From frequency to meaning, Vector space models of semantics

8 Turney & Pantel, 2010

Abstract

From frequency to meaning: Vector space models of semantics (Turney & Pantel, 2010) 리뷰입니다. Neural machines 이 자연어처리 분야에서 이용되기 전, semantic 한 자연어처리를 위한 전통 방법들의 survey paper 입니다. 10 년 정도 된 논문이지만, **semantic 한 작업의 원리**를 **이해**하기에 정말 좋은 논문입니다.

From frequency to meaning?

이 논문은 **term frequency matrix** 와 같은 **vector space 에서 semantics** 를 **학습하기 위한 전통 방법**들을 소개합니다. Deep learning + NLP 의 방법을 이용하는 사람들이라 하여도 정독하면 좋을 논문입니다. 저자 중 한 명인 Peter D. Turney 는 오래전부터 sentiment classification 와 그 외의 다양한 자연어처리 연구를 하셨습니다. Sentiment analysis 를 공부할 때 한 번은 보게 되는 “Thumbs up or thumbs down?” 의 저자이기도 합니다.

이 포스트는 논문의 내용을 그대로 정리하거나 번역하지 않습니다. 이 논문에서 중요하다고 생각되는 개념들을 정리하였습니다. 이 논문을 읽기 전에, 논문에서 하려는 이야기가 무엇인지 파악하기 위하여 예고편을 보듯이 이 포스트를 읽으시길 바랍니다.

.

(1 장). Introduction

이 논문에서 이용하는 semantic 이란 단어의 의미는 일반적인 “**유사성**”의 의미입니다. Semantic web 과 같이 특정 분야에서 이용되는 semantic 이 아닙니다. 느낌적인 느낌입니다.

Vector space model 의 아이디어는 표현하고 싶은 객체를 벡터 공간의 점으로 표현하는 것입니다. **벡터 공간에서 가까운 점**들은 **semantically similar** 하다고 해석될 수 있습니다. 반대로 **서로 떨어져 있는 두 점**은 **semantically distant** 를 의미합니다. 이 문장은 term frequency representation 이나 word2vec 과 같은 word embedding 에서도 통용됩니다.

Word semantic 을 표현하는 수단 중 하나인 WordNet은 사람이 단어의 ontology 를 입력하기 때문에 이를 구축하는 비용이 많이 듭니다. 하지만 **Vector space models은 term distribution 으로 문서 간 유사성을 설명**합니다. 검색 엔진에서 **tf-idf vector 만으로도 query vector 와 document vectors 의 유사도 계산이 가능**합니다.

(2 장). Vector space models of semantics

2 장에서 semantics 을 표현하기 위한 세 가지 vector space models 을 언급합니다.

- **term-document matrix**: **for similarity of documents**
- **word-context matrix** : **for similarity of words**
- **pair-pattern matrix** : **for semantic relations**

(2.1 장). Term-document matrix

Document term matrix 는 문서 d_i 에 등장한 단어 t_j 의 빈도수 M[i,j] 를 문서 d_i 의 **단어 t_j 의 중요성으로 표현**합니다. 그리고 **문서 간 유사성**은 **문서에 등장한 단어 중요도 벡터의 유사성으로 정의**합니다. 벡터 공간이기 때문에 cosine distance 와 같은 distance metric 이 이용될 수 있습니다. 이 행렬을 transpose 하면 **단어 간 유사성**은 그 단어가 등장한 **문서로 표현**됩니다. **같은 문서에서 자주 등장**하였던 **단어는 semantically similar** 합니다.

(2.1 장). Term-document matrix

이 개념이 정교한 수학 모델을 통하여 확장된 것이 **Latent Semantic Indexing (LSI)** 이나 Latent Dirichlet Allocation (LDA) 입니다. Term-document matrix 에서는 **단어의 특징을 설명하는 descriptor (features)는 문서**입니다. 그렇기 때문에 term-document matrix 수준에서 **단어가 semantically similar** 하다는 의미는 **문서 수준에서 비슷**하다는 뜻입니다.

문서 수준에서의 semantic은 문맥 (context), 품사 (part of speech), WordNet 의 개념 (taxonomy) 보다는 **주제 (topics)** 입니다. 우리가 **두 뉴스 기사가 비슷**하다고 말할 때는, 두 뉴스 기사의 **주제가 비슷**하다는 의미입니다. **term document matrix 에서의 단어 간 similarity는 topical similarity** 입니다.

(2.2 장). Word-context matrix

(개, 고양이)나 (커피, 홍차) 와 같은 수준에서의 semantic similarity를 표현하기 위해 word-context matrix가 이용될 수 있습니다.

언어학자 Harris 의 distributional hypothesis 에 바탕을 둡니다. Word2Vec의 hypothesis 이기도 합니다. 좋은 구절이니 그대로 인용합니다.

Words that occur in similar contexts tend to have similar meanings

Contexts 는 phrases, sentences, paragraphs, chapters, documents 처럼 다양하게 정의될 수 있습니다. 언급된 순서에서 뒤쪽으로 갈수록 context window 가 넓습니다. **좁은 문맥을 이용할수록 단어의 local 한 의미가 강조**됩니다. Local information 일수록 (개, 고양이) 의 유사성에 집중할 수 있습니다. 반대로 **global context 일수록 topical similarity 를 의미**합니다.

(2.2 장). Word-context matrix

Word-context matrix는 context를 정의한 다음, 이를 **벡터처럼 취급**합니다. 문장에서 **한 단어의 앞과 뒤 2 단어를 문맥으로 이용**한다면, **단어와 문맥 간의 co-occurrence matrix**를 만들 수 있습니다. 아래 예제에서는 **sit 의 앞, 뒤에 등장한 단어와의 co-occurrence frequency 를 계산**합니다.

```
sentence = [a, cat, sit, on, the, table]
base = 'sit'
contexts = [a, cat, on, the]

matrix['sit', 'a'] += 1
matrix['sit', 'cat'] += 1
matrix['sit', 'on'] += 1
matrix['sit', 'the'] += 1
```

(2.2 장). Word-context matrix

혹은 문맥을 {a, cat}, {on, the} 이나 {a, cat, on, the} 처럼 정의할 수도 있습니다. 단, word-context co-occurrence 를 계산하는 과정에서 co-occurrence 종류가 다양하여 out of memory 가 발생하기도 합니다. 구현 시 주기적으로 infrequent patterns 을 제거하는 과정을 넣어주면 좋습니다.

벡터로 표현하려는 대상은 '단어'이며, **문맥은** 단어의 특징을 기술하는 **descriptor** 입니다. Vector space models 에서는 대상의 특징을 표현하는 정보가 무엇인지를 이해해야 합니다.

(2.3 장). Pair-pattern matrix

이 논문의 저자 중 한명인 Pantel 이 많이 연구한 방법입니다. '의사:병원' 이나 '학생:학교'와 같은 **단어 간 관계를 표현하기 위한 vector space models** 입니다. 'mason:stone' 은 'X cuts Y' 의 관계가, 'carpenter:wood' 는 'X works with Y' 의 관계가 있습니다. **Pair 는** term-document matrix 에서의 **term 에 해당**하며, **pattern 은 document 에 해당**합니다.

pair \ pattern	X cuts Y	X works with Y	...
mason:stone	53	0	...
carpenter:wood	0	224	...

(2.3 장). Pair-pattern matrix

위처럼 **pairs 와 patterns 을 표현**하면 term-document matrix 처럼 **similarity of pairs** 나 **similarity of patterns 을 표현**할 수 있습니다. mason:stone, carpenter:wood, potter:clay, glassblower:glass 는 **artisan:material 의 관계**가 있습니다. Matrix 의 rows 와 columns 을 바꾸면 (transpose), **‘the X used the Y to’** 나 **‘the X shaped the Y into’** 는 **비슷한 patterns 임을 표현**할 수 있습니다.

(2.4 장). Similarities

‘유사하다’라는 단어는 모호합니다. **(car, bicycle)은 의미적으로 비슷**합니다. Hypernym인 vehicle 을 공유합니다. **이를 taxonomically similarity**라 합니다.

(bee, honey)는 함께 등장하는 경우가 많습니다. **Semantically associated** 되어 있습니다. 그러나 **taxonomically similar 하지는 않습**니다.

‘유사성’을 표현/학습하기 이전에, 무엇을 유사하다고 말할지부터 잘 정의해야 합니다. 그리고 **그에 맞는 descriptors 를 이용**해야 합니다.

(2.5 장). Other semantic VSM

그 외의 vector space models 에 대하여 이야기합니다. 한 예로 triple pair - pattern matrix 가 언급됩니다.

이런 word-pattern representation 은 TOFLE 에서의 단어 간 관계성 문제 등을 이용하여 성능이 검증되었습니다.

(2.7 장). Hypotheses

자연어처리 분야에서의 hypotheses 에 대하여 정리하였습니다. 의미를 제대로 전달하기 위하여 원문의 정의를 인용합니다. 비슷해 보이지만 분명한 차이들이 있습니다. 이는 vector space models 로 표현하려는 대상이 무엇인지, 그리고 그 대상의 특징을 설명하는 descriptor 가 무엇인지에 따라 달라집니다.

- Statistical semantics hypothesis : If units of text have similar vectors in a text frequency matrix, then they tend to **have similar meanings**. - **텍스트 단위**가 텍스트 빈도 매트릭스에 **유사한 벡터를 가지고 있다면**, 그들은 **유사한 의미**를 갖는 경향이 있다.
- Bag of words hypothesis : **The frequencies of words in a document** tend to indicate the relevance of the document to a query - **문서 내 단어 빈도**는 **문서가 질의와 관련성**을 나타내는 경향이 있다.
- Distributional hypothesis : **Words that occur in similar contexts** tend to have **similar meanings - 비슷한 context에서 발견되는 단어들은 비슷한 의미를 갖는 경향이 있다.**
- Extended distributional hypothesis : Patterns that **co-occur with similar pairs** tend to have **similar meaning**
- Latent relation hypothesis : Pairs of words that **co-occur in similar patterns** tend to have **similar semantic relations**

위 세 가지 hypotheses 는 다른 자연어처리 논문들에서도 자주 볼 수 있는 개념입니다. 아래 두 hypotheses 는 Pantel의 연구의 철학입니다.

(3 장). Linguistic processing for vector space models

텍스트를 vector space models 로 표현하기 위한 과정은 세 가지 모듈로 구성되어 있습니다.

첫번째 과정은 tokenization 입니다. 토크나이징은 문장을 token stream 으로 표현하는 것이며, token 의 정의는 문제에 따라 달라집니다. 자세한 설명은 이전 포스트를 참고하세요.

두번째 과정은 normalization 입니다. 이 과정에서는 **영어의 stemming 이나 대/소문자의 통일** 등이 포함됩니다.

세번째 과정은 annotation 입니다. 목적에 따라서 단어의 **sense tagging, part of speech tagging**, 혹은 **구문 분석을 위한 dependency parsing** 등이 이용될 수 있습니다. 특히 word-context matrix 로 표현하기 위해서는 tagging 과정이 큰 도움을 줄 수 있습니다.

(4 장). Mathematical processing for vector space models

(4.1 장). **Building the frequency matrix**

(4.2 장). **Weighting the elements**

(4.3 장). **Smoothing the matrix**

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

토크나이징이 된 텍스트는 일반적으로 **frequency matrix 로 표현**됩니다. 이 때 weights 를 조절하거나 sparsity 의 문제를 해결하기 위하여 matrix smoothing 을 할 수 있습니다. 이 논문에서의 matrix smoothing 은 SVD 와 같은 차원축소의 방법을 의미합니다.

tf-idf 는 모든 문서에 빈번히 등장하는 단어의 영향력을 줄입니다. tf-idf 은 정보 검색 (information retrieval) 분야에서 제안된 개념입니다. 모든 문서에서 등장한 단어는 query 에 포함되어 있어도 큰 신경을 쓸 필요가 없는 단어일 가능성이 높습니다. 특별한 경우를 제외하고는 'a, the, i, am' 과 같은 단어는 무시하여도 됩니다.

Length normalization 을 합니다. 혹은 cosine distance 에는 length normalization 과정이 포함되어 있습니다. 두 벡터의 내적을 계산한 뒤, 각 벡터의 norm 으로 나누기 때문입니다. 물론 similarity 를 계산하는 과정 내내 두 벡터의 norm 을 계산하지 않으려면 데이터의 모든 벡터의 L2 normalization 을 하여 unit vector 로 만든 뒤, cosine similarity 대신 inner product 만 해도 됩니다.

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

Frequency vector 를 이용하면 문서의 길이에 따라서 영향력이 달라지기 때문에 length normalization 이 필요합니다. 하지만, 반드시 L2 normalization 이 옳다고는 생각하지 않습니다. 예를 들어 네 단어로 구성된 문장과 스무 단어로 구성된 문장의 벡터에서의 각 단어의 weights 크기는 매우 다릅니다. Twitter 나 댓글과 같이 매우 짧은 단문과 두 세 문장이 함께 포함된 데이터셋에서는 긴 문장들이 cosine similarity 를 이용할 때 비슷한 문장의 후보에서 밀려나는 경우가 빈번합니다. 이 점도 고려해야 합니다.

Feature selection 은 weighting 의 과정으로 볼 수 있습니다. Point Mutual Information (PMI) 는 word-context matrix 에서 이용할 수 있는 feature selection 방법 입니다. PMI 의 변형인 **Positive PMI (PPMI) 역시 semantic 을 학습하기에 매우 좋은 feature selection** 입니다.

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

확률에서 두 확률 분포가 서로 독립인가 (두 확률은 서로 상관이 없는가)의 조건은 joint probability 와 각각의 marginal probability 의 곱의 비율이 1인가 입니다. 식으로는 다음처럼 표현됩니다.

$$\frac{p_{i,j}}{p_i \times p_j}$$

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

안경을 쓰는 것과 저녁을 먹는 것은 서로 상관없는 문제입니다. 전체 1200 명의 사람 중 저녁을 먹은 사람은 400 명, 1/3 입니다. 1200 명 중 안경을 쓴 사람은 300 명, 1/4 입니다. 그리고 안경을 썼으며, 저녁을 먹은 사람은 1/12 입니다. 각각의 확률의 곱과 같습니다.

다른 관점으로 살펴봅니다. 안경을 썼을 경우로 제한하면, 저녁을 먹은 사람의 비율은 100 명 / 300 명 = 1/3 입니다. 안경을 쓰지 않은 사람이 저녁을 먹은 비율은 300 / 900 = 1/3 입니다. 안경을 썼는지와 저녁을 먹었는지가 서로 상관없는 일이라면, 안경을 썼는지와 상관없이 저녁을 먹은 사람의 비율이 일정해야 합니다. 위 식은 이 말을 수식으로 표현한 것입니다.

$\frac{p_{i,j}}{p_i \times p_j}$ 은 $\frac{p_{j|i}}{p_j}$ 이기 때문

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

PMI 는 위 수식에 log 를 취합니다. log(1)=0 이므로, **서로 상관이 없는 i 와 j 의 PMI 는 0**입니다.

$$PMI_{i,j} = log\left(\frac{p_{i,j}}{p_i \times p_j}\right)$$

안경 \ 저녁	저녁을 먹었다	저녁을 안먹었다	marginal prob
안경을 썼다	100	200	$\frac{1}{4} = \frac{300}{1200}$
안경을 안썼다	300	600	$\frac{3}{4} = \frac{900}{1200}$
marginal prob	$\frac{1}{3} = \frac{400}{1200}$	$\frac{3}{4} = \frac{800}{1200}$	..

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

반대로 안경을 썼을수록 저녁을 먹을 가능성이 높다면 (서로 양의 상관성이 있다면) **PMI** 는 0 보다 큽니다.

안경 \ 저녁	저녁을 먹었다	저녁을 안먹었다	marginal prob
안경을 썼다	200	100	$\frac{1}{4} = \frac{300}{1200}$
안경을 안썼다	300	600	$\frac{3}{4} = \frac{900}{1200}$
marginal prob	$\frac{5}{12} = \frac{500}{1200}$	$\frac{7}{12} = \frac{700}{1200}$	..

$$PMI(\text{안경}+, \text{저녁}+) = log\left(\frac{\frac{200}{1200}}{\frac{200}{300} \times \frac{200}{500}}\right) = log(1.2) = 0.182$$

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

음의 상관관계가 있다면 **PMI** 는 음수입니다. 자연어처리에서의 semantic 에서는 negative correlation 에 관심이 적습니다. **PPMI** 는 0 보다 작은 **PMI** 의 값을 모두 0 으로 만들고, positive correlation 에만 초점을 둡니다.

$$PPMI_{i,j} = max(0, PMI_{i,j})$$

(4.1 장). Building the frequency matrix & (4.2 장). Weighting the elements

Word-context matrix 를 PPMI 로 표현한 뒤, 이를 matrix factorization 으로 압축하면 Word2Vec 과 비슷한 학습 결과를 얻는다는 연구도 NIPS 2014 에서 제안되었습니다. Context co-cooccurrence 를 이용하면 semantics 이 표현됩니다. 각 알고리즘별로 그 과정과 output 의 형태는 다르지만 본질은 비슷합니다.

PMI 의 단점 중 하나는 infrequent pattern 에 대해서 지나치게 민감하게 반응한다는 것입니다. p_j 의 확률이 지나치게 작으면 $p_{j|i}$ 를 나눴을 때 그 수를 매우 크게 만들 수 있습니다. 이를 해결하는 간단한 방법 중 하나는 **infrequent pattern 의 probability p_j 에 α 를** 더하는 것입니다. 이 방법은 language model 에서의 smoothing 과도 비슷합니다.

(4.3 장). **Smoothing the matrix**

문법 기능을 하거나 지나치게 빈번히 등장하는 the, have 는 semantic discrimination power 가 거의 없습니다. 아무 맥락에서나 등장하면 특정 맥락을 지칭할 수 없으니까요. Matrix 의 차원을 축소하는 방법은 이런 단어를 걸러내는 역할도 합니다. **Latent Semantic Indexing (LSI), Random Projection (RP)** 와 같은 방법들은 **차원을 축소하면서도 각 차원이 특별한 semantics 을 지니도록 학습**합니다. 이 논문에서는 차원이 **semantics 을 얻도록 학습하는 차원 축소 방법들의 장점에 대하여 정리**하였습니다.

- **latent meaning** : 차원이 축소되는 과정에서 각 차원이 특정한 의미를 지닐 수 있습니다.
- **noise reduction** : 불필요한 단어들은 하나의 차원으로 살아남지 못합니다. 혹은 모든 문맥에서 등장하는 단어의 정보 역시 사라집니다.
- **high order co-occurrence** : 두 단어가 반드시 한 문서에 등장하지 않았다 하더라도 두 단어의 유사성이 반영될 수 있습니다.
- **sparsity reduction** : sparsity 가 줄어듭니다.

(5 장), Open source VSM systems

- 이 논문이 작성된지 10 여년이 지났기 때문에 이 포스트에서 언급하지 않습니다.

(6 장). Applications

이 논문에서는 각 vector space models 이 활용될 수 있는 applications 에 대하여 정리하였습니다. 불필요하다 생각되는 항목은 제외하였습니다.

(6.1 장). **Term-document matrix**

- **document retrieval** : 질의어와 비슷한 문서를 검색합니다.
- **document clustering** : 비슷한 문서들을 하나의 집합으로 그룹화 합니다.
- **document classification** : 학습데이터의 labels 을 이용하여 테스트 문서의 label 을 판별하는 판별기를 만듭니다.
- **document segmentation** : documents 를 sections 으로 나눌 수 있습니다.
- **question answering** : Q&A 는 최근 연구들의 방법들이 이미 이전 방법들을 모두 압도해 버렸네요.

(6.2 장). **Word-context matrix**

- word similarity : 의미가 비슷한 단어를 찾습니다.
- word clustering : 의미가 비슷한 단어들을 그룹화 합니다.
- automatic thesaurus generation : WordNet 과 같은 resources 는 만드는데 많은 비용이 듭니다. 데이터 기반으로 이를 도와줄 수 있습니다.
- word sense disambiguation : WSD 는 apple (과일, 회사)와 같은 **동음이의어의 의미를 분류**합니다.
- context sensitive spelling correction : WSD 처럼 상황에 따라 다른 여러 개의 solutions 이 있을 때는 **문맥 정보를 이용**해야 합니다.
- semantic role labeling : labeling parts of sentences
- query expansion : 검색 과정에서 sparsity 문제를 해결하기 위하여 query vector 를 확장합니다.
- information extraction : 장소, 사람 이름과 같은 단어의 종류를 판단하는 named entity recognition (NER) 역시 문맥 정보를 이용합니다.

(6.3 장). **Pair-pattern matrix**

- **pattern similarity** : X acts Y, X roles Y 처럼 templates 의 유사성을 정의합니다.
- **analogical mapping** : korea:seoul = japan: ? 의 답을 찾습니다.
- **etc**: relational similarity, relational clustering, relational classification

Conclusion

이 논문에서 기억해야 할 점은 **벡터 공간은 비슷한 points 는 semantically similar** 합니다.

분석의 목적에 따라 객체 (예를 들어 단어)를 표현하기 위한 descriptor 를 어떻게 설정할 것이냐를 잘 정의해야 합니다. Context 는 phrase, sentence, documents 등 다양하게 정의될 수 있습니다. 심지어 X eat Y 처럼 pattern 일 수도 있습니다.

PMI 와 PPMI 는 semantics 를 표현하는 유용한 방법입니다.

References

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Advances in neural information processing systems (pp. 2177-2185).

Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 417-424). Association for Computational Linguistics.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, 37, 141-188.