Co-occurrence based Keyword & related-word analysis

Hyunjoong Kim

soy.lovit@gmail.com

github.com/lovit

(Positive) Point Mutual Information

PMI, PPMI

- Word2Vec 같은 word embedding 방법이 등장하기 전부터 semantics 을 학습하기 위한 연구들이 제안되었습니다.
- Point Mutual Information (PMI) 는 bow models 같은 sparse vector representation 에서 semantics 을 학습하기 위해 이용된 방법입니다.

Point Mutual Information (PMI)

- 확률 이론에서는 두 확률이 서로 독립인지 판단하는 방법을 제공합니다.
  - 전체 공간에서의 $p(y)$ 와 x 조건에서의 $p(y|x)$ 가 같으면 x, y 는 독립입니다.

$$\frac{p(x,y)}{p(x) \times p(y)} = \frac{p(y|x)}{p(y)} = 1$$

Point Mutual Information (PMI)

- 확률 이론에서는 두 확률이 서로 독립인지 판단하는 방법을 제공합니다.
  - 두 확률이 독립이면 다음 조건이 성립합니다.

$$\frac{p(x,y)}{p(x)\times p(y)} = 1 \text{ , } \frac{p(\text{안경 } o,\text{저녁 } o)}{p(\text{안경 } o)\times p(\text{저녁 } o)} = \frac{\frac{1}{12}}{\frac{3}{12}\times\frac{4}{12}} = 1$$

	저녁을 먹었다	저녁을 먹지 않았다	Prob.
안경을 썼다	100	200	3 / 12
안경을 쓰지 않았다	300	600	9 / 12
Prob	4 / 12	8 / 12	

Point Mutual Information (PMI)

- 서로 양의 상관성이 있으면 $\frac{p(x,y)}{p(x)\times p(y)}$ 이 1보다 큽니다.

$$\frac{p(x,y)}{p(x)\times p(y)} = 1 \text{ , } \frac{p(\text{안경 } o,\text{저녁 } o)}{p(\text{안경 } o)\times p(\text{저녁 } o)} = \frac{\frac{2}{12}}{\frac{5}{12}\times\frac{3}{12}} = 1.2$$

	저녁을 먹었다	저녁을 먹지 않았다	Prob.
안경을 썼다	200	100	3 / 12
안경을 쓰지 않았다	300	600	9 / 12
Prob	5 / 12	7 / 12	

Point Mutual Information (PMI)

- PMI 는 두 경우의 상관성을 표현하는 index 입니다.

$$PMI(x, y) = \log \frac{p(x, y)}{p(x) \times p(y)}$$

  - 양의 상관관계라면 0 보다 큰 값을 반대라면 0 보다 작은 값을 지닙니다.
  - 값의 방향성에 해석력이 있습니다.

Positive PMI (PPMI)

- 자연어처리에서의 semantic 에서는 음의 상관관계에 큰 의미가 없습니다.
  - 양의 상관관계의 패턴을 강조하기 위해 0 보다 작은 값을 0 으로 변환합니다.

$$PPMI(x, y) = \max\big(0, PMI(x, y)\big)$$

Smoothing PMI

- PMI (PPMI) 는 infrequent 에 민감합니다.

$$PMI(x,y) = \log \frac{p(x,y)}{p(x) \times p(y)} = \log \frac{p(y\,|x)}{p(y)}$$

  - $p(y)$ 가 지나치게 작으면, 대부분의 y 가 x 에서 발생할 가능성이 있습니다.

Smoothing PMI

- 한 가지 해결책으로 $p(y)$ 에 일정한 값 α 를 더합니다.
  - $p(y\,|x)$ 가 α 이상인 경우에만 $PMI(x,y)$ 를 계산하는 효과가 있습니다.
  - α 는 threshold 역할을 합니다.

$$PMI(x,y) = \log \frac{p(x,y)}{p(x) \times (p(y) + \alpha)} = \log \frac{p(y\,|x)}{p(y) + \alpha}$$

Smoothing PMI

- α 역시 x 에 따라 다르게 적용되야 합니다.
  - x 에 따라 $\log\frac{p(y\,|x)}{p(y)+\alpha}$ 의 경향은 달라집니다.
  - Universal parameter α 는 없습니다.

$$PMI(x,y) = \log\frac{p(x,y)}{p(x)\times(p(y)+\alpha)} = \log\frac{p(y\,|x)}{p(y)+\alpha}$$

Defining contexts

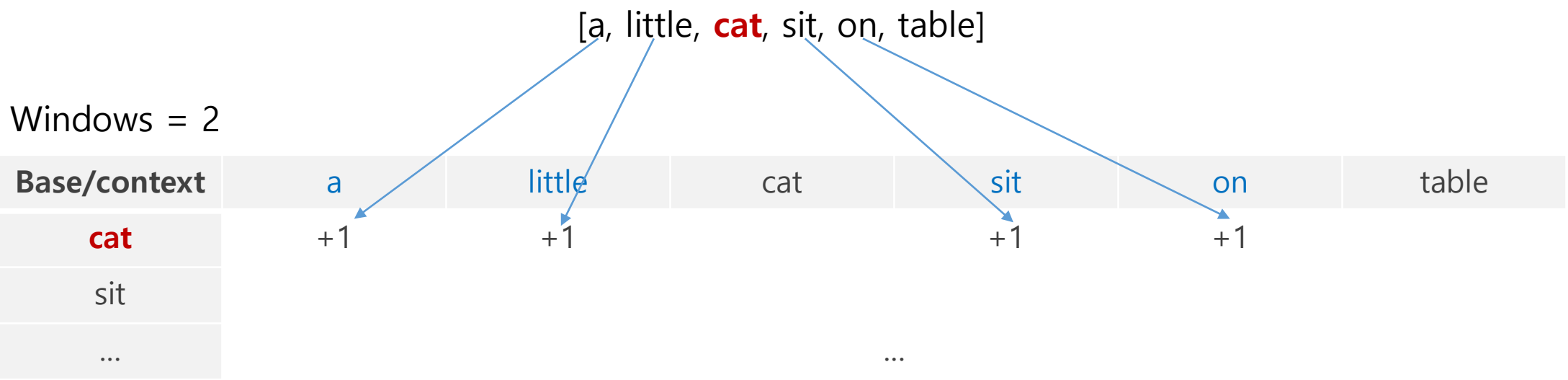
- Semantic 을 표현할 대상과 이를 설명하는 정보들을 설정합니다.
  - x 는 semantic 을 표현할 대상입니다.
  - y 는 x 를 설명하는 정보 (context) 입니다.

$$PMI(x, y) = \log \frac{p(y|x)}{p(y)}$$

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

Defining contexts

- (term, context terms) 을 (x, y) 로 표현할 수 있습니다.



Defining contexts

- (term, context terms) 을 (x, y) 로 표현할 수 있습니다.

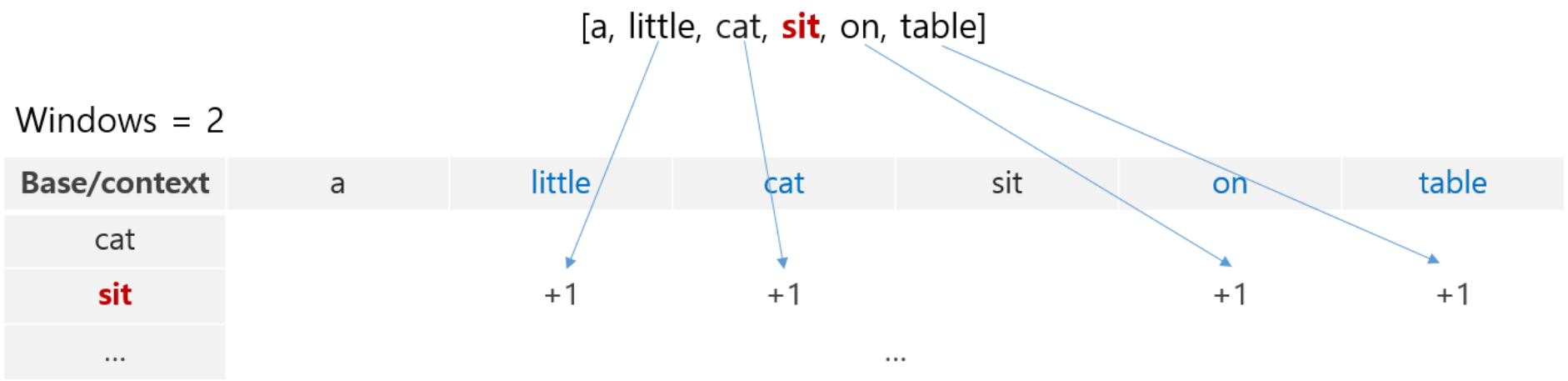
[a, little, cat, **sit**, on, table]

Windows = 2

Base/context	a	little	cat	sit	on	table
cat						
sit		+1	+1		+1	+1
…				…		

Defining contexts

- (word – context) pair 은 Word2Vec 의 개념과 유사합니다.
  - Levy & Goldberg (2014, NIPS) 에서 Word2Vec 은 PMI 행렬에 차원축소 기법을 적용한 것과 비슷하다는 사실이 밝혀졌습니다.



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

Defining contexts

- (term, context terms) 을 (x, y) 로 표현할 수 있습니다.

from message = ['지금', '어디', '야']
response = ['신도림', '이야']

Base/context	지금	어디	야	신도림	이야	…
지금	↑					
어디				↑		
야						

Packages

- https://github.com/lovit/soynlp : PMI computing modules

```
from soynlp.word from pmi
from soynlp.word from sent_to_word_context_matrix

# create word - context matrix
x = sent_to_word_context_matrix(sents, windows=3, min_tf=10, tokenizer=lambda x:x.split(), verbose=True)

# computing pmi score
pmi_dok_matrix = pmi(x, min_pmi=0, verbose=True)
```

Packages

- https://github.com/lovit/soynlp : PMI computing modules

```
from soynlp.word from PMI

pmi_extractor = PMI(windows=3, min_tf=10, verbose=True,
    tokenizer=lambda x:x.split(), min_pmi=0, alpha=0.0001)

pmi_extractor = pmi_extractor.train(corpus)
pmi_extractor.most_similar_words(query)
pmi_extractor.most_related_contexts(query)
```

Keyword with Proportion ratio

키워드란?

- 키워드 개념적으로는 이해되지만, 명확히 정의되지 않았습니다.
  - 키워드를 추출하기 전에, **키워드가 무엇인지 정의부터** 해야 합니다.
  - 자주 등장한 단어는 키워드가 아닐 수 있습니다.
    - 뉴스에서는 '오늘', '뉴스', '기자' 라는 단어는 늘 등장합니다.

키워드란?

- 키워드 관련 논문들에서 공통적으로 언급되는 키워드의 기준입니다.
  - Saliency (coverage)
    - 한 집합을 대표하는 키워드는 그 집합의 문서에 자주 등장합니다
  - Distinctiveness (discriminative power)
    - 키워드를 이용하면 그 집합과 다른 집합을 구분할 수 있습니다.

키워드란?

- 키워드를, “한 관점에서 유독 자주 등장하는 단어“로 정의할 수 있습니다.

  - (예시) “오늘”의 키워드, “인물 별” 키워드

  - (예시) 여름 철 평상시 뉴스에서 ‘폭우’가 0.1% 등장합니다. 하지만 오늘 뉴스에서는 ‘폭우’가 1% 등장하였다면, 평상시보다 10 배 더 언급된 단어입니다. ‘폭우’는 오늘 뉴스의 키워드라 할 수 있습니다.

상대적 출현 비율을 이용한 키워드 추출

- 한 단어를 기준으로, 관심있는 문서 집합 (D_T)에서의 단어 등장 비율과 비교 대상 문서 집합 (D_R) 에서의 등장 비율을 이용하여 키워드 점수를 정의합니다.

$$score_{keyword}(w) = \frac{P(w|D_T)}{P(w|D_T) + P(w|D_R)}$$

$P(w|D_T)$: 단어 w 의 관심있는 문서 집합에서의 등장 비율

$P(w|D_R)$: 단어 w 가 비교 문서 집합 (평상시)에서의 등장 비율

상대적 출현 비율을 이용한 키워드 추출

- 한 단어를 기준으로, 관심있는 문서 집합 (D_T)에서의 단어 등장 비율과 비교 대상 문서 집합 (D_R) 에서의 등장 비율을 이용하여 키워드 점수를 정의합니다.

$$score_{keyword}('폭우') = \frac{P('폭우'|D_T)}{P('폭우'|D_T) + P('폭우'|D_R)}$$

$$= \frac{1\ \%}{1\ \% + 0.1\ \%} = \frac{1}{1.1} = 0.909$$

상대적 출현 비율을 이용한 키워드 추출

- 제안된 방법은 해석력이 있습니다.
  - 제안된 점수는 [0, 1] 범위 안의 keyword score 를 가질 수 있습니다.

$$score_{keyword}(w) = \frac{P(w|D_T)}{P(w|D_T) + P(w|D_R)}$$

단어 w 가 D_T 에만 등장한 경우	단어 w 가 D_T 와 D_R 에 동일하게 등장한 경우	단어 w 가 D_R 에만 등장한 경우
$\frac{0.01}{0.01 + 0} = 1$	$\frac{0.01}{0.01 + 0.01} = 0.5$	$\frac{0}{0 + 0.01} = 0$

상대적 출현 비율을 이용한 키워드 추출

- 레이블링을 할 문서 군집을 D_T 로, 그 외의 문서 집합을 D_R 로 정의합니다.
  - 앞선 방법을 이용하면 군집을 해석할 수 있는 labels 를 추출할 수 있습니다.

no.	meaning	Keywords
1	렌트카 광고	제주렌트카, 부산출발제주도, 제주신, 이끌림, 제주올레, 왕복항공, 불포함, 제주도렌트카, 064, 롯데호텔, 자유여행, 객실, 제주여행, 특가, 해비치, 제주시, 제주항, 티몬, 2박3일, 올레, 유류, 항공권, 조식, 제주도여행, 제주공항, 2인
2	중고차 매매	최고급형중고, 최고급, 프리미어, 프라임, 2011년식, YF소나타TOP, 2010년식, 풀옵션, 2011년, YF소나타PR, 1인, Y20, 2010년, 완전무사고, 판매완료, 군포, 검정색, YF쏘나타, 2011, 하이패스, 2010, 무사고, 등급, 파노라마, 허위매물
3	클래식 음악	금관악기, 아이엠, Tru, 트럼펫, 트럼, 나팔, 금관, 텔레만, Eb, 호른, 오보에, Tr, Concerto, 하이든, 협주곡, Ha, 악기, 연주하는, 오케, 오케스트라, 독주, 악장, 작곡가, 곡
4	아이비 "유혹의 소나타 "	Song, 공부할, 부른, 노래, 가사, 부르는, 가수, 보컬, 목소리, 발라드, 명곡, 신나, 들으면, 듣기, 유혹의, 앨범,아이비, 제목
5	광염 소나타 및 일제강점기 소설들	백성수, 발가락, 현진, 이광수, 김유, 자연주의, 친일, 평양, 운수, 유미, 저지르, 야성, 탐미, 김동인, 복녀, 광염, 닮았다, 사실주의, 광기, 저지, 1920, 단편소설, 범죄, 감자, 동인, 한국문학

연관어 분석

- 기준 단어가 등장한 문서 집합의 키워드는 기준 단어의 연관어입니다.

$$S(w) = \frac{P(w|D_S)}{P(w|D_S) + P(w|\widetilde{D_S})}$$

$P(w|D_S)$: 기준 단어 S가 등장한 문서 집합 D_S 에서 단어 w 의 등장 비율

$P(w|\widetilde{D_S})$: 기준 단어 S가 등장하지 않은 문서 집합 $\widetilde{D_S}$ 에서 단어 w 의 등장 비율

키워드 / 연관어 분석

- 2016-10-20, 하루치 뉴스에서의 연관어 (명사) 분석 결과 (빈도수, 점수)

Seed word: 아이오아이	엠카운트다운 (221, 1.00)	잠깐 (162, 0.99)	타이틀곡 (311, 0.99)
방탄소년단 (638, 0.98)	키미 (297, 0.98)	보컬 (155, 0.98)	에이핑크 (237, 0.98)
유정 (161, 0.98)	파워풀 (152, 0.98)	형은 (311, 0.98)	프로듀스 (185, 0.98)
샤이니 (299, 0.98)	불독 (1212, 0.98)	다이아 (182, 0.98)	음반 (204, 0.98)
컴백 (536, 0.98)	세이 (267, 0.98)	순위 (259, 0.98)	콘셉트 (320, 0.98)
멤버들 (504, 0.98)	소라 (262, 0.97)	무대 (1332, 0.97)	발랄 (250, 0.97)
언니 (172, 0.97)	진영 (304, 0.97)	뮤직 (195, 0.97)	서바이벌 (203, 0.97)
싱글 (432, 0.97)	당당 (242, 0.97)	걸그룹 (1060, 0.96)	사운드 (189, 0.96)
각오 (168, 0.96)	강렬 (352, 0.96)	101 (341, 0.96)	실감 (167, 0.96)
쇼케이스 (549, 0.95)	작사 (230, 0.95)	1위 (1357, 0.95)	데뷔 (1365, 0.95)
미니앨범 (197, 0.95)	멤버 (624, 0.95)	프로듀서 (223, 0.95)	신곡 (400, 0.94)
신인 (328, 0.94)	일산 (194, 0.94)	뉴스1스타 (357, 0.94)	롤링 (391, 0.94)

키워드 / 연관어 분석

- 2016-10-20, 하루치 뉴스에서의 연관어 (명사) 분석 결과 (빈도수, 점수)

Seed word: 트와이스	가온차트 (191, 1.00)	스트리밍 (329, 1.00)	미니앨범 (197, 1.00)
1주년 (201, 1.00)	티저 (289, 0.99)	두번째 (201, 0.99)	뮤직비디오 (553, 0.99)
타이틀 (270, 0.99)	누적 (799, 0.99)	아이돌 (301, 0.98)	유튜브 (473, 0.98)
컴백 (536, 0.98)	0시 (190, 0.98)	1위 (1357, 0.98)	프로모션 (198, 0.98)
음반 (204, 0.98)	1억 (415, 0.98)	타이틀곡 (311, 0.97)	93 (181, 0.97)
데뷔 (1365, 0.97)	코너 (239, 0.97)	맞은 (316, 0.97)	맞이 (293, 0.97)
걸그룹 (1060, 0.97)	한류 (297, 0.97)	앞둔 (370, 0.97)	판매량 (231, 0.96)
2016년 (1337, 0.96)	대만 (251, 0.96)	페스티벌 (334, 0.96)	팬들 (999, 0.96)
신곡 (400, 0.96)	아이오아이 (270, 0.96)	잠실 (188, 0.95)	24일 (1136, 0.95)
콘서트 (463, 0.95)	입증 (297, 0.95)	6개월 (382, 0.95)	음원 (318, 0.95)
4월 (823, 0.95)	콘셉트 (320, 0.95)	축하 (189, 0.95)	73 (246, 0.94)
블랙핑크 (190, 0.94)	번째 (1158, 0.94)	돌파 (569, 0.94)	문구 (152, 0.94)

키워드 / 연관어 분석

- 2016-10-20, 하루치 뉴스에서의 연관어 (명사) 분석 결과 (빈도수, 점수)

Seed word: 박근혜	수석비서관회의 (208, 1.00)	재단들 (152, 1.00)	연설문 (204, 0.99)
누구라 (178, 0.99)	불법행위 (240, 0.99)	퇴임 (188, 0.98)	엄정 (388, 0.98)
창조경제 (226, 0.98)	처벌받 (227, 0.98)	미르 (604, 0.98)	스포츠재단 (676, 0.97)
더블루케이 (194, 0.97)	최씨 (695, 0.97)	재단 (1690, 0.97)	자유학기제 (201, 0.97)
비선실세 (219, 0.97)	최순실씨 (520, 0.97)	미르재단 (247, 0.96)	게이트 (303, 0.96)
대통령 (5682, 0.96)	모녀 (223, 0.96)	행복교육 (227, 0.95)	실세 (309, 0.95)
비선 (288, 0.95)	최순실 (1318, 0.95)	의혹 (3602, 0.95)	고양 (278, 0.95)
국정 (185, 0.94)	청와대 (2112, 0.94)	지지층 (151, 0.94)	킨텍스 (332, 0.94)
체육 (221, 0.94)	재계 (152, 0.93)	민생 (164, 0.93)	2002년 (186, 0.93)
정권 (596, 0.93)	가중 (175, 0.93)	유용 (359, 0.93)	전경련 (348, 0.93)
주재 (459, 0.93)	국민들 (441, 0.93)	백승 (216, 0.92)	갤러리 (271, 0.92)
기업들 (808, 0.92)	지지율 (336, 0.92)	확산 (800, 0.91)	철저히 (327, 0.91)

키워드 / 연관어 분석

- 2016-10-20, 하루치 뉴스에서의 연관어 (명사) 분석 결과 (빈도수, 점수)

Seed word: 최순실	게이트 (303, 1.00)	정유라 (329, 1.00)	연설문 (204, 0.99)
모녀 (223, 0.99)	비선 (288, 0.99)	더블루케이 (194, 0.98)	실세 (309, 0.98)
스포츠재단 (676, 0.98)	최씨 (695, 0.98)	최경희 (223, 0.98)	이화여대 (651, 0.98)
특혜 (532, 0.98)	미르재단 (247, 0.98)	학점 (191, 0.98)	비선실세 (219, 0.98)
이대 (419, 0.97)	미르 (604, 0.97)	재단 (1690, 0.97)	정유라씨 (200, 0.97)
엄정 (388, 0.97)	사퇴 (463, 0.96)	의혹 (3602, 0.96)	누구라 (178, 0.96)
사임 (245, 0.96)	교수들 (183, 0.96)	입학 (356, 0.96)	창조경제 (226, 0.96)
최순실씨 (520, 0.95)	수석비서관회의 (208, 0.95)	총장 (1215, 0.95)	문체부 (268, 0.95)
국정 (185, 0.95)	색깔론 (160, 0.95)	침묵 (223, 0.95)	불법행위 (240, 0.95)
모금 (238, 0.95)	재단들 (152, 0.95)	처벌받 (227, 0.95)	본관 (204, 0.95)
비리 (427, 0.94)	청와대 (2112, 0.94)	박근혜 (1445, 0.94)	퇴임 (188, 0.94)
개입 (473, 0.93)	설립 (1522, 0.93)	전경련 (348, 0.93)	더블 (225, 0.93)

상대적 출현 비율 vs PMI

- 상대적 출현 비율을 이용한 방법은 PMI 와 비슷합니다.
  - 키워드 추출 방법은 (단어, 문서집합) 간의 co-occurrence 를 이용합니다.
  - 연관어 추출 방법은 (단어, 단어) 간의 co-occurrence 를 이용합니다.
- 해석력을 얻기 위해 co-occurrence 를 다른 방식으로 이용하였습니다.

LASSO regression for keyword extraction

- LASSO 선택하는 모델은 두 가지 조건을 만족합니다

  (1) 분별력이 좋으면서

  (2) 많은 문서에서 등장한 단어를 우선적으로 선택

  - 몇 번 등장하지 않은 단어는 분별력은 좋을 수 있지만, L1 cost를 높입니다
  - 동일한 분별력을 가질 때에는 좀 더 자주 등장한 단어를 선택합니다

    (문서에 대한 coverage가 높은 단어가 선택될 가능성이 더 높습니다)

LASSO regression for keyword extraction

- 문서 종류를 명확히 구분하면서도 그 종류의 문서에서 자주 등장한 단어를 키워드로 정의할 수도 있습니다
  - 변별력이 좋고 설명하려는 문서의 다수에서 등장하는 단어 집합을 이용하면, 적은 수의 단어로 해당 문서 집합을 표현할 수 있습니다.
  - 이는 LASSO가 추출하는 features와 keywords의 정의가 일치합니다

- LASSO 는 correlation 이 높은 단어셋이 있다면, 그 중 대표 단어를 선택하여 중복적인 키워를 제거합니다.
  - 하지만, [‘바락’, ‘오바마’]가 늘 함께 등장한다면, 하나만 키워드로 선택될 수 있습니다.

L1 Regularization

T11, T13, T14를 이용하면 y=1을 완벽히 인식할 수 있고,
이 문제는 {T1, T2, T3, T11, T13, T14}만 이용해도 잘 풀림

y	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
0	5	3													
0	3	2		5		1			2						
0	2	4		4											
0	5		2				4	5	3						
0	1		1		2										
0	4			1											
1	2								2					2	
1	3							5					4	4	
1	5								1	1		3			
1	1								2			2			3
1	3								4		1		2	1	1
1	2								4				1		2

Packages

- https://github.com/lovit/soykeyword : Proportion ratio for keyword

```
from soykeyword.proportion import CorpusbasedKeywordExtractor


corpusbased_extractor = CorpusbasedKeywordExtractor(min_tf=20, min_df=10, tokenizer= lambda x:x.split())
corpusbased_extractor.train(Corpus(tokenized_corpus_fname))
lassobased_extractor.extract_from_word('아이오아이', minimum_number_of_keywords=30)
```

```
[KeywordScore(word='아이오아이', frequency=270, score=1.0),
 KeywordScore(word='엠카운트다운', frequency=221, score=0.997897148491129),
 KeywordScore(word='펜타곤', frequency=104, score=0.9936420169665052),
 KeywordScore(word='잠깐', frequency=162, score=0.9931809154109712),
 KeywordScore(word='엠넷', frequency=125, score=0.9910325251765126),
 KeywordScore(word='걸크러쉬', frequency=111, score=0.9904705029926091),
 KeywordScore(word='타이틀곡', frequency=311, score=0.987384461584851),
 KeywordScore(word='코드', frequency=105, score=0.9871835929954923),
 KeywordScore(word='본명', frequency=105, score=0.9863934667369743),
 … ]
```

Packages

- https://github.com/lovit/soykeyword : Lasso for keywords

```
from soykeyword.lasso import LassoKeywordExtractor


lassobased_extractor = LassoKeywordExtractor(min_tf=20, min_df=10)
lassobased_extractor.train(x, index2word)
lassobased_extractor.extract_from_word('아이오아이', minimum_number_of_keywords=30)
```

```
[KeywordScore(word='너무너무너무', frequency=86, coefficient=3.8159005957233778),
 KeywordScore(word='선의', frequency=40, coefficient=3.2584820410431181),
 KeywordScore(word='산들', frequency=90, coefficient=2.4407245228574896),
 KeywordScore(word='엠카운트다운', frequency=221, coefficient=1.7601587420428146),
 KeywordScore(word='챔피언', frequency=105, coefficient=1.4864913827165669),
 KeywordScore(word='사나', frequency=46, coefficient=1.4183641861333143),
 KeywordScore(word='드림', frequency=119, coefficient=1.3338856375792103),
 KeywordScore(word='뮤직', frequency=195, coefficient=1.1767179765646125),
 KeywordScore(word='먹고', frequency=216, coefficient=1.1632972589808017),
 KeywordScore(word='완전체', frequency=77, coefficient=1.121112062888608),
 KeywordScore(word='일산', frequency=194, coefficient=0.96056172786240313),
…
```