

Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov



Background Theory

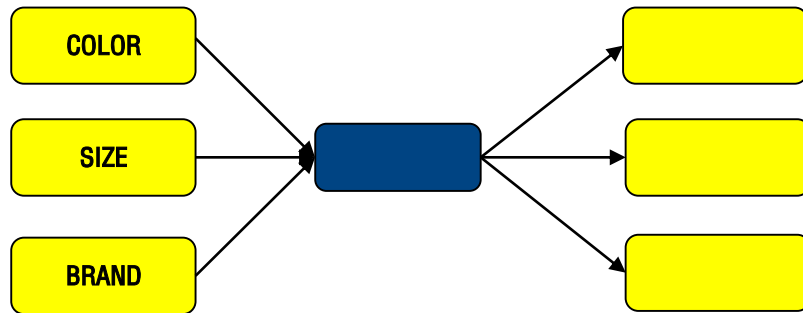
■ (2010.) If units of text have similar vectors in a text frequency matrix, then they tend to have similar meanings.

■ Non-distributed representation의 문제점 해결을 위해
Distributed representation이 제안됨

Background Theory

■ (2010.) If units of text have similar vectors in a text frequency matrix, then they tend to have similar meanings.

■ Non-distributed representation의 문제점 해결을 위해 Distributed representation이 제안됨



[Figure 1.] Distributed representation

저장하고자 하는 정보의 개수만큼 Unit을 만들어 Unit의 개수가 Exponential하게 많아지는 Non-distributed representation과 달리, 단 3개 Unit으로 조합되어 output에 정보가 모두 담길 수 있다.

Background Theory

■ One of the main advantage of these models is that the distributed representation achieves a level of generalization that is not possible with classical n-gram language models.

Background Theory

■ One of the main advantage of these models is that the distributed representation achieves a level of generalization that is not possible with classical n-gram language models.

The problem with n-gram

“올림픽개최” → “올림”, “올림픽“, “올림픽개”, “올림픽개최” ...

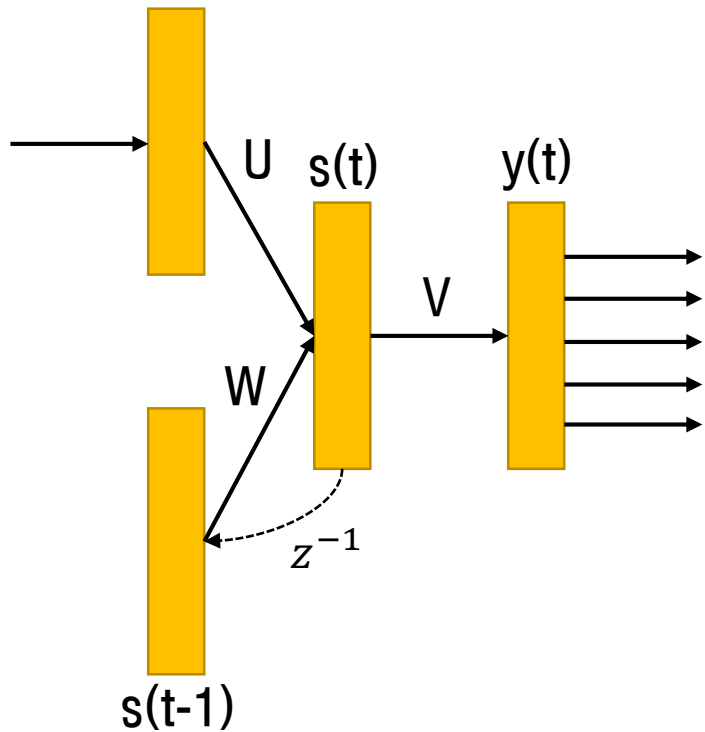
이렇게 보면 n-gram이 가장 정확하게 분석할 수 있지 않을까 하지만, 실제 형태소를 분석하면 “올림”의 의미는 수학의 올림도 있지만 올림픽의 올림의 출현빈도나 인지도 등에서 앞서는 까닭에 올림픽을 추출하게 된다는 단점이 있다.

Thesis's goal

“새로운 Vector Offset Method와 RNN의 결합을 통한 좋은 성능”



Recurrent Neural Network Model



[Figure 2.] Recurrent Neural Network Language Model

$w(t)$: input vector

$s(t)$: hidden layer

$y(t)$: output layer

U : word representation

W : $s(t)$, $s(t-1)$ 의 관계

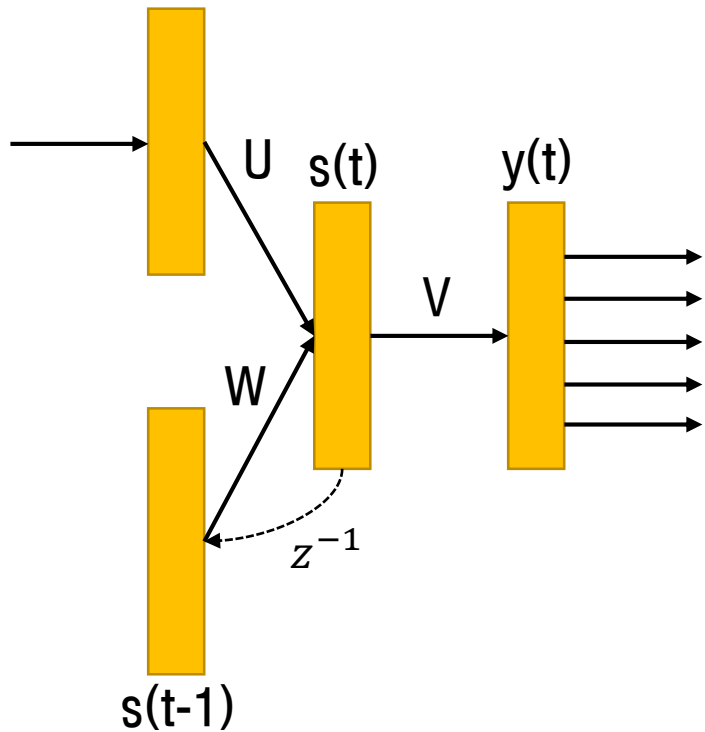
V : 특징 값

$$s(t)=f(Uw(t)+Ws(t-1))$$

$$y(t)=g(Vs(t))$$

$$f(z)=\frac{1}{1+e^{-z}} \quad , \quad g(z_m)=\frac{e^{z_m}}{\sum_k e^{z_k}}$$

Recurrent Neural Network Model



[Figure 2.] Recurrent Neural Network Language Model

$w(t)$: input vector
 $s(t)$: hidden layer
 $y(t)$: output layer
 U : word representation
 W : $s(t)$, $s(t-1)$ 의 관계
 V : 특징 값

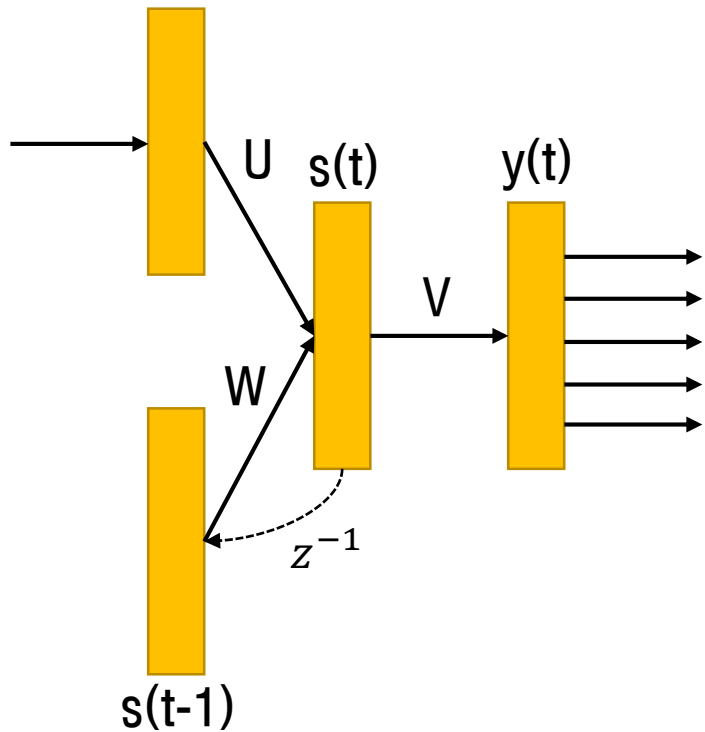
$$s(t) = f(Uw(t) + Ws(t-1))$$
$$y(t) = g(Vs(t))$$

$$f(z) = \frac{1}{1 + e^{-z}}, \quad g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

1

tanh가 아니라 sigmoid가 사용됨

Recurrent Neural Network Model



[Figure 2.] Recurrent Neural Network Language Model

2 Vectors generated by the RNN toolkit of Mikolov(2012)

$w(t)$: input vector

$s(t)$: hidden layer

$y(t)$: output layer

U : word representation

W : $s(t)$, $s(t-1)$ 의 관계

V : 특징 값

$$s(t) = f(Uw(t) + Ws(t-1))$$

$$y(t) = g(Vs(t))$$

$$f(z) = \frac{1}{1 + e^{-z}}, \quad g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

1

tanh가 아니라 sigmoid가 사용됨

Measuring Linguistic Regularity

■ A Syntactic Test Set

“a is to b as c is to ____”

위의 형태를 analogy(비유)라고 한다. B의 위치에 해당하는 형태가 ____에도 들어가야 한다.
Syntax는 한국어로 문법이고, 여기선 카테고리를 형용사, 명사, 동사로 나눔

형용사 : good-better-best (base/comparative/superlative)

명사1 : year-years (singular/plural)

명사2 : Tom-Tom' s (non-possessive/possessive)

동사 : see-saw-sees (base/past/3rd person present tense form)

Measuring Linguistic Regularity

■ A Syntactic Test Set

Category	Relation	Patterns Tested	# Questions	Example
Adjectives	Base/Comparative	JJ/JJR, JJR/JJ	1000	good:better rough:___
Adjectives	Base/Superlative	JJ/JJS, JJS/JJ	1000	good:best rough:___
Adjectives	Comparative/ Superlative	JJS/JJR, JJR/JJS	1000	better:best rougher:___
Nouns	Singular/Plural	NN/NNS, NNS/NN	1000	year:years law:___
Nouns	Non-possessive/ Possessive	NN/NN_POS, NN_POS/NN	1000	city:city's bank:___
Verbs	Base/Past	VB/VBD, VBD/VB	1000	see:saw return:___
Verbs	Base/3rd Person Singular Present	VB/VBZ, VBZ/VB	1000	see:sees return:___
Verbs	Past/3rd Person Singular Present	VBD/VBZ, VBZ/VBD	1000	saw:sees returned:___

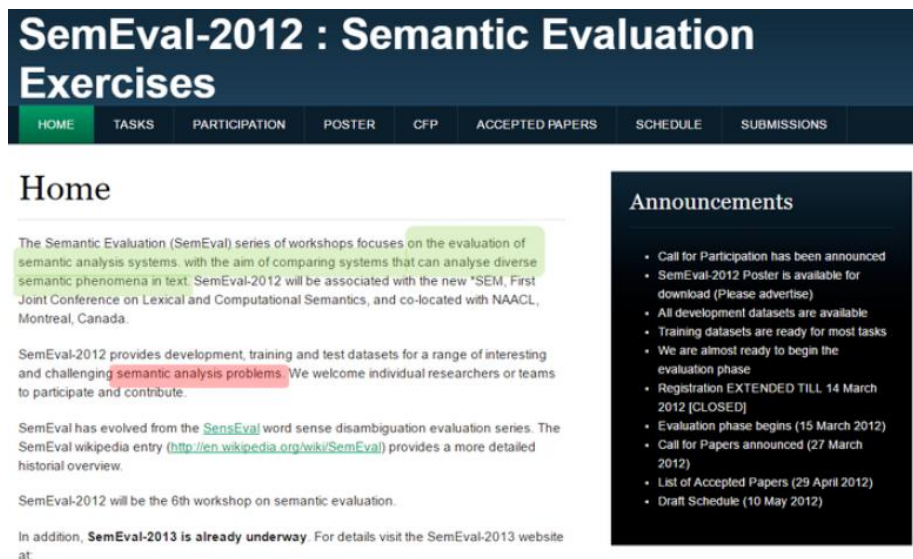
Table 1: Test set patterns. For a given pattern and word-pair, both orderings occur in the test set. For example, if “see:saw return:___” occurs, so will “saw:see returned:___”.

8개의 조합은 각각 analog하게 표현이 가능
Ex.) ‘good is better as bad is to worse.’

Measuring Linguistic Regularity

■ A Semantic Test Set

“SemEval-2012”



The screenshot shows the SemEval-2012 website. The header is dark blue with the title "SemEval-2012 : Semantic Evaluation Exercises" in white. Below the header is a navigation bar with links: HOME, TASKS, PARTICIPATION, POSTER, CFP, ACCEPTED PAPERS, SCHEDULE, and SUBMISSIONS. The main content area is divided into two columns. The left column is titled "Home" and contains text about the SemEval series, the purpose of SemEval-2012, and a link to the SemEval Wikipedia entry. The right column is titled "Announcements" and contains a list of updates, including the call for participation, poster availability, training datasets, and the registration deadline.

SemEval-2012 : Semantic Evaluation Exercises

HOME TASKS PARTICIPATION POSTER CFP ACCEPTED PAPERS SCHEDULE SUBMISSIONS

Home

The Semantic Evaluation (SemEval) series of workshops focuses on the evaluation of semantic analysis systems, with the aim of comparing systems that can analyse diverse semantic phenomena in text. SemEval-2012 will be associated with the new *SEM, First Joint Conference on Lexical and Computational Semantics, and co-located with NAACL, Montreal, Canada.

SemEval-2012 provides development, training and test datasets for a range of interesting and challenging semantic analysis problems. We welcome individual researchers or teams to participate and contribute.

SemEval has evolved from the [SenseEval](#) word sense disambiguation evaluation series. The SemEval wikipedia entry (<http://en.wikipedia.org/wiki/SemEval>) provides a more detailed historical overview.

SemEval-2012 will be the 6th workshop on semantic evaluation.

In addition, **SemEval-2013 is already underway**. For details visit the SemEval-2013 website at:

Announcements

- Call for Participation has been announced
- SemEval-2012 Poster is available for download (Please advertise)
- All development datasets are available
- Training datasets are ready for most tasks
- We are almost ready to begin the evaluation phase
- Registration EXTENDED TILL 14 March 2012 [CLOSED]
- Evaluation phase begins (15 March 2012)
- Call for Papers announced (27 March 2012)
- List of Accepted Papers (29 April 2012)
- Draft Schedule (10 May 2012)

SemEval-2012는 analog한 형태로 semantic한 관계의 word를 묶어 둔 data set으로, Recurrent Neural Network Language Model의 word vector가 의미 정보를 가지고 있는지 평가함

Measuring Linguistic Regularity

■ A Semantic Test Set

“Class : Element”

- Semantic도 syntactic처럼 analogy 형태로 표현
- Clothing과 dish가 카테고리이고, shirt와 bowl은 각각의 예시

clothing : shirt

dish : bowl



“clothing is to shirt as dish is to bowl”

Measuring Linguistic Regularity

- Syntactic, semantic linguistic regularity는 analogy 표현이 가능

Measuring Linguistic Regularity

■ Syntactic, semantic linguistic regularity는 analogy 표현이 가능



■ Analogy만 쉽게 해결하면 linguistic regularity를 identify 할 수 있다.

Measuring Linguistic Regularity

- Syntactic, semantic linguistic regularity는 analogy 표현이 가능



- Analogy만 쉽게 해결하면 linguistic regularity를 identify 할 수 있다.



새로운 vector offset method가 제안됨

Cosine Distance 기반의 Vector offset Method
=> Assume relationships are present as vector offsets.

The Vector Offset Method

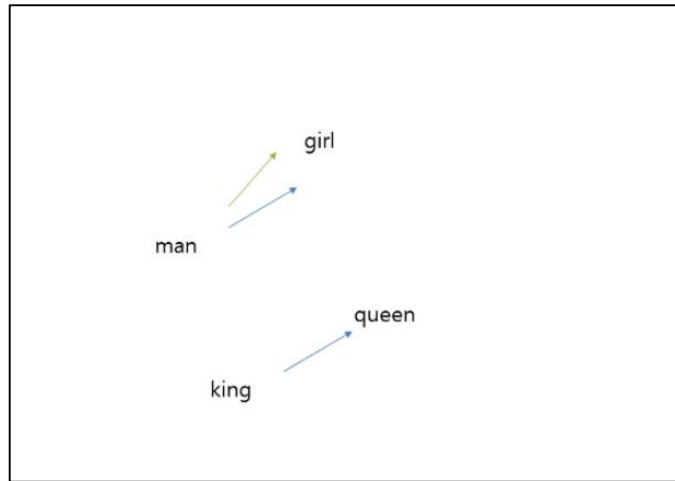
■ $a:b=c:_$ 를 구할 때, $_$ 가 unknown이면,
 $y=x_b - x_a + x_c$ 에서 y 가 결정되면 좋지만, 항상 그럴 수 없다.
그래서 가장 유사한 것을 **Cosine Similarity**를 통해 찾는다.

$$w^* = \operatorname{argmax}_w \frac{x_w y}{\|x_w\| \|y\|}$$

cosine similarity를 구하는 식으로, 이 식이 최대가 되게 하는 w 를 반환한다. 이 식이 필요한 이유는 y 가 정해지지 않을 때 가장 유사한 벡터를 찾기 위함이다.

The Vector Offset Method

■ $a:b=c:_$ 를 구할 때, $_$ 가 unknown이면,
 $y = x_b - x_a + x_c$ 에서 y 가 결정되면 좋지만, 항상 그럴 수 없다.
그래서 가장 유사한 것을 **Cosine Similarity**를 통해 찾는다.



[Figure 3.] argmax를 통해 구해진 값 예시

King-queen을 나타내는 벡터는 있는데, man-woman이 없고
Man-girl이 있을 때, argmax를 통해 구한 값인 girl이 대신 할당된다.

The Vector Offset Method

■ 성별과 명사의 단수-복수를 의미하는 Vector



[Figure 4.] argmax를 통해 구해진 값 예시

파란색 선은 성별을 의미하는 vector, 빨간색 선은 명사의 복수를 의미하는 vector이다.

Evaluating



Evaluating

■ cosine similarity를 이용한 vector offset을 통해 LSA model과 RNN model을 학습한 결과

Method	Adjectives	Nouns	Verbs	All
LSA-80	9.2	11.1	17.4	12.8
LSA-320	11.3	18.1	20.7	16.5
LSA-640	9.6	10.1	13.8	11.3
RNN-80	9.3	5.2	30.4	16.2
RNN-320	18.2	19.0	45.0	28.5
RNN-640	21.0	25.2	54.8	34.7
RNN-1600	23.9	29.2	62.2	39.6

Table 2: Results for identifying syntactic regularities for different word representations. Percent correct.

Vector의 차원을 높일수록 LSA보다 약 3배 정도 RNN의 성능이 높다.

Evaluating

■ cosine similarity를 이용하기 위해 vector offset을 생성하는 모델들의 유사 성능 비교

Method	Adjectives	Nouns	Verbs	All
RNN-80	10.1	8.1	30.4	19.0
CW-50	1.1	2.4	8.1	4.5
CW-100	1.3	4.1	8.6	5.0
HLBL-50	4.4	5.4	23.1	13.0
HLBL-100	7.6	13.2	30.2	18.7

Table 3: Comparison of RNN vectors with Turian's Collobert and Weston based vectors and the Hierarchical Log-Bilinear model of Mnih and Hinton. Percent correct.

Method	Spearman's ρ	MaxDiff Acc.
LSA-640	0.149	0.364
RNN-80	0.211	0.389
RNN-320	0.259	0.408
RNN-640	0.270	0.416
RNN-1600	0.275	0.418
CW-50	0.159	0.363
CW-100	0.154	0.363
HLBL-50	0.149	0.363
HLBL-100	0.146	0.362
UTD-NB	0.230	0.395

Table 4: Results in measuring relation similarity

CW, HLBL 모델에도 같은 실험을 진행하였는데, HLBL이 RNN과 syntactic한 측면에서 유사한 성능을 보였다.
하지만, semantic의 측면에선 CW, HLBL 둘 다 부진하고 RNN이 좋은 성능을 보였다.

Evaluating

■ cosine similarity를 이용하기 위해 vector offset을 생성하는 모델들의 유사 성능 비교

Method	Adjectives	Nouns	Verbs	All
RNN-80	10.1	8.1	30.4	19.0
CW-50	1.1	2.4	8.1	4.5
CW-100	1.3	4.1	8.6	5.0
HLBL-50	4.4	5.4	23.1	13.0
HLBL-100	7.6	13.2	30.2	18.7

Table 3: Comparison of RNN vectors with Turian's Collobert and Weston based vectors and the Hierarchical Log-Bilinear model of Mnih and Hinton. Percent correct.

Method	Spearman's ρ	MaxDiff Acc.
LSA-640	0.149	0.364
RNN-80	0.211	0.389
RNN-320	0.259	0.408
RNN-640	0.270	0.416
RNN-1600	0.275	0.418
CW-50	0.159	0.363
CW-100	0.154	0.363
HLBL-50	0.149	0.363
HLBL-100	0.146	0.362
UTD-NB	0.230	0.395

Table 4: Results in measuring relation similarity

CW, HLBL 모델에도 같은 실험을 진행하였는데, HLBL이 RNN과 syntactic한 측면에서 유사한 성능을 보였다. 하지만, semantic의 측면에선 CW, HLBL 둘 다 부진하고 RNN이 좋은 성능을 보였다.

Conclusion

Cosine Similarity를 이용한 Vector Offset Method를 사용하면,
Linguistic Regularities를 identify 하는데 더 간단하지만 비슷한 경과를 가져오고,
이 Method와 RNN을 결합하면 Linguistic Regularity를 **CPATURE**하는데 엄청난 두각을 보인다.



좋은 성능!