딥러닝 스터디

Efficient Estimation of Word Representation in Vector Space

김제우

목차

딥러닝스터디

Efficient Estimation of Word Representation in Vector Space

Word2Vec 논문 읽기

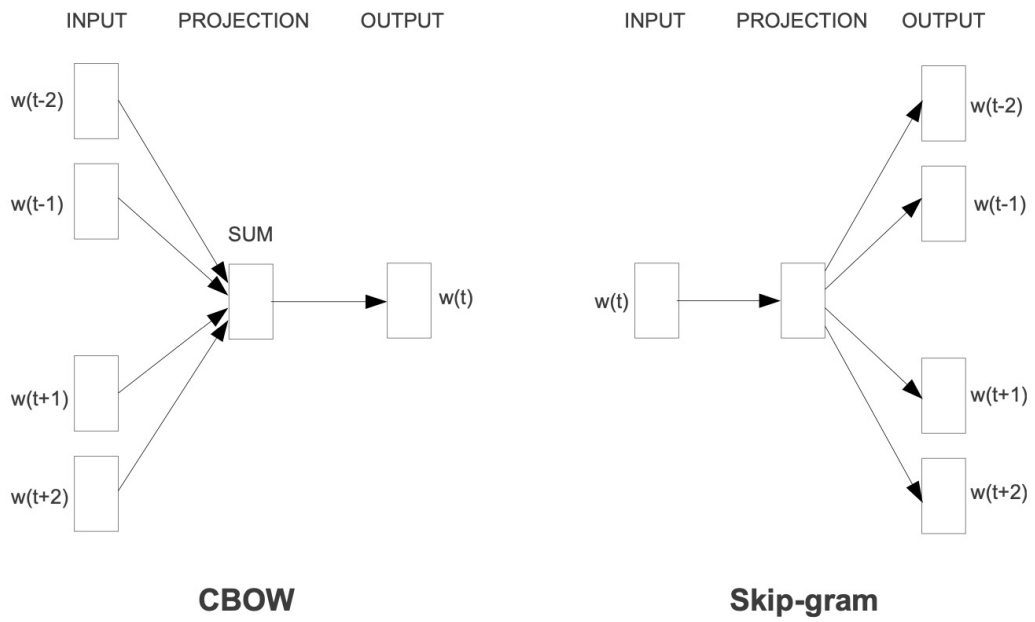
제목 보고 감 잡기

- Efficient Estimation of Word Representation in Vector Space
- 벡터 공간에서의 단어 표현의 효율적인 추정

예상할 수 있는 것
- Efficient 라는 의미에서 기존에 word representation을 어떻게 했는지 나올 것이다.
- 기존에 비해 어떤 점을 개선했는지를 보면 좋을것

Word2Vec 논문 읽기

모델 구조 main feature 보고 예상 해보기



예상할 수 있는 것

- CBOW와 Skip-gram의 구조에 대해서 설명한다. -> weight는 어디에 들어가는지?
- 단어를 t를 기준으로 양쪽 2개씩 본다. -> 더 늘리기도 하나?
- CBOW는 단어들을 보고 1개의 단어추론, Skip-gram은 단어 하나를 보고 4개의 단어 추론

Abstract

- 초 대형 데이터셋에서 연속적인 단어들의 벡터 표현을 연산하기 두 개의 새로운 모델을 제안
- 이 단어 표현들의 퀄리티는 단어 유사도 task를 통해 측정했으며 결과들은 이전 까지 best 성능을 보인 기술이었던 neural network 기반 방식들과 비교했다.
- 정확도(accuracy)에서 더 적은 연산으로 큰 성능 향상이 있음을 확인했다.
- 다시 말해 1.6 billion words 데이터셋에서 높은 퀄리티의 단어 벡터 표현을 하루도 학습시키지 않고 얻을 수 있었다.
- 나아가 우리는 이 벡터표현들이 state-of-the-art(최첨단) 성능을 보인다는 것을 우리의 테스트 셋에서 syntactic(구문)과 semantic(의미) 단어 유사도 측정을 통해 보여줄것이다.

1. Introduction

- 원자 단위로 표현하는 방식 (N-gram)
  - 단어들은 사전 색인으로 표현한다
  - 단어들 간의 유사도를 표현할 수 없다.
  - 단순, robust(이상치에 영향을 조금 받음), 적은 데이터를 가지고 복잡한 모델을 훈련하는것이 많은 데이터를 가지고 훈련한 간단한 모델이 더 뛰어나다는 관측이 장점
  - N-gram

unigrams : an, adorable, little, boy, is, spreading, smiles

bigrams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles

trigrams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles

4-grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

1. Introduction

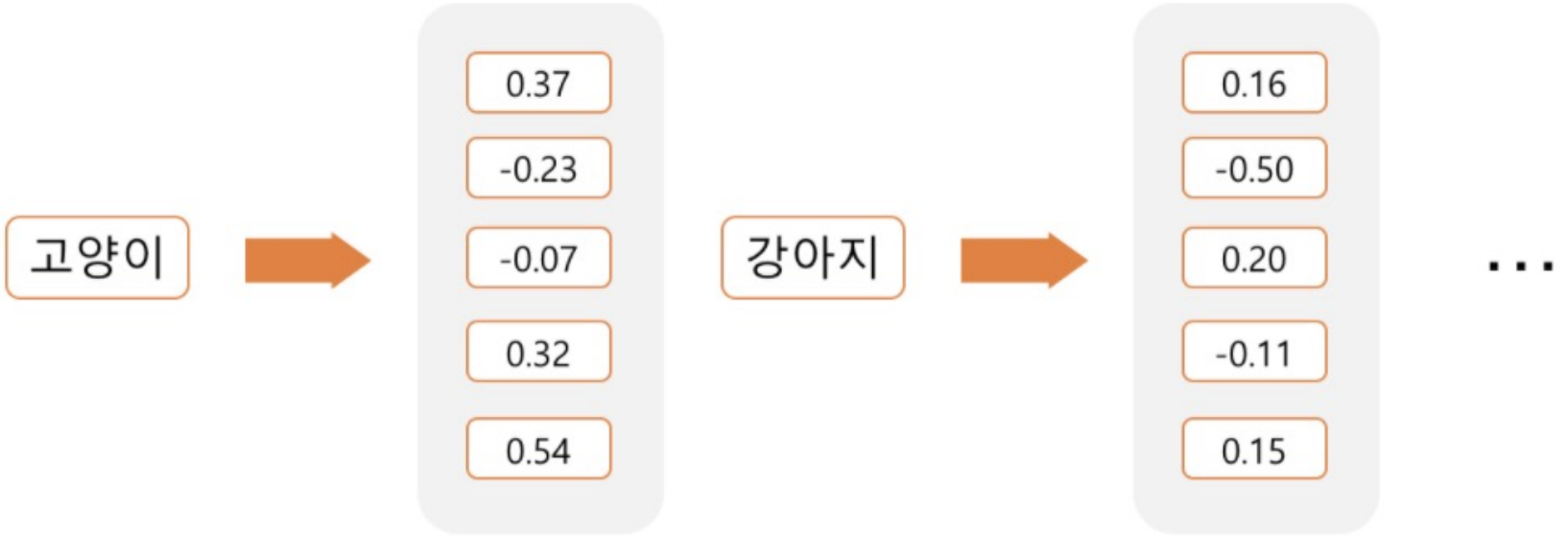
색인으로 표현하는 단어 One-hot

	You 0	Say 1	Goodbye 2	And 3	I 4	Hello 5	. 6
you	1	0	0	0	0	0	0
say	0	1	0	0	0	0	0
goodbye	0	0	1	0	0	0	0
and	0	0	0	1	0	0	0
i	0	0	0	0	1	0	0
hello	0	0	0	0	0	1	0
.	0	0	0	0	0	0	1

[0,1,2,3,4,1,5,6]

1. Introduction

- Distributed Representation
  - Neural network 기반의 방식들
  - N-gram 보다 성능이 훨씬 좋다.



1. Introduction

- Goals of the Paper
  - 이전보다 월등히 많은 데이터셋에서 높은 퀄리티의 단어 벡터를 학습하는 것
  - 퀄리티를 측정하는 법
    - 비슷한 단어들은 가까이에 있다는 가정하에 벡터들간의 다중 유사도(multiple degrees of similarity)를 측정하는 방식
    - vector('King') – vector('Man') + vector('Woman') = vector('Queen')
  - 이 논문에서는 새로운 모델 구조를 만들어서 vector representation연산의 정확도를 높일것임
  - 구문 규칙과 의미 규칙을 측정하기 위한 새로운 데이터셋을 만들었음
  - 훈련시간과 정확도가 단어의 벡터 차원과 training data 양에 얼마나 의존하는지도 확인함.

1. Introduction

- Previous Work
  - NNLM
    - Linear Projection layer와 Non-linear Hidden layer를 기반으로 feedforward neural network를 통해 단어 벡터 표현과 통계학적인 언어 모델을 결합함.
  - 다른 구조의 NNLM
    - 단어 벡터들을 single hidden layer로 한번 학습하고 다시 nnlm으로 학습하는 방식
    - 단어를 single hidden layer에 우선 학습한다.
    - 계산 비용이 매우 크다.

2. Model Architectures

- Training Complexity 정의

$$O = E \times T \times Q,$$

- E 는 에폭의 수
- T 는 training set의 단어의 개수
- Q는 각 model 구조에 의해 정의 된다.

이미지 출처 : https://www.youtube.com/watch?v=sidPSG-EVDo

2. Model Architectures

딥러닝스터디

- Feed-Forward Neural Net Language Model(NNLM)

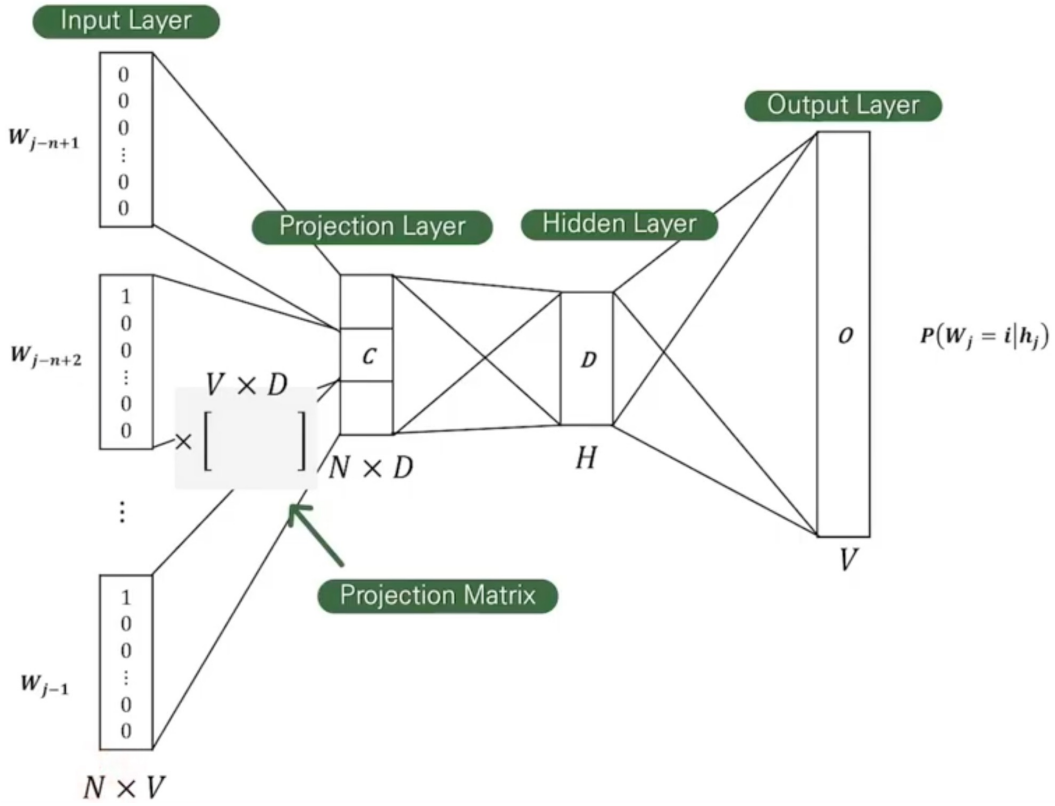
Notation

N: $input$으로 들어가는 이전 단어의 개수

D: $projection$ 후의 $dimension$

H: $hidden\ layer\ size$

V: $Vocabulary\ size$



이미지 출처 : https://www.youtube.com/watch?v=sidPSG-EVDo

2. Model Architectures

- Feed-Forward Neural Net Language Model(NNLM)
  - Linear Projection layer와 Non-linear Hidden layer를 결합한 형태
  - 문맥의 길이에 따라 사용할 단어의 개수를 고정해주어야 한다.
  - History만을 보고 예측하기 때문에 미래 시점의 단어들을 고려하지 않는다.
  - 모델 복잡도

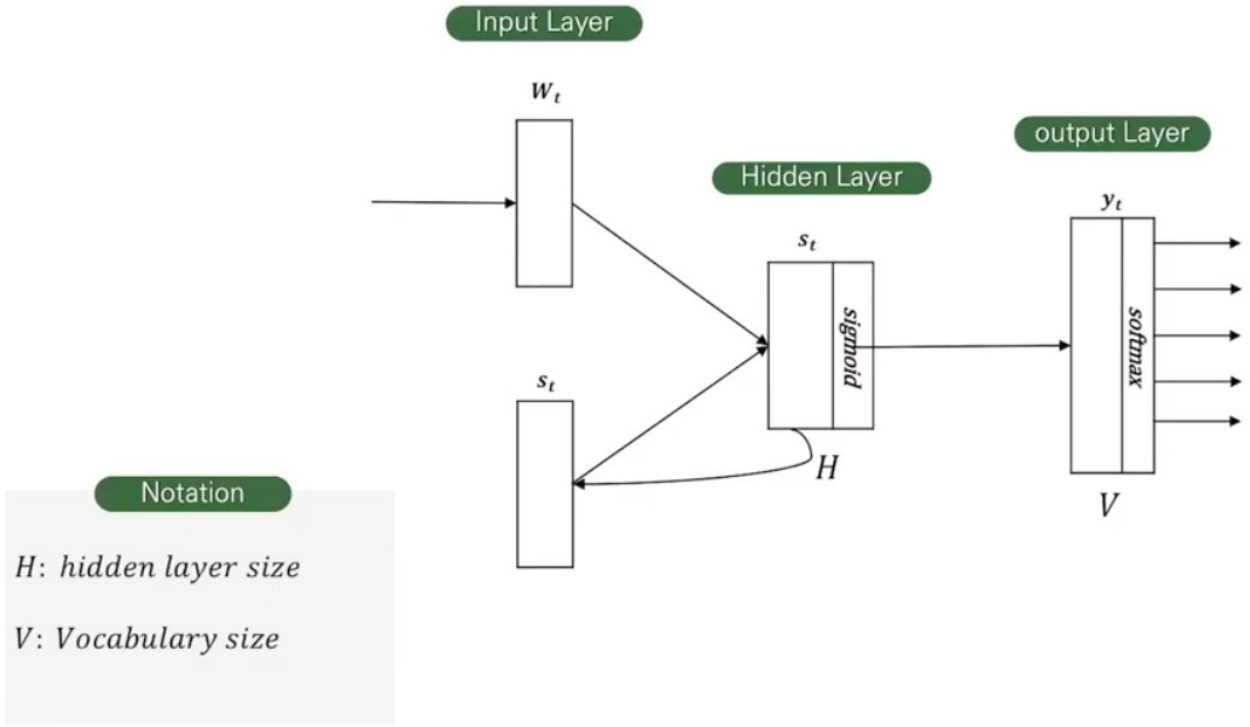
$$Q = N \times D + N \times D \times H + H \times V,$$

  - H는 500~1000개
  - P는 500~2000개
  - N는 10정도
  - H x V가 매우 큰데 Hierarchical softmax기법을 사용하면 줄일 수 있다.
  - 따라서 N x D x H가 Q에 가장 크게 영향을 미친다.

2. Model Architectures

딥러닝스터디

- Recurrent Neural Net Language Model (RNNLM)



이미지 출처 : https://www.youtube.com/watch?v=sidPSG-EVDo

2. Model Architectures

- Recurrent Neural Net Language Model (RNNLM)
  - NNLM의 한계를 극복하기 위해 생겨남. (문맥의 길이 명시해야한다는 문제점)
  - projection layer가 없음
  - recurrent matrix가 hidden layer와 시간 흐름의 연결을 갖고 있음
  - short term memory를 생성할 수 있고, 과거의 정보가 지속적으로 반영 될 수 있다.
  - 모델 복잡도

$$Q = H \times H + H \times V,$$

  - HxV는 역시 hierarchical softmax로 줄일 수 있음
  - HxH가 가장 주요함.

2. Model Architectures

- Parallel Training of Neural Networks
  - 거대한 데이터셋에 대해 실험하기 위해 병렬학습을 활용함.
  - DistBelief라는 프레임워크를 사용
  - Adagrad를 사용한 미니배치 경사하강법을 사용함.

3. New Log-linear Models

- 계산 복잡도를 줄일 것임
- 2개의 새로운 모델 구조
- 비선형 hidden layer가 가장 큰 계산 복잡도의 원인이였다.

3. New Log-linear Models

3.1 Continuous Bag-of-Words Model

- 비선형 hidden 층이 사라짐
- projection 층이 모든 단어들에서 공유된다.
- 이외에는 feedforward NNLM과 유사함.
- 단어의 순서가 projection에 영향을 주지 않기 때문에 bag-of-words 모델이라함.
- 앞뒤로 4개씩의 단어를 입력으로 분류함.

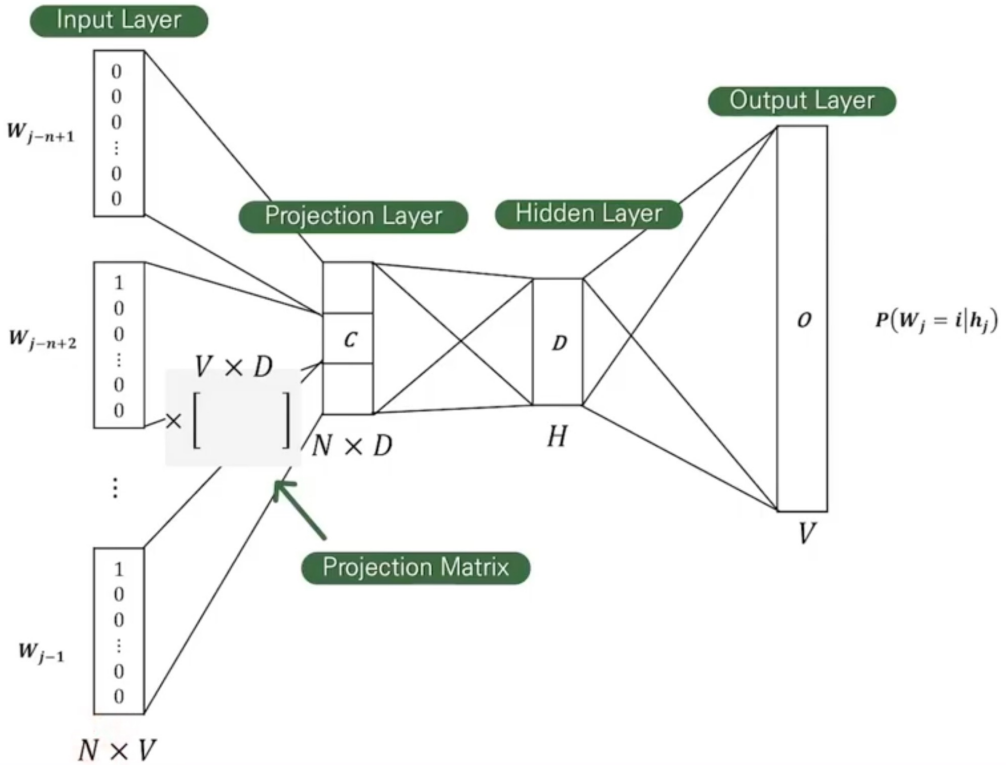
$$Q = N \times D + D \times log_2(V).$$

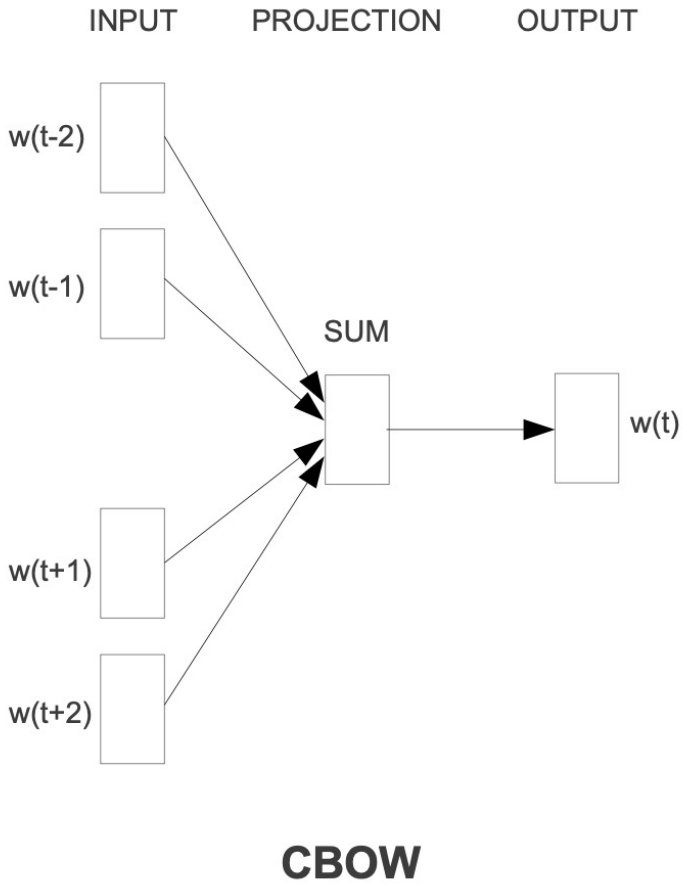
$$Q = N \times D + N \times D \times H + H \times V,$$

3. New Log-linear Models

딥러닝스터디

3.1 Continuous Bag-of-Words Model





3. New Log-linear Models

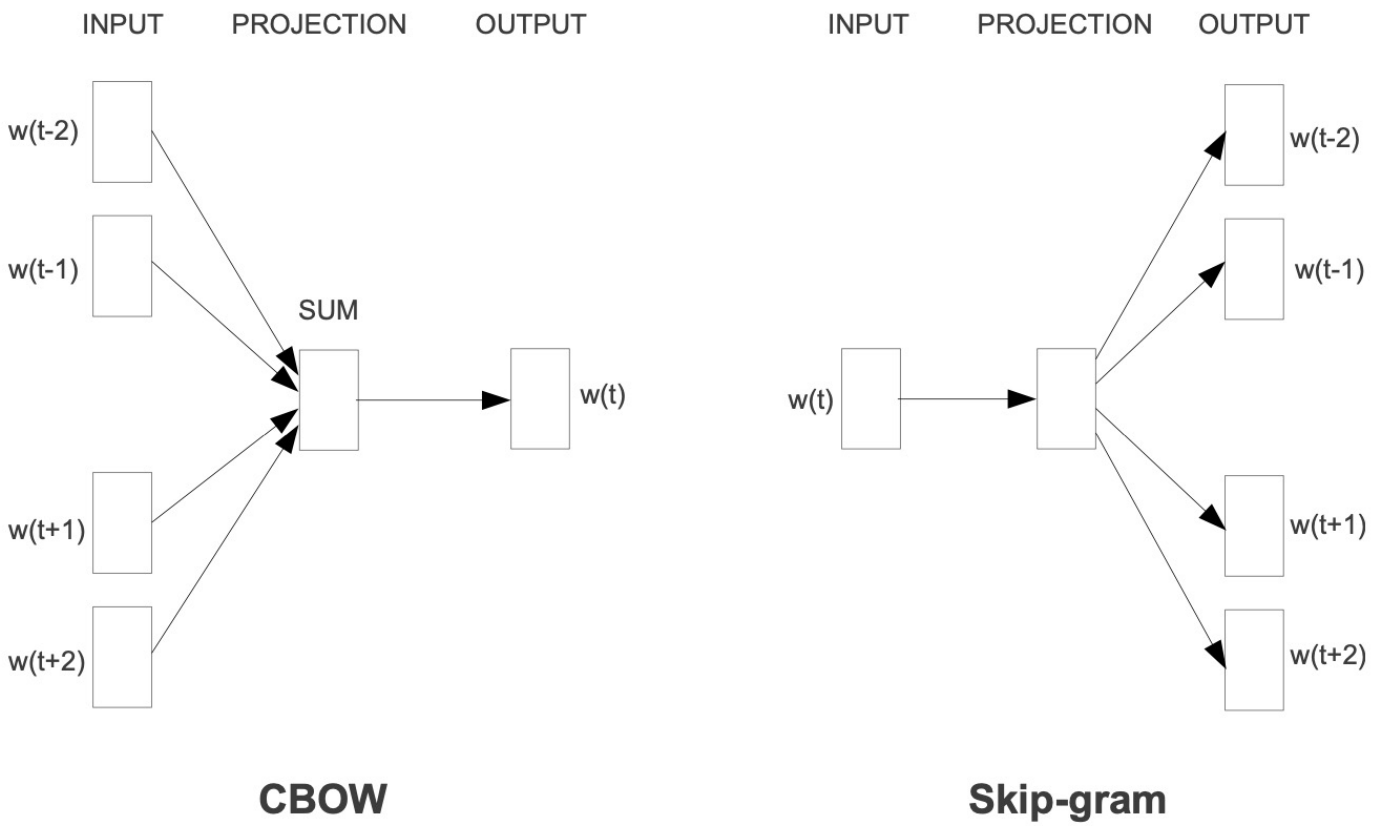
3.2 Continuous Skip-gram Model

- context에 기반해 현재 단어를 예측하는 대신에 같은 문장의 다른 단어에 기반한 단어의 분류를 극대화한다.
- continuous projection layer와 함께 log-linear classifier에 사용하고, 현재 단어 앞뒤의 특정 범위안의 단어를 예측함.
- range를 증가시키면 단어 벡터의 퀄리티가 향상되지만 계산 복잡도가 늘어남.
- 거리가 먼 단어는 가까운 단어보다 현재 단어와 연관성이 떨어질 것이므로 훈련 세트에서 이런 단어들은 샘플링을 적게 해서 가중치를 줄였다.

$$Q = C \times (D + D \times log_2(V)),$$

$$Q = N \times D + N \times D \times H + H \times V,$$

3.2 Continuous Skip-gram Model



4. Results

- 다른 버전의 단어 벡터들의 성능을 비교하기 위하여 이전 논문들의 예시 단어와 가장 비슷한 단어를 표로 보여줄 것이고 이를 직관적으로 이해할 것이다.

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

4. Results

4.1 Task Description

- 5가지 종류의 의미론적 질문과 9가지 종류의 구문론적 질문을 담는 테스트 셋을 정의함.
- 8869개의 의미론적인 질문들과 10675개의 구문론적인 질문
- 예를 들어 68개의 미국 도시와 주들을 쌍으로 만들어서 2.5k 개의 문제 제작.
- New York 같은 두개의 단어로 이루어진 단어는 포함하지 않음.

4. Results

4.2 Maximization of Accuracy

- 60억개의 단어를 포함하는 구글 뉴스 말뭉치를 훈련에 사용함
- 어휘의 크기를 100만개로 제한하였음.
- 3만개로 우선 CBOW 테스트 실험한 결과
- 차원이나 데이터가 늘어나도 향상이 절감되는 것을 볼 수 있음.

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

4. Results

4.2 Maximization of Accuracy

- 차원과 데이터의 양을 동시에 증가 시킬 것임.
- 다시 말하면 차원과 데이터 양이 많은 경우에만 쓸 수 있는 기법임.
- 계산 복잡도는 훈련 데이터 양을 두배로 늘릴때 벡터 크기를 두배 늘린 것과 같은 계산 복잡도가 증가했다.
- 학습률은 0.025이고 서서히 감소 시키는 weight decay
- SGD 3사이클 진행

4. Results

4.3 Comparison of Model Architectures

- 다른 모델 들과의 비교
- 640차원 단어 벡터를 사용하면서 비교
- 테스트셋 전체 사용 (LDC 말뭉치 사용)
- 1개 cpu에서 8주간 훈련한 RNN 언어모델과 비교함.

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

4. Results

딥러닝스터디

4.3 Comparison of Model Architectures

- NNLM벡터들은 RNN보다 더 좋은 성능을 보인다.
- CBOW는 NNLM보다 구문론적인 작업에서 좋은 성능을 보이고 의미론적은 비슷Skip-gram은 CBOW모델보다 구문론적으로는 약간 부족함.
- Skip-gram은 모든 부문에서 좋음 구문론은 CBOW가 더 좋음

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

4. Results

딥러닝스터디

4.3 Comparison of Model Architectures

- 공식 규격과의 비교 데이터

Model	Vector Dimensionality	Training words	Accuracy [%]		
			Semantic	Syntactic	Total
Collobert-Weston NNLM	50	660M	9.3	12.3	11.0
Turian NNLM	50	37M	1.4	2.6	2.1
Turian NNLM	200	37M	1.4	2.2	1.8
Mnih NNLM	50	37M	1.8	9.1	5.8
Mnih NNLM	100	37M	3.3	13.2	8.8
Mikolov RNNLM	80	320M	4.9	18.4	12.7
Mikolov RNNLM	640	320M	8.6	36.5	24.6
Huang NNLM	50	990M	13.3	11.6	12.3
Our NNLM	20	6B	12.9	26.4	20.3
Our NNLM	50	6B	27.9	55.8	43.2
Our NNLM	100	6B	34.2	**64.5**	50.8
CBOW	300	783M	15.5	53.1	36.1
Skip-gram	300	783M	**50.0**	55.9	**53.3**

4. Results

4.4 Large Scale Parallel Training of Models

- 대규모 병렬 훈련DistBelief 분산 framework 사용

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

4. Results

딥러닝스터디

4.4 Microsoft Research Sentence Completion Challenge

- MRSCC는 NLP 기술 Task
- 문장당 글자가 빠져있는 1040개의 문장 테스트
- Skip-gram 단일 보다 RNN과 조합해서 높은 성능을 얻음

Architecture	Accuracy [%]
4-gram [32]	39
Average LSA similarity [32]	49
Log-bilinear model [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

5. Examples of the Learned Relationships 딥러닝스터디

- 정답과 비교할때 표의 결과는 60% 정도 정확도이다

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

5. Examples of the Learned Relationships 딥러닝스터디

- 더 큰 데이터로 학습한다면 더 좋은 성능을 낼 것이다.
- 정확도를 높이는 또 다른 방법은 한 개 이상의 관계를 공급하는 것이다.
- 관계를 형성하기 위해 1개에서 10개로 예시를 늘렸을때 10% 정도의 정확도 향상
- 다른 문제를 해결하는데 이 벡터화된 단어들 연산을 쓸 수도 있을 것이다.
- 예를 들면 가장 먼 벡터를 찾아내는 것.

6. Conclusion

- 본 논문에서는 의미론적, 구문론적 언어의 수집 작업의 다양한 모델에서 유도된 단어의 벡터 표현의 성능에 대해 연구했다.
- 우리는 인기있는 모델인 신경망 모델과 비교했을 때 매우 간단한 모델 구조를 사용하여 좋은 성능의 단어 벡터를 얻었다.
- 거대한 데이터 셋이 있다면 적은 계산 복잡도로 높은 차원의 단어 벡터를 정확하게 구할 수 있을 것이다.
- 기존보다 월등히 많은 단어 말뭉치(크기 제한 없음) CBOW skip-gram을 학습할 수 있게 되었다.
- Spearmans' rank에서 그전 최고 결과보다 50% 향상을 이끌었다.
- 단어의 의미를 기반으로 하는 감정 분석이나 비유 탐색등의 새로운 작업들에 적용할 수 있다.
- 기계 번역 실험에서도 결과가 좋았다.

7. Follow-Up Work

- 단일 머신 multi-threaded c++로 작업해서 훈련 속도를 향상 시킬것임.
- 1억 단어 이상으로 훈련된 140만개의 벡터를 배포할것임

2. Model Architectures

- Parallel Training of Neural Networks
  - 거대한 데이터셋에 대해 실험하기 위해 병렬학습을 활용함.
  - DistBelief라는 프레임워크를 사용
  - Adagrad를 사용한 미니배치 경사하강법을 사용함.

1. Introduction

- 그렇지만 여러 task들에서 이러한 간단한 방식은 한계를 보인다.
- 예를 들면, 관련(relevant) 도메인 데이터 내에서 자동 음성인식의 양이 제한된다.
  - 성능은 '고품질의 음으로 기록된 음성 데이터의 크기'에 의해 성능에 지배 된다.
- 기계번역에서는 현존하는 많은 언어로 된 말뭉치들은 몇 십억 단어나 그 이하 정도 이다.
- 따라서 기존 방식에서 약간의 스케일업을 하는 상황에서는 어떤 중요한 발전도 만들어내지 못한다.
- 그리고 우리는 더 고도화된 방식에 집중해야한다.

1. Introduction

- 최근 몇년간의 머신러닝의 발전에는 더 큰 데이터셋을 복잡한 모델구조로 학습시키는 것이 가능해졌고, 일반적으로 간단한 모델들을 능가한다.
- 아마 가장 성공적인 컨셉은 단어의 분산표현방식을 사용하는 것일 것이다.
- 예를 들면 neural network에 기반한 언어 모델들은 N-gram model들을 크게 능가한다.

1. Introduction

- 많은 최근 NLP 시스템들과 기술들은 원자 단위(atomic)로 단어를 다룬다.
  - 여기엔 단어들간의 유사도에 관한 notion(생각)이 없다.
  - 그래서 이 단어들은 사전안에서 색인으로서 표현된다.??
- 원자 단위로 단어를 다루는데는 좋은 이유들이 있다. – 간단하고, 강건하고, 관측 that 대형 데이터로 학습시킨 간단한 모델은 복잡한 시스템을 적은 데이터로 학습 시킨 것을 능가한다(outperform)는 점
- 원자 단위의 NLP 시스템 중 하나를 예로 들자면 N-gram 모델이고, 통계적 언어 모델을 사용한다.
  - 오늘날 N-gram을 단어가 사실상 virtually(사실상)10조개의 단어도 학습 시키는 것이 가능하다.

- 그렇지만 여러 task들에서 이러한 간단한 방식은 한계를 보인다.
- 예를 들면, 관련(relevant) 도메인 데이터 내에서 자동 음성인식의 양이 제한된다.
  - 성능은 '고품질의 음으로 기록된 음성 데이터의 크기'에 의해 성능에 지배 된다.
- 기계번역에서는 현존하는 많은 언어로 된 말뭉치들은 몇 십억 단어나 그 이하 정도 이다.
- 따라서 기존 방식에서 약간의 스케일업을 하는 상황에서는 어떤 중요한 발전도 만들어내지 못한다.
- 그리고 우리는 더 고도화된 방식에 집중해야한다.

1.1 Goals of the Paper

- 이 논문의 메인 목표는 10억 이상의 단어와 수백만 단어의 사전으로 만들어진 대형 데이터 셋을 사용해 높은 성능의 단어 벡터를 학습하는 것을 가능하게하는 기술을 소개하는 것이다.
- 우리가 아는 한 이전까지 제안된 모델들은 적당한(modest) 단어의 차원 수인 50-100개 정도에서 몇 억 단어만 넘어가도 성공적으로 학습할 수가 없었다.
- 최근에 제안된 기술들 중 벡터 표현의 퀄리티를 측정하는 방법들 중 우리는 유사한 단어들끼리는 서로 가까이에 있는 경향(tend)이 있는 것 뿐만 아니라 그 단어들은 다중 유사도가 있다는 기대치를 사용할 것이다.
- 이것은 inflectional languages(굴절 언어-억양)에서 빠르게 관측될 수 있다. 예를 들면 명사들은 여러 개의 단어 끝을 가질 수 있는데, 우리가 기존 벡터 공간 속 부분 공간에 있는 유사 단어를 검색할 때 비슷한 끝부분을 가진다는 것을 통해 찾을 수 있다.

1.1 Goals of the Paper

- 놀라운 점은 단어 표현의 유사도는 간단한 구문 규칙 패턴(원래 우리가 아는 구문 패턴)을 넘어선다는 것을 확인할 수 있었다.
- 단어 오프셋 기술을 사용하면 간단한 대수 operation들이 단어 벡터들에서 동작한다. 예시를 통해 볼 수 있는데 'King'벡터 – 'Man'벡터 + 'Woman'벡터의 결과는 단어 Queen벡터의 표현과 가장 가까웠다.
- 이 논문에서는 이런 벡터 연산의 정확도를 극대화한다.(새로운 모델 아키텍처(단어들간의 선형 규칙성을 보존한다(preserve))를 개발해서)
- syntactic과 semantic 규칙들을 동시에 측정할 수 있는 종합적인(comprehensive) 테스트셋을 설계했고, 많은 규칙들이 높은 정확도로 학습될 수 있음을 보여줄 것이다.
- 단어 벡터의 차원에 따라, 데이터의 양에 따라 학습 시간과 정확도는 어땠는지도 보여줄것이다.

1.2 Previous Work

- 단어의 표현을 continuous vector로 표현 하는 것은 긴 역사를 가지고 있음.
- 유명한 모델 구조 중 하나인 neural network language model을 추정하는 방식은 [1]에서 제안되었고, linear projection layer와 non-linear hidden layer 로 만들어진 feedforward neural network는 단어 벡터의 표현과 통계학적 언어 모델을 합치기 위해 사용되었다.
- 이 연구는 다른 방식으로도 많이 연구되었다.

- [1] A neural probabilistic language model. Journal of Machine Learning Research 2003, Y. Bengio, R. Ducharme, P. Vincent

1.2 Previous Work

- 다른 흥미로운 NNLM 아키텍쳐 방식은 [13,14]에서 소개되었는데, 이 방식은단어 벡터들을 단일 은닉층(hidden layer)를 이용하여 neural network 방식으로 처음 학습된다.
- 그 다음에 단어 벡터들은 NNLM을 학습하기 위해 사용된다.
- 결과적으로 단어 벡터들은 전체 NNLM 구조를 만들지 않아도 학습될 수 있다.
- 우리 논문에서는 이 구조를 확장하고, 하나의 스텝만으로 단어 벡터가 간단한 모델을 이용하여 학습하는 것에 집중할 것이다.
- 나중에 단어 벡터가 NLP 어플리케이션들[4,5,29]을 크게 향상시키고 단순화 했다는 것이 밝혀졌다.

[13] Language Modeling for Speech Recognition in Czech

[14] Neural network based language models for highly inflective languages

1.2 Previous Work

- 단어 벡터 자체의 추정은 다양한 모델 아키텍처를 사용하여 수행되었고 다양한 말뭉치 '4,29,23,19,9'에 대해 훈련되었으며, 그 결과 단어 벡터 중 일부는 향후 연구와 비교에 사용할 수 있게 되었다.
- 우리가 알기로는 이러한 구조들은 연산 비용이 비싸다. log-bilinear model은

2 Model Architectures