터 검증 방법으로는, 사람이 생각하는 단어 간의 유사도와 단어 벡터로 표현된 단어들로 생성된 cosine similarity와의 통계적 비교 분석 방법이 있다. 통계적 비교 분석 방법으로는 Spearman's Rank Correlation Coefficient (SRCC) 방법을 이용하였다. 이 방법은 Spearman's rho (ρ)로 불리기도 하며, 두 변수간의 rank를 이용한 통계적 관계를 나타내는 파라미터이다. 두 변수간의 rank가 서로 정확히 일치하게 되면, +1을 나타내는 특성이 있다. Word-pair에 대해서, 사람이 생각하는 단어간의 유사도와 단어 벡터의 cosine similarity로 생성된 유사도간의 rank의 통계적 관계를 살펴보게 된다. Word-pair로 널리 이용되는 코퍼스로는 WS-353, MEN-3000, RW-2034 등이 있다. WS-353 dataset은 353개의 영어 word pairs로 구성되어 있으며, 각 word pair는 사람이 직접 점수를 준 similarity 값을 가지고 있다 [11]. MEN-3000 data set은 3,000개의 word pairs로 구성되어 있으며, uKWaC와 Wackypedia 코퍼스에서 최소 700번 이상 나온 단어들로 구성되어 있다 [12]. RW-2034는 rare-words로 구성된 2,034개 word pairs로 구성되어 있다 [13]. 이를 기존의 DNN 언어모델 연구에서 생성된 word vector와의 비교 평가를 수행하였다. Word vector를 생성하기 위한 학습 자료는 약 3,940만 단어로 구성된 영어 텍스트 자료를 이용하였다. 다음의 <표 3>는 100차원 word vector에 대한 word-pair 코퍼스의 SRCC 결과를 보여주고 있다. DNN 언어모델을 이용한 word vector 생성 방법은 DNN 구조 기반으로 언어모델 학습 과정시 나오게 되는 단어의 벡터 결과물을 이용하였다. SRCC 분석 결과를 통해 알 수 있는 사실은 사람이 생각하는 단어간 관계를, 기존의 DNN 언어모델링에서 사용되었던 단어 벡터보다 'Word2Vec'의 skip-gram 모델 방식이 더 잘 표현하고 있음을 확인하였다.

표 3. 100차원 단어 벡터에 대한 word-pair 코퍼스의 SRCC
Table 3. SRCC of word-pair corpora using 100-dimensional word vector

Word vector 생성 방법	SRCC (Spearman's rho (ρ))		
	WS-353	MEN-3000	RW-2034
Word2Vec (skip-gram)	0.508	0.430	0.358
기존 DNNLM	0.284	0.214	0.207

[10]에서는, 기존의 미리 학습된 word vector들에 대한 성능 비교 평가를 수행한 결과를 제시하였다. WS-353 word-pair 코퍼스에 대해서, 사람이 생각하는 단어간의 관계 유사도와 단어 벡터로 생성한 단어간의 관계 유사를 측정한 결과를 제시하고 있다. Word2Vec의 skip-gram 방식의 80차원의 SRCC는 0.680이고, 기존 DNNLM으로 생성된 SRCC는 0.367으로 결과를 제시하고 있다. 이는 <표 3>에서 실험한 결과와 같이 Word2Vec의 방식이 차원의 크기에 관계 없이, 사람이 생각하는 단어간의 관계를 통계적 차원에서 더 잘 표현한다는 결과를 보여주었다. 이를 DNN 언어모델 적용되는 방식은 다음의 형태로 진행된다. DNN 언어모델 학습시 Word2Vec의 결과를 lookup 테이블 형태로 구축하고, 이로부터 각 단어의 단어 벡터를 입력 층에 전달하는 방식으로 적용한다. 이에 대한 WSJ 도메인의 언어모델 성능 평가는 다음 표 4와 같다.

표 4. WSJ 도메인 실험 결과
Table 4. Performance evaluation of language model for WSJ domain

언어모델	Perplexity	WER (%)	Relative WER reduction (%)
3-gram	146.91	12.32	기준
DNN모델 (Word2Vec 미적용)	122.32	11.35	7.87
DNN 모델 (Word2Vec 적용)	125.95	11.40	7.46

5. 결론

DNN을 이용한 언어모델은, 단어와 같이 index만을 표현하는 이산 공간에서의 벡터가 입력으로 사용되고, 차원이 높은 벡터가 입력으로 사용되기에 DNN 적용시 계산량이 많이 요구되는 문제점이 있어, 학습에 많은 시간이 소요되었다. 이를 개선하기 위하여, 본 연구에서는 입력 층의 표현을 1-of-$|V|$ coding 형태에서 연속 단어 벡터를 이용하여 입력 차원을 감소 방법을 제시하고, 성능 검증을 진행하였다. 약 353개 word-pair에 대해서, 사람이 생각하는 단어간의 관계 유사도와 단어 벡터로 생성한 단어간의 관계 유사를 측정해 보았다. 통계적 correlation을 보여주는 SRCC 값이 0.508으로, 기존 DNN 언어모델의 0.284 보다 높게 측정되었다. 연속 단어 벡터 모델이 사람이 생각하는 단어 간의 관계를 더 잘 반영하고 있음을 보

여주며, 이는 추후 DNN 기반 언어모델의 성능 향상에 영향을 줄 것으로 기대한다.

참고문헌

[1] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C. (2003). A neural probabilistic language model, J*ournal of Machine Learning Research*, Vol. 3, 1137-1155.

[2] Bengio, Y. (2009). Learning deep architectures for AI, *Journal of Foundations and Trends in Machine Learning*, Vol. 2, No. 1, 1-127.

[3] Schwenk, H. & Gauvain, J. (2005). Training neural network language models on very large corpora, in *Proc. Empirical Methods in Natural Language Processing*, 201-208.

[4] Arisoy, E., Sainath, T., Kingsbury, B. and Ramabhadran, B. (2012). Deep neural network language models, in *Proc. NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, 20-28.

[5] Turney, P. & Pantel, P. (2010) From frequency to meaning: vector space models of semantics, *Journal of Artificial Intelligence Research*, Vol. 37, No. 1, 141-188.

[6] Schutze, H. & Pedersen, J. (1995). Information retrieval based on word sense, in *Proc. Symposium on Document Analysis and Information Retrieval*, 161-175.

[7] Rubenstein, H. & Goodenough, J. (1965) Contextual correlates of synonymy, *Communications of the ACM*, Vol. 8, No. 10, 627-633.

[8] Bruni, E., Boleda, G., Baroni, M. and Tran, N. (2012). Distributional semantics in technicolor, in *Proc. 50th Annual Meeting of the Associations for Computational Linguistics*, 136-145.

[9] Mikolov, T. (2013). Word2Vec, https://code.google.com/p/word2vec.

[10] Faruqui, M. & Dyer, C. (2014). Community evaluation and exchange of word vectors at wordvectors.org, in *Proc. Associations for Computational Linguistics*, 1-6.

[11] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E. (2001). Placing search in context: the concept revisited, in *Proc. The Tenth International World Wide Web Conference*, 406-414.

[12] Bruni, E., Boleda, G., Baroni, M. and Tran, N. (2012). Distributional semantics in technicolor, in *Proc. 50th Annual Meeting of the Associations for Computational Linguistics*, 136-145.

[13] Luong, M., Socher, R. and Manning, C. (2013). Better word representations with recursive neural networks for morphology, in *Proc. Computational Natural Language Learning*, 1-10.

• **김광호 (Kim, Kwang-Ho), 제1저자**
서강대학교 컴퓨터공학과
서울시 마포구 신수동 1번지
Tel: 02-715-2715
Email: kimkwangho@sogang.ac.kr
관심분야: Speech Recognition, Natural Language Processing, Spoken Multimedia Content Search
현재 컴퓨터공학과 대학원 박사과정 재학 중

• **이동현 (Lee, Donghyun), 제2저자**
서강대학교 컴퓨터공학과
서울시 마포구 신수동 1번지
Tel: 02-715-2715
Email: redizard@sogang.ac.kr
관심분야: Speech Recognition, Spoken Multimedia Content Search
현재 컴퓨터공학과 대학원 박사과정 재학 중

• **임민규 (Lim, Minkyu), 제3저자**
서강대학교 컴퓨터공학과
서울시 마포구 신수동 1번지
Tel: 02-715-2715
Email: lmkhi@sogang.ac.kr
관심분야: Speech Recognition, Spoken Multimedia Content Search
현재 컴퓨터공학과 대학원 박사과정 재학 중

• **김지환 (Kim, Ji-Hwan), 교신저자**
서강대학교 컴퓨터공학과 부교수
서울시 마포구 신수동 1번지
Tel: 02-715-2715
Email: kimjihwan@sogang.ac.kr
관심분야: Spoken Multimedia Content Search, Speech Recognition using Cloud Computing and Dialogue Understanding Speech Recognition, Spoken Multimedia Content