A Structured Self-Attentive Sentence Embedding

모두의연구소 풀잎스쿨 NLP Bootcamp
정미연

목 차

A Structured Self-Attentive Sentence Embedding

1.구조화된
2.스스로 집중하는 매커니즘으로
3.문장 임베딩 할거야
4.그럼 좀 더 잘될거야!

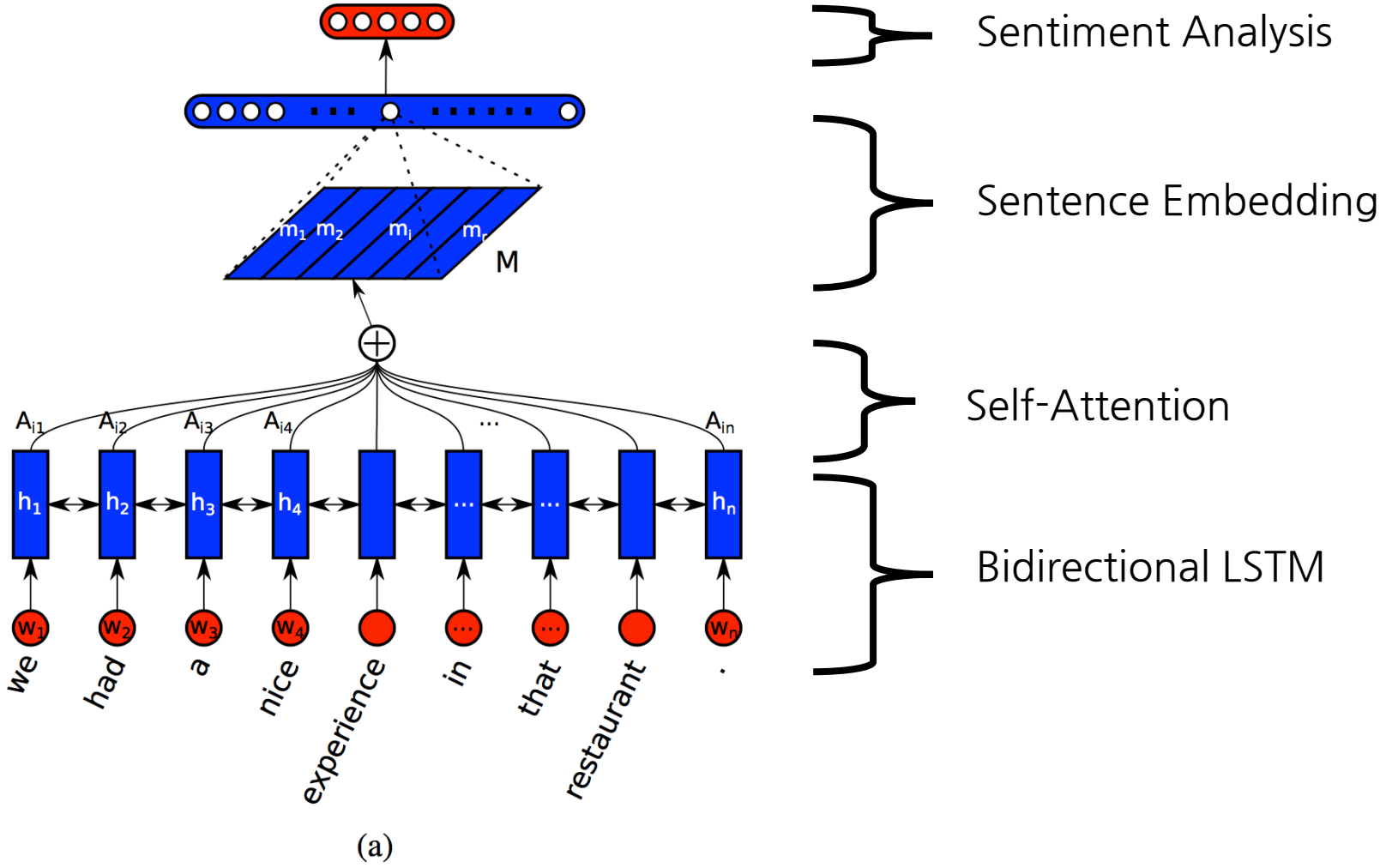
A Structured Self-Attentive Sentence Embedding

기존에는 RNNs의 final hidden states 혹은
Convolved N-grams에서 도출된 Max/Average Pooling 값으로
Simple Vector Representation을 하고 있었다

하지만, 우리 기법을 쓰면!

1. 문장을 여러 벡터의 표현으로 추출이 가능하다
2. LSTM에서 발생하는 long-term memorization burden이 줄어든다
3. (예상치도 못했지만 게다가) 추출된 임베딩을 해석하는게 쉽고 명백해진다

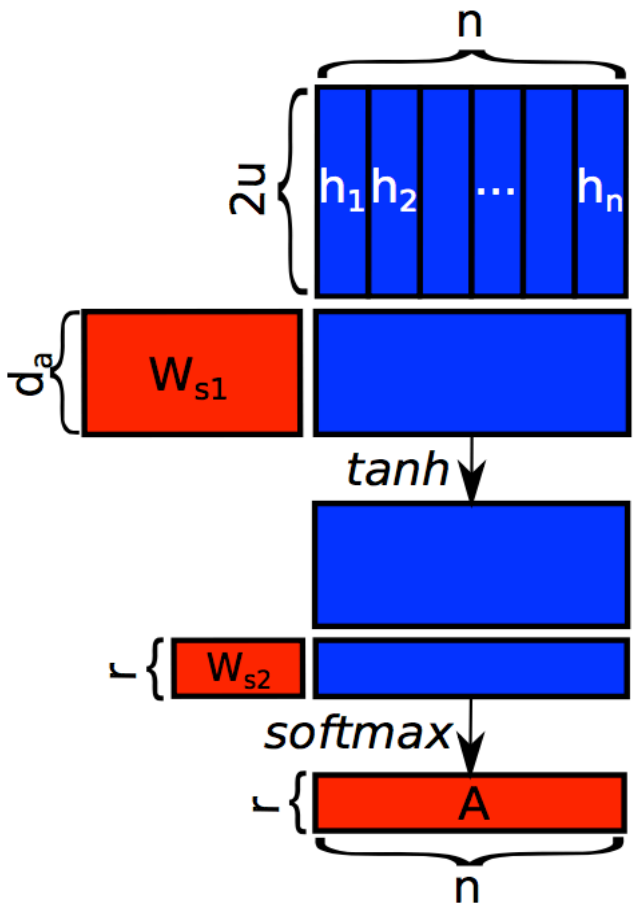
모델의 전반적인 구조



(a)

[Figure 1] A sample model structure (for Sentiment analysis)

Self-Attention Mechanism은 이렇게 합니다.



(b)

Blue colored : Hidden representations
Red colored : weights, annotations, input/output

[Figure 2] Specific Diagram Sentence Embedding from a bidirectional LSTM (weights) $(A_{i1}, ..., A_{in})$

Model Structure

$$S = (\mathbf{w_1}, \mathbf{w_2}, \cdots \mathbf{w_n}) \quad (1)$$

*S*는 문장 임베딩 벡터 (n x d)
*w*는 d차원의 워드 임베딩 벡터

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, \overrightarrow{h_{t-1}}) \quad (2)$$

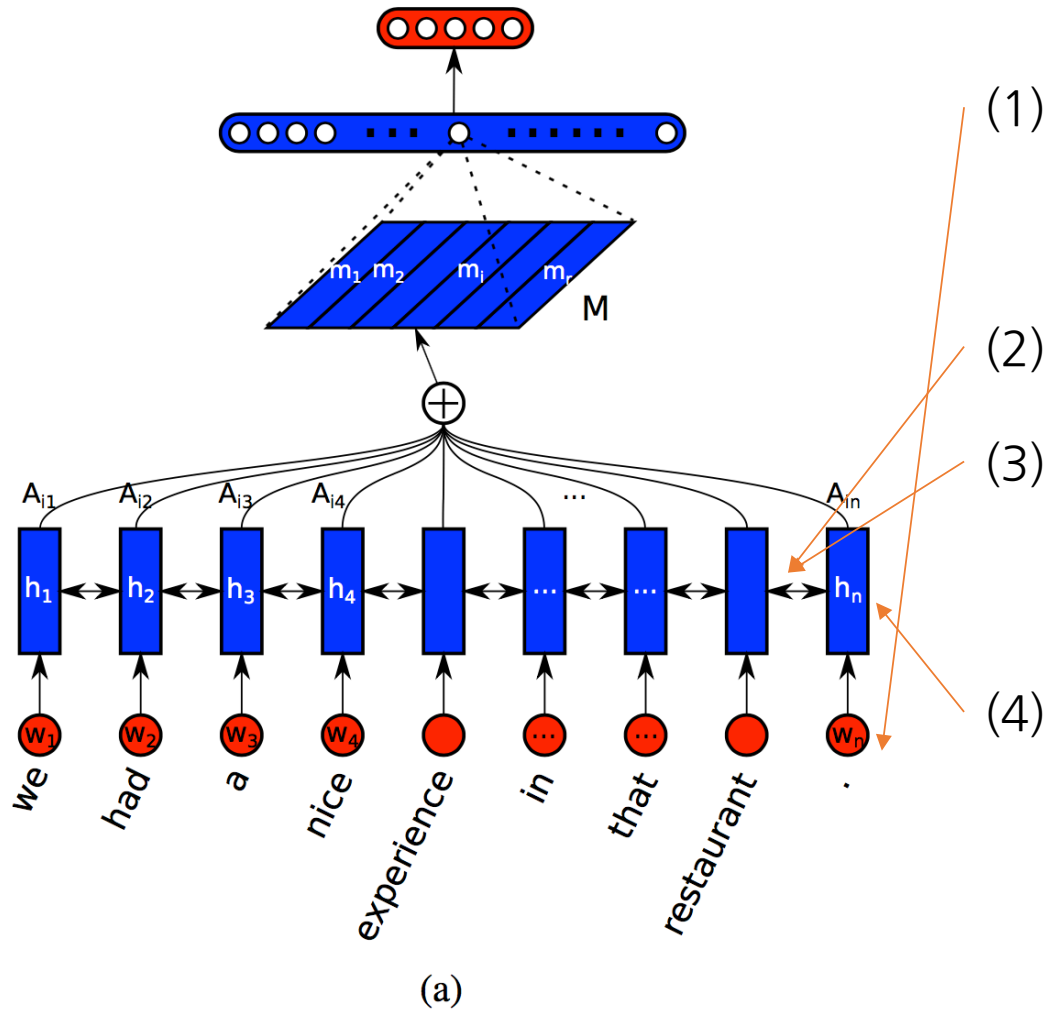
$$\overleftarrow{h_t} = \overleftarrow{LSTM}(w_t, \overleftarrow{h_{t+1}}) \quad (3)$$

Bidirectional LSTM
한 문장 내에서 인접한 단어 사이의 Dependency를 얻기 위해 양방향 LSTM을 사용

$$H = (\mathbf{h_1}, \mathbf{h_2}, \cdots \mathbf{h_n}) \quad (4)$$

(2),(3)번을 실행해서 합치면 Hidden State(h_t)를 얻음
*H*는 Hidden state의 나열 (n x 2u)

Model Structure



*S*는 문장 임베딩 벡터 (n x d)
*w*는 d차원의 워드 임베딩 벡터

Bidirectional LSTM
한 문장 내에서 인접한 단어 사이의
Dependency를 얻기 위해 양방향 LSTM을 사용

(2),(3)번을 실행해서 합치면
Hidden State(h_t)를 얻음
*H*는 Hidden state의 나열 (n x 2u)

1. Model

Variable length sentence를 고정된 사이즈로 임베딩 해야하는데, 이때 Self-attention을 사용합니다

$$\mathbf{a} = softmax\left(\mathbf{w_{s2}} tanh\left(W_{s1} H^T\right)\right) \qquad (5)$$

a → annotation vector
H → n x 2u
W_{s1} → d_a x 2u
W_{s2} → d_a

*n : 단어의 갯수
**2u: 각 히든 유닛의 차원

Self-attention 과정 (1)

$$\mathbf{a} = softmax\left(\mathbf{w_{s2}} tanh\left(W_{s1} H^T\right)\right)$$

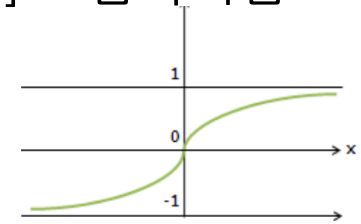
* d_a는 d차원. 우리가 정해야 하는 하이퍼파라미터

(5-1) $W_{s1} H^T$

1. W_{s1} 는 d_a x 2u 형태의 학습이 필요한 weights matrix
2. H는 n x 2u의 워드 임베딩 Hidden vector의 나열
3. 얼마나 Hidden vector를 압축할지 고민

(5-2) $tanh(W_{s1} H^T)$

1. (5-1)에 tanh를 적용하여 결과값을 [-1, 1]로 압축시킴
2. 비선형적이고 값의 범위도 잘 제한된다



(5-3) $W_{s2} tanh(W_{s1} H^T)$

1. W_{s2} 는 d_a 형태의 학습이 필요한 weights matrix
2. (5-3) 값은 길이가 n인 벡터의 길이가 나온다

* (5-1), (5-2), (5-3)는 실제 논문에는 없습니다. 이해를 돕기 위한 표식입니다 😁

내용 인용 : http://keunwoochoi.blogspot.com/2018/08/structured-self-attentive-sentence.html

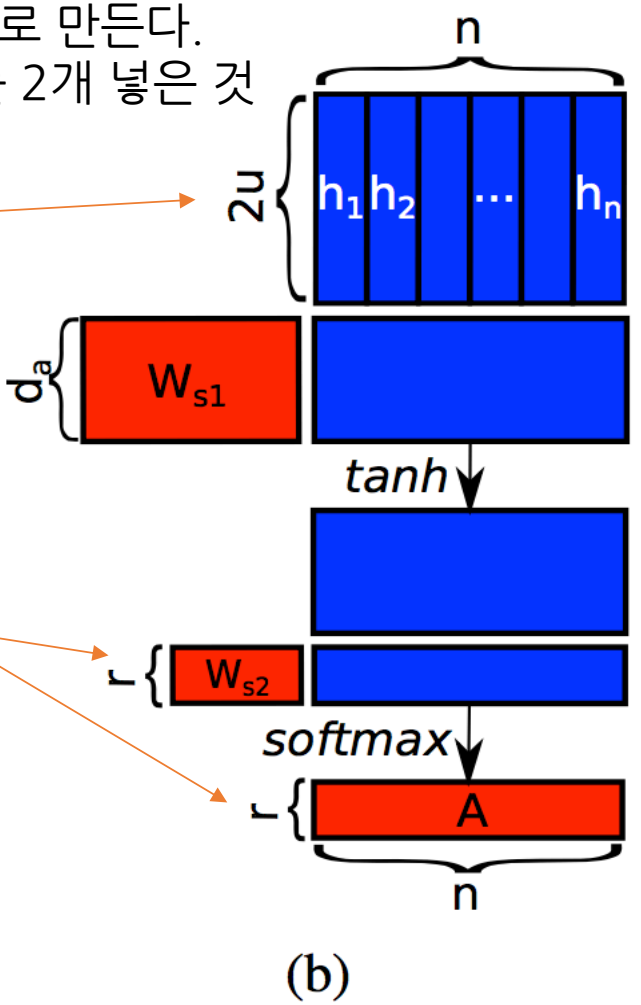
Self-attention 과정 (2)

$$A = softmax\left(W_{s2} tanh\left(W_{s1} H^T\right)\right) \quad (6)$$

softmax함수를 적용해서
계산된 가중치 전부의 합을 1로 만든다.
이때 편향없는 MLP 레이어를 2개 넣은 것

A → annotation matrix
H → n x 2u
W_{s1} → d_a x 2u
W_{s2} → d_a

*n : 단어의 갯수
**2u: 각 히든 유닛의 차원



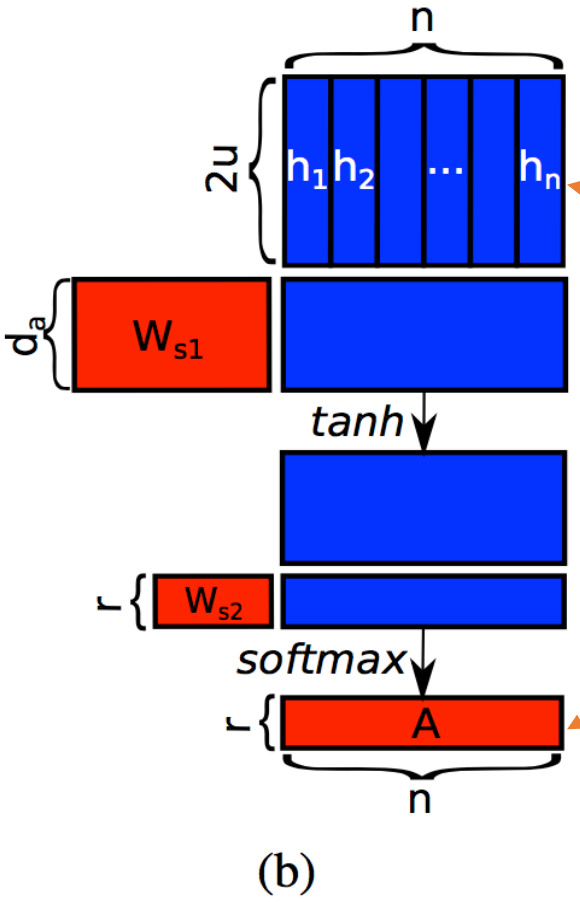
Annotation a랑 A는 뭐가 달라요? -> 벡터 / 행렬식

-> a가 여러번 필요하게 되면 A로 산출한다.
벡터값 w_{s2}값을 넣으면 (5)번식, 행렬 W_{s2}를 넣으면 (6)번식

내용 인용 : http://keunwoochoi.blogspot.com/2018/08/structured-self-attentive-sentence.html

Model Overview

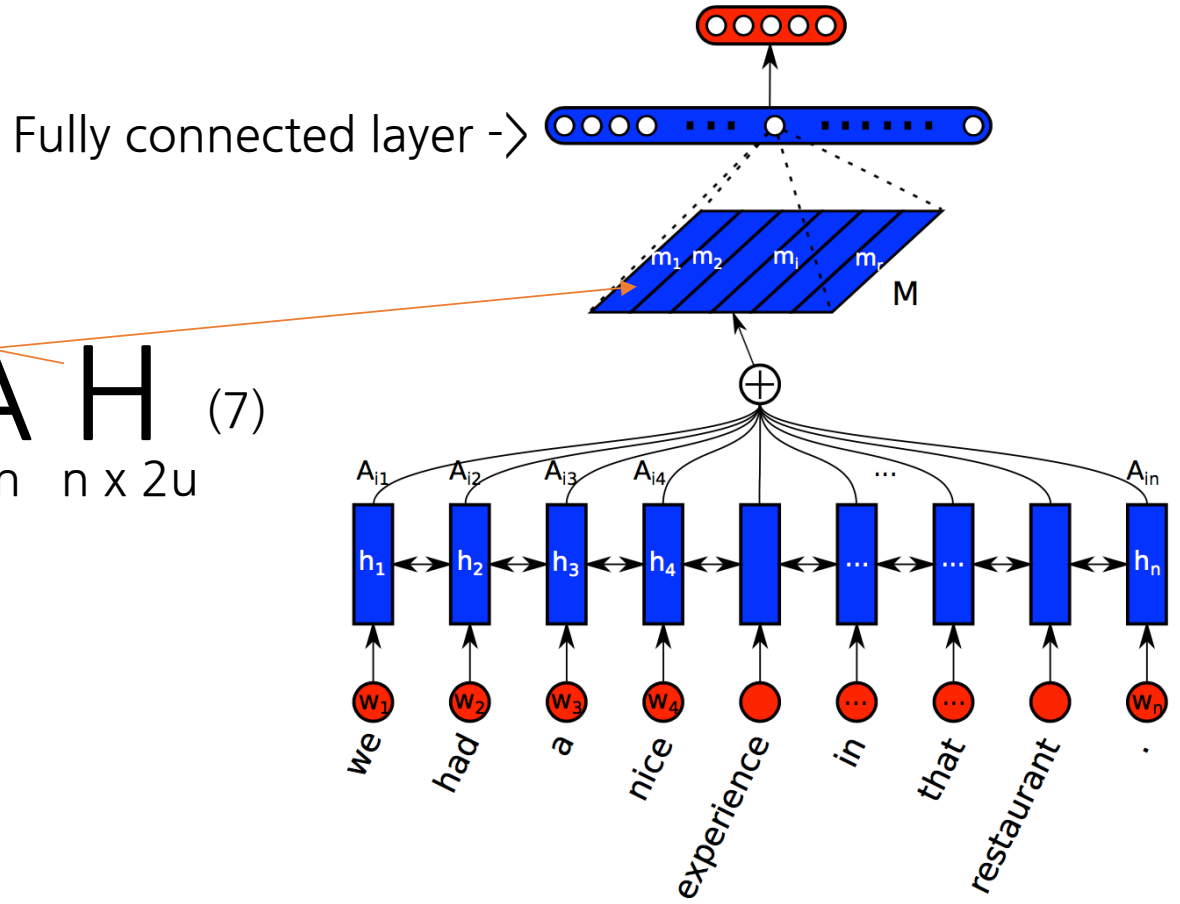
Embedding vector *m*는 *r* x 2*u*인 임베딩 매트릭스 *M*이 됨.
가중치 합 r은 Annotation matrix A와 LSTM Hidden States H의 곱으로 계산이 가능



(b)

$$M = A H \quad (7)$$

r x 2u r x n n x 2u



(a)

2. Penalization Term

Penalty Term의 필요성

- Attention Mechanism이 모든 *r* hops에 대해서 항상 비슷한 합산 가중치를 제공하게 된다면, Embedding Matrix M는 **중복성 문제**때문에 어려움을 겪을 수 있습니다.

- 그래서, **가중치 벡터 합의 Diversity를 높여줄만한 페널티가 필요**합니다.

- Diversity를 측정하는 방법은 보통 쿨백-라이블러 발산(KullBack-Leibler Divergence)을 이용합니다.

- 하지만 우리 모델에는 별로 안정적이지는 않습니다. 우리는 쿨백-라이블러 집합을 최대화 시키는데 Annotation Matrix A가 다른 softmax 출력유닛에서의 충분히 작거나 0의 값을 많이 갖도록 최적화하고 있으며 이 엄청난 양의 0이 트레이닝을 불안정하게 합니다.

- 각 행이 어떤 '의미'에 집중하기를 원합니다.

KL Divergence $$D_{KL}(P||Q) = E_{X\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = E_{X\sim P}\left[\log P(x) - \log Q(x)\right]$$

[정보이론] 두 확률분포의 차이를 계산하는 데 사용하는 함수.
EX) 우리가 가지고 있는 데이터의 분포 P(x)P(x)와 모델이 추정한 데이터의 분포 Q(x)Q(x) 간에 차이

쿨백 라이블러 개념 인용 : https://ratsgo.github.io/statistics/2017/09/22/information/

2. Penalization Term

그래서 전용 Penalty Term를 만들었습니다,

- **A와 A^T의 내적에서 단위행렬을 뺀 것을 중복성의 척도로 사용**합니다. $P = \left\| \left(AA^T - I \right) \right\|_F^2$

- 이 방법은 KL Divergence 연산보다 1/3 걸립니다!

- P는 계수형태로 곱해져서 원래 손실과 함께 최소화하는 패널티 텀으로 사용합니다.

- A 내의 벡터값을 다 합치면 1입니다. 이산확률분포에서의 확률질량으로 간주됩니다.
이는 두 분포의 요소곱(element-wise)으로도 대응이 가능합니다.

- 확률 분포 a^i와 a^j가 있고, 각 벡터의 k번째 요소를 a^i_k, a^j_k라고 정의했을 때

$$0 < a_{ij} = \sum_{k=1}^{n} a_k^i a_k^j < 1$$

분포가 다르다 — 분포가 비슷하다

- AA^T에서 항등행렬을 뺌으로써 대각 원소들을 근사적으로 1로 가깝게 만드는데 이는 합 벡터는 가능한 적은 수에 집중하게 합니다. 다른 요소들은 0으로 설정되어 서로 다른 합벡터 사이에 중복성을 부여합니다.

시각화도 함께 제공합니다.

- Sentence Embedding의 해석은 꽤 직설적인데 Annotation Matrix A 때문입니다.

- 각 열 Sentence Embedding matrix M 마다 대응하는 Annotation vector a^i들을 갖게 됩니다.

- 이 벡터들의 각 요소들은 해당 위치에 있는 **토큰들의 LSTM Hidden State에 얼마나 기여하는지**를 보여줍니다.

- 우리는 Embedding Sentence M의 각 행에 대해 히트맵을 그릴 수 있습니다.
이 방법은 임베딩의 각 부분에 인코딩 된 내용에 힌트를 제공하고 추가 해석을 제공합니다.

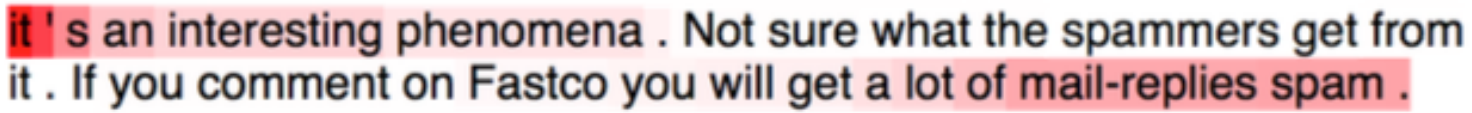
- 두 번째 방법 시각화 방법은 **모든 Annotation Vector를 합산한 다음,결과 가중치 벡터들을 합산이 1**이 되도록 정규화를 하는 방법이 있습니다.

- 문장 의미에 대한 모든 측면을 고려하기 **때문에 임베딩이 가장 어디에 중점을 두어야 하는지에 대한 일반적인 관점**을 나타냅니다. 어떤 단어가 많이 선택되는지, 어떤 단어가 스킵되는지 확인할 수 있습니다.

Penalty Term과 함께 시각화를 함께 진행해 보았습니다.

페널티를 줬을 때 중복성을 줄이기 때문에 히트맵으로 잘 보입니다.

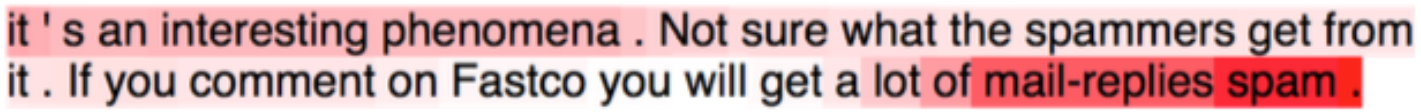
원래 문장 : 흥미로운 현상입니다. 스패머는 여기서 무엇을 얻는 지 확신을 못 합니다. Fastco에 관해 의견 주시면 당신에게 많은 스팸 메일이 발송됩니다.



(c) without penalization

어텐션 모델이 확인하는 문장단어

It', Fastco you will get a lot of Mail-replies spam.



(d) with 1.0 penalization

a lot of Mail-replies spam.
It's an interesting phenomena.
Not Sure What the spammers
get from it.
If you comment on

1. Author Profiling
2. Sentiment Analysis

Author profiling : is a method of analyzing a given number of texts to try to uncover various characteristics of the author (e.g. age and gender) based on stylistic- and content-based features.

Sentiment Analysis : aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event

실험데이터셋[1](Age) : 영어, 스페인어, 독일어로 쓰인 트위터 데이터셋(성별, 나이(5개 범주)포함)
실험데이터셋[2](Yelp) : 외국에서의 리뷰 사이트인데, 자연어 처리를 위해 무료로 데이터셋을 제공

Models	Yelp	Age
BiLSTM + Max Pooling + MLP	61.99%	77.40%
CNN + Max Pooling + MLP	62.05%	78.15%
Our Model	**64.21%**	**80.45%**

정의 참고 : https://en.wikipedia.org/wiki/Author_profiling, Sentiment_analysis
Dataset[1] : http://pan.webis.de/clef16/pan16-web/author-profiling.html
Dataset[2] : https://www.yelp.com/dataset challenge

각 리뷰에 대해서 그럴듯하게 포인트 단어들을 찾습니다

Mediocre, suck, gross, grease, crazy, disgusting, disgust, disappointment, angry, horrible ashy, bad

- if I can give this restaurant a 0 I will we be just ask our waitress leave because someone with a reservation be wait for our table my father and father-in-law be still finish up their coffee and we have not yet finish our dessert I have never be so humiliated do not go to this restaurant their food be mediocre at best if you want excellent Italian in a small intimate restaurant go to dish on the South Side I will not be go back
- this place suck the food be gross and taste like grease I will never go here again ever sure the entrance look cool and the waiter can be very nice but the food simply be gross taste like cheap 99cent food do not go here the food shot out of me quick then it go in
- everything be pre cook and dry its crazy most Filipino people be used to very cheap ingredient and they do not know quality the food be disgusting I have eat at least 20 different Filipino family home this not even mediocre
- seriously f *** this place disgust food and shitty service ambience be great if you like dine in a hot cellar engulf in stagnate air truly it be over rate over price and they just under deliver forget try order a drink here it will take forever get and when it finally do arrive you will be ready pass out from heat exhaustion and lack of oxygen how be that a head change you do not even have pay for it I will not disgust you with the detailed review of everything I have try here but make it simple it all suck and after you get the bill you will be walk out with a sore ass save your money and spare your self the disappointment
- i be so angry about my horrible experience at Medusa today my previous visit be amaze 5/5 however my go to out of town and I land an appointment with Stephanie I go in with a picture of roughly what I want and come out look absolutely nothing like it my hair be a horrible ashy blonde not anywhere close to the platinum blonde I request she will not do any of the pop of colour I want and even after specifically tell her I do not like blunt cut my hair have lot of straight edge she do not listen to a single thing I want and when I tell her I be unhappy with the colour she basically tell me I be wrong and I have do it this way no no I do not if I can go from Little Mermaid red to golden blonde in 1 sitting that leave my hair fine I shall be able go from golden blonde to a shade of platinum blonde in 1 sitting thanks for ruin my New Year's with 1 the bad hair job I have ever have

(a) 1 star reviews

Enjoy, affordable, highlight fantastic thank, love, favorite,good, incredible, much fun, great, good, good lobster, worth, amazing

- i really enjoy Ashley and Ami salon she do a great job be friendly and professional I usually get my hair do when I go to MI because of the quality of the highlight and the price the price be very affordable the highlight fantastic thank Ashley i highly recommend you and ill be back
- love this place it really be my favorite restaurant in Charlotte they use charcoal for their grill and you can taste it steak with chimichurri be always perfect Fried yucca cilantro rice pork sandwich and the good tres lech I have had.The desert be all incredible if you do not like it you be a mutant if you will like diabeetus try the Inca Cola
- this place be so much fun I have never go at night because it seem a little too busy for my taste but that just prove how great this restaurant be they have amazing food and the staff definitely remember us every time we be in town I love when a waitress or waiter come over and ask if you want the cab or the Pinot even when there be a rush and the staff be run around like crazy whenever I grab someone they instantly smile acknowlegde us the food be also killer I love when everyone know the special and can tell you they have try them all and what they pair well with this be a first last stop whenever we be in Charlotte and I highly recommend them
- great food and good service .... what else can you ask for everything that I have ever try here have be great
- first off I hardly remember waiter name because its rare you have an unforgettable experience the day I go I be celebrate my birthday and let me say I leave feel extra special our waiter be the best ever Carlos and the staff as well I be with a party of 4 and we order the potato salad shrimp cocktail lobster amongst other thing and boy be the food great the lobster be the good lobster I have ever eat if you eat a dessert I will recommend the cheese cake that be also the good I have ever have it be expensive but so worth every penny I will definitely be back there go again for the second time in a week and it be even good ..... this place be amazing

(b) 5 star reviews

[Figure 3] Heatmap of Yelp reviews with the two extreme score.

3. Textual Entailment

Textual entailment: in natural language processing is a directional relation between text fragments. The relation holds whenever the truth of one text fragment follows from another text

실험데이터셋[3] :사람이 쓴 영어 문장들, 직접 레이블링한 데이터셋 SNLI Corpus(Bowman et al., 2015

Classification task - Entailment / Contradiction / Neutral

Model	Test Accuracy
300D LSTM encoders (Bowman et al., 2016)	80.6%
600D (300+300) BiLSTM encoders (Liu et al., 2016b)	83.3%
300D Tree-based CNN encoders (Mou et al., 2015a)	82.1%
300D SPINN-PI encoders (Bowman et al., 2016)	83.2%
300D NTI-SLSTM-LSTM encoders (Munkhdalai & Yu, 2016a)	83.4%
1024D GRU encoders with SkipThoughts pre-training (Vendrov et al., 2015)	81.4%
300D NSE encoders (Munkhdalai & Yu, 2016b)	**84.6%**
Our method	84.4%

5. Conclusion

결국 논문에서 한 일은 고정적인 크기의 matrix sentence embedding에 Self-attention을 도입, Sentence Embedding을 해석하는 방법을 제안

장점

- 다른 임베딩 모델에 비해서 **성능도 좋고**
- **장기의존성 문제**를 해결했고
- Hidden states에서 작업하지 않고 Attention만 사용하니까 **간단**(?)하고
- 길이를 고정된 크기로 인코딩이 가능하기 때문에 **활용 가능성**도 높아요.

한계점

- Downstream application에 의존해서 **비지도학습은 아직 못 합니다**. 디코딩하는 동안 나눠지고 재구성되어야 할 다른 열들을 사전에 어떻게 다르게 해줘야 할지 모릅니다.
- 비지도학습을 센텐스 임베딩 위에 얹어서 시퀀스 디코더를 하면 비지도학습은 가능하지만 **다른 디코더를 찾아봐야 하지 않을까**요?