

# An Unsupervised Neural Attention Model for Aspect Extraction

---

Aspect 추출을 위한 비지도 뉴럴 어텐션 모델

## Abstract

---

Aspect 추출의 경우 aspect based sentiment analysis 에서 중요한 일.

현존하는 aspect 추출 모델의 경우 다양한 topic model(LSA, LDA) 등을 사용하고 있다.

이러한 방법은 매우 성공적이긴 하지만 coherent, 일관되지 못하다.

이 논문에서는 일관된 aspects 들의 추출에 대한 neural 접근을 제안한다.

Neural word embedding(단어 vector화를 통해서) 의 분포를 이용하면서 일관성을 향상시킨다.

자주적으로 산출된 words들을 가정하는 topic models 들과 달리, word embedding 모델들은 embedding space 에서 유사한 contexts 에서 나타나는 단어들이 서로 가깝게 위치하게 만든다.

게다가, 논문의 저자들은 학습을 하는 동안에 덜 관계있는 단어들을 덜 강조하기 위해서 attention 모델을 사용한다. 그리고 aspect의 일관성을 더욱 개선한다.

keyword :

일관성있는 aspect 추출 시도

word embedding 사용

attention 모델로 덜 중요한 단어들을 배제

## Introduction

---

- 우리가 sentiment analysis based on aspect 를 할 때 가장 중요한 것 중 하나는 분석할 aspect 를 추출하는 것이다. 표현된 opinions (review 같은 것) 에서 entity aspect 를 추출하는 것이다. 예를 들어서 문장 "The beef was tender and melted in my mouth" 이 있다고 하자. 그럼 여기서 aspect term 은 beef 이다. 두 가지 sub-tasks들이 aspects extraction 에 사용되는데,

1. 모든 aspects term 들을 review corpus 에서 추출한다.
2. aspects term 들을 군집화해서 비슷한 의미를 가지고 있는 것끼리 카테고리화하는데, **각 카테고리가 단일 aspect 를 나타내는 범주로(군집화) 분류한다.**

하나의 군집은 여러 개의 aspect term 들로 이루어져 있지만, 군집 하나는 단일 aspect 를 나타내는 걸로 이해했다.

ex) cluster 'beef','pork', 'pasta' , 'tomato' 는 food aspect 라고 할 수 있다.

- 이전의 aspect 추출 작업들은 세 개의 접근 방식으로 분류될 수 있다.
  - Rule based
    - rule based 방식은 extracted 된 aspect term 들을 카테고리화 하지 않는다.
  - supervised
    - supervised 방식은 data anntotation 을 요구한다.
    - supervised 방식은 domain 적응 문제를 겪는다.
  - unsupervised
    - 비지도 방법들은 labeled 되지 않은 데이터를 사용할 수 있다.
- LDA
  - 최근에 비지도 extract 방식으로 LDA 가 주목받고 있다.
  - LDA는 corpus 를 topics(aspects) 들의 혼합체로 간주한다.
  - 그리고 topics들을 단어들의 혼합체로 간주한다.
  - 논문의 저자들은 LDA 에 의해서 발견된 aspects 들의 혼합체들이 corpus 를 꽤 잘 설명하는 반면에, 개별 aspects 들은 꽤 낮은 성능을 보임을 발견했다.

나는 이 말을 topics 들(aspects) 이 corpus 는 잘 설명하는 반면에 개별 topic(aspect) 자체만을 살펴보면, 그것을 구성하고 있는 단어 혹은 개념들이 그 topic or aspect 를 잘 설명하고 있지 못하다고 이해했다.
  - aspects 들은 종종 관련이 전혀 없거나 느슨하게 관련된 개념으로 구성되어져 있다.
  - 이러한 문제는 두 가지 reason 이 존재한다.
    - 전통적인 LDA 모델들은 topic 의 일관성을 유지할 수 있는 가장 중요한 자원들인 단어 동시 발생 행렬을 직접적으로 인코딩하지 않는다.
    - 각 단어가 **독립적**으로 생성된다고 가정하고, 문서 수준에서 단어 속성을 모델링하여 이러한 패턴을 암묵적으로 포착한다. (LDA 가 n 번째 단어를 생성하는 과정을 보면 왜 독립적인지 이해가 가능하다.)
    - LDA 모델은 기본적으로 document 의 topic 의 분포를 가정하는데, review document 들은 짧은 경향이 있는데, 이 때문에 topic의 분포 추정은 더 어렵다.
- novel neural approach
  - LDA 모델의 약점을 다루는 새로운 neural approach 를 제안한다.
  - 논문은 neural word embedding 을 먼저 시작한다.이 임베딩은 단어들을 mapping 하는데, 보통 embedding space 에서 같은 맥락에서 주변 지점들에서 동시 발생하는 단어들을 매핑한다.
  - 논문 저자들은 문장 내에 존재하는 word embeddings 들을 attention mechanism 을 이용해서 걸러낸다.

- 아마 attention mechanism 으로 aspect 와 관계가 깊은 word embedding 들만 걸러낸다는 말이 아닐까?
- 그리고 걸러낸 word embedding 들을 aspect embedding을 조립하기 위해 사용한다.
- 앞서 aspect 들을 모두 추출하고, 군집화한 이후에, 그 군집 하나당 single aspect 를 나타낸다고 이야기 했는데, 그 이야기가 아닐까?
- aspect embedding의 training 과정은 autoencoder 와 유사하다.  
차원 축소를 사용하는데, 임베딩된 문장(sentence)과 재구조화된 각각의 문장들 중에서 공통된 요소를 추출하기 위해 사용한다.(aspect embedding들의 선형 결합을 이용해서 )  
LSA 를 살펴보면 임베딩된 문장의 차원을 축소하면 숨겨져 있던 중요한 정보들이 나옴을 살펴봤는데, 그런 이야기를 더 발전시킨 이야기가 아닌가 싶다.
- 사용하는 attention mechanism 은 어떠한 aspect 에도 포함되지 않은 단어들을 덜 강조한다. 그리고 이는 모델이 aspect word 에 초점을 두게끔 만든다.
- 논문의 저자들은 위에서 설명한 novel neural approach model 을
- Attention-based Aspect Extraction (ABAE) 라고 칭한다.!  
LG 프로젝트 코드가 이 코드인가??싶다.

LDA based model 과 대조적으로, 우리가 제안한 방법은 명시적으로 단어 발생 통계를 word embeddings 에 인코딩한다.

## Related Work

- Aspect 를 추출하는 과정은 최초로 Rule 기반이었다.
  - HU 와 LIU(사람임) 는 빈번하게 등장하는 명사들과 명사구를 찾는 것을 통해 상품들의 특징들을 추출하기를 제안했다.
  - 그들은 또한 WordNet을 통해 opinion seed words 의 동의어 반의어를 찾음으로써 opinion term 들을 추출했다.
  - 이를 따라서(규칙 기반), 상품의 aspects들을 추출하기 위해서 수많은 방법들이 빈번한 item mining과 의존성 정보에 기반해서 제안되어왔다.
  - 이러한 방법들은 사전 지정 규칙들에 꽤 의존하고 있는데, 이 사전 지정 규칙들은 aspect terms들이 명사들의 작은 그룹으로 제한되어져 있을 때 잘 작동한다.
- 지도 학습 접근법은 일반적으로 표준 시퀀스 레이블링 문제로 aspect extraction을 모델링한다.
  - rule based 기반은 추출된 aspect term 들을 충분히 범주화 하는데 세련되지 못하다.
  - 반면에 지도학습 방법은 많은 양의 labeling 데이터를 요하게 된다.
- 비지도학습 방식으로는 label data 에 의존하는걸 피하기 위해서 topic model 이 제안되어져 왔다.
  - topic model 의 output 은 각 aspect 에 대한 단어 분산 표현이거나 랭킹이다.

- 최근의 작업들은 LDA 의 연장선이라고 할 수 있다.
- 2015년의 wang et al 은 주어진 문장의 aspect와 관련 sentiment를 동시에 추출하여 aspect 와 sentiment를 RBM에서 별도의 숨겨진 변수로 취급하는 제한된 볼츠만 기계(RBM) 기반 모델을 제안했다.
- 동시 발생 단어쌍을 산출하는 biterm topic model(BTM) 이 제안됐다.
  - 논문에서 여러 업무에 걸쳐서 ABAE와 BTM 을 비교할 예정
- Attention 메커니즘
  - 어텐션 메커니즘은 모든 정보를 사용하는 게 아니라, task 에 가장 관련이 있는 정보에 집중하여 작업을 수행한다.
  - 논문에서는 attention mechanism 을 적용
  - aspect extraction task의 비지도 학습 세팅에서, 논문에서 제안한 어텐션 기반의 실험 결과는 효과적이다.

## Model Description

---

ABAE 모델을 이 section 에서 제안한다.

궁극적인 목적은 **embedding space**에서 가장 가까운 단어들을 관찰함으로써 해석될 수 있는 각각의 **aspect**의 집합을 학습하는 것이다. (위에서 살펴봤듯이 **aspect term** 들의 군집 개체 하나를 하나의 **aspect** 로 간주한다.)

- vocabulary 안에 있는 단어  $w$ 를 feature vector  $e_w$ 에 연관짓는다.
- 우리는 word embedding을 **feature vector**를 위해서 사용한다. 왜냐하면 word embedding 들은 종종 문맥안에서 동시 발생하는 단어들을 embedding space 내에서 가깝게 위치한 point에 매핑하기 위해서 설계되어져 있기 때문이다. (word2vec)
- **word embedding matrix  $E(V \times d)$**  의 행에 일치하는 단어들과 feature vectors 들은 연관되어있다. ( $V$ 는 vocabulary size를 의미한다.)
- 다른 words 들과 똑같은 embedding space(차원 동일) 를 공유하고 있는 aspects 들의 embedding 을 얻는게 목적.
- **aspect embedding matrix  $T(k \times d)$** 를 만들 수 있는데 이는  $k \times d$ 의 shape 을 가진다.  $k$ 는 정의된 aspect 의 개수를 의미하고,  $d$ 는 차원을 의미한다. 당연히 본래 vocabulary 에서 aspect 들을 추출하는 것이기 때문에  $k$ 는  $V$ 보다 적다. ( $k < V$ )
- aspect embedding들은 aspect word들에 근사하는데 사용되어진다. 여기서 aspect word 들은 attention mechanism 에 의해서 필터링된다.
- ABAE 에 대한 각각의 input sample 은 review 문장에 있는 단어들에 대한 index들의 리스트이다. (정확하게 무슨 말이 지?) => input 이 주어지면 두 단계들이 수행된다.
  - **(sentence 1차 construct 단계)** attention mechanism을 통해서 aspect words들이 아닌 단어들에 대해 weight를 낮게 주면서 non\_aspect 단어들을 걸러낸다. 그리고 weighted 된 word embeddings 들을 통해서 문장 임베딩  $Z_s$ 를 구조화한다. => 문장 임베딩  $Z_s$  생성
  - **(sentence reconstruct 단계)**  $T$  matrix(aspect embedding들의 matrix :  $T$ )로부터의 aspect embeddings의 선형 결합으로써 문장(sentence) embedding 을 다시 구조화(re-construct) 하는 것을 시도한다. ( $r_s$  생성)

- 이 re-constructions( $r_s$ )과 차원 축소는 (ABAE가 첫번째로 필터링된 문장들의 embedding인  $Z_s$ 를 최소한의 축소를 수행한 re-construct된  $r_s$  embedding으로 전환하는걸 목표로 하는)  $k$  임베드 space 안에서(**d dimension -> k dimension reduction**) aspect 단어들의 최대한의 정보를 보존한다. ( $Z_s \rightarrow r_s$  로 re-construct 하는데, 이 작업을 통해 aspect 단어들의 최대한의 정보를 보존한다는 것 같음 : 차원 축소를 통해서..)
- 더 자세한 과정은 논문의 밑의 세부적인 설명을 들어야 하는 듯.

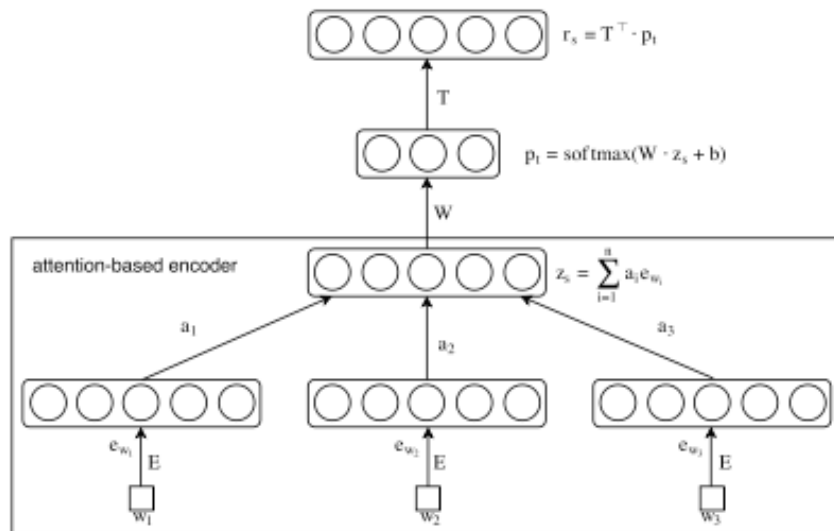


Figure 1: An example of the ABAE structure.

ABAE 의 궁극적인 목적은 aspect embedding 을 학습하는 것

1. input 이 들어오면 aspect 와 관련이 없는 단어들을 필터링
2. 필터링된 word embedding 으로부터 sentence embedding  $z_s$  를 만든다
3. aspect와 word가 같은 embedding 공간에 있도록 하기 위해서 aspect embedding 행렬  $T$ 를 이용하여 sentence embedding 을 aspect embedding으로 선형 결합으로 reconstruct 한다.

## Sentence Embedding with Attention Mechanism

- ABAE 모델은 첫 번째 step으로 input sentence ( $s$ ) 에 대해서 vector 표현  $z_s$  를 만든다.
- 

$$\mathbf{z}_s = \sum_{i=1}^n a_i \mathbf{e}_{w_i}.$$

- 논문의 저자들은 sentence의 aspect(topic) 와 최대한 관련된 정보들을 포착한 sentence의 vector 표현을 구하려고 한다.
- 이때 논문의 저자들은  $z_s$ 는 sentence 내의  $n$ 개의 문장들의 word embedding 의 가중합으로 정의한다. ( $e_{w_i}$ 의 가중합)
- 이때 가중치를  $a_i$  ( $i$ 는 sentence내의 word의 index) 라고 표현한다. 이는  $i$ 번째 단어  $w_i$ 가 sentence 의 주요 topic(aspect를 의미하는 듯)을 잘 포착하는 것에 중점을 둔 올바른 단어일 확률이다.
  - $a_i$ 는 sentence의 topic을 word가 잘 표현할 확률을 의미하는데, attention mechanism을 활용해서 계산 된다.
  - attention model 은 sentence의 전체적인 맥락 뿐만 아니라 단어  $e_{w_i}$ 에 조건을 둔 attention model 이다.
  -

$$a_i = \frac{\exp(d_i)}{\sum_{j=1}^n \exp(d_j)} \quad (2)$$

$$d_i = \mathbf{e}_{w_i}^\top \cdot \mathbf{M} \cdot \mathbf{y}_s \quad (3)$$

$$\mathbf{y}_s = \frac{1}{n} \sum_{i=1}^n \mathbf{e}_{w_i} \quad (4)$$

- 식의 4번부터 참고하면 좋다. 4-> 3 -> 2
- $y_s$ 는 word\_embeddings들의 평균 값이다. 논문 저자들은  $y_s$ 가 sentence의 권역 맥락을 가장 잘 고려했다고 믿는다.
- $M(dx \times d)$  matrix 는 권역 맥락 embedding  $y_s$  와 word embedding  $e_w$  사이를 매핑하는 matrix다. 그리고 training process 중에 학습된다. (나는 sentence 내의 word representation의 평균 값인  $y_s$ 와 sentence 내의 각각의 word embedding 과의 유사도 매핑으로 이해했다. 이게 맞나??)
- 논문의 저자들은 attention mechanism 을 두 개의 step process로 간주했다.
  - sentence 가 주어지면 sentence에 대한 표현을 sentence 내의 모든 word representation 의 평균으로 계산한다. =>  $y_s$

논문의 저자들은 aspect 를 가장 잘 표현한 sentence 의 표현을 얻고 싶기 때문에 sentence 에 대한 representation 과 sentence 내에 존재하는 word들의 representation 이 필요하다. (내적을 통한 유사성 측정을 위해) 따라서  $y_s$ 로 임의의 sentence에 대한 representation 을 설정한다.

- 그 다음 두 가지를 고려하여 word의 weight이 할당된다.
  - 첫 번째로,  $k$  aspects들과 관련있는 word를 포착할 수 있는 변형  $M$ 을 통해서 word를 걸러낸다. (aspect 과 관련있는 word 포착)
  - 그리고 문장과 걸러진 단어들의 관련성을 전역 context  $y_s$ 와 걸러진 word와의 내적을 통해서 포착한다. (aspect 와 관련있는 word와 해당 단어가 포함된 문장의 유사도 측정) => sentence 와 word의 관련성  $d_i$ 로 나타내기
  - 결국  $i$ 번째 단어의 가중치를 구하기 위해서 aspect 단어와  $i$ 번째 단어와의 관련성 그리고 문장 맥락과의 관련성을 고려한다.

- 이 유사도가 그 word 의 weight으로 간주하는 것 같다.
- 각각의 word  $w_i$  가 sentence 의 topic 과 관련되어 있을 확률을  $a_i$  로 표현한다.

## Sentence Reconstruction with Aspect Embeddings

- 위의 과정에 따라서 sentence embedding 을 얻었다 ( $z_s$ )
- 이제 sentence embedding 재구조화(reconstruction) 하는 방법을 묘사할 예정. 이 방식은 autoencoder 방식 과 유사하다.
- 논문의 저자들은 직관적으로, sentence embedding 재구조화(reconstruction) 를 T(aspect embedding matrix) 로부터의 aspect embedding 의 선형 조합으로 간주한다. (무슨말이진지는 내용을 계속 읽어봐야 될 듯 하다.)

○

$$\mathbf{r}_s = \mathbf{T}^T \cdot \mathbf{p}_t$$

- $r_s$ 는 재구조화(reconstructed) 된 vector 표현이다.
- $p_t$ 는 K aspect embedding에 대한 weighted 된 vector 이다. 각각의 weight 들은 input sentence가 관련된 aspect에 속할 확률을 나타낸다.
  - aspects들 중에서 어떤 aspect에 속하는지에 대한 weight? => input sentence가 각 aspect에 속할 확률 ( $p_t$ )
- $p_t$ 는 간단하게  $z_s$  sentence embedding을 d 차원을 k차원으로 줄이면서 얻을 수 있다. 그리고 정규화된 음수가 아닌 weights를 산출하는 softmax 비선형을 적용한다.
- $z_s$ 는 이미 attention 이 적용된 sentence 에 대한 representation. 차원은 (,d) 로 본래의  $e_{w_i}$ 의 차원과 동일하다. 왜냐하면 단어에 가중치만 곱해서 더한 것이기 때문이다.  
이때  $p_t$ 를 쉽게 얻을 수 있는 방법이 있는데  $z_s$ 의 차원 d를 정의된 aspects 들의 개수인 K로 줄이는 방법이다.

○

$$\mathbf{p}_t = \text{softmax}(\mathbf{W} \cdot \mathbf{z}_s + \mathbf{b})$$

- W는 weight 의 matrix를 의미하고 b는 bias를 의미한다. 이것들은 training 과정에서 학습이 된다.
- Aspect embedding 행렬 T(K x d) 를 설정하여 K개 aspect에 대한 aspect embedding 을 학습한다.

## Training Objective

- ABAE 는 재구조화(reconstruction) error를 최소화하기 위해 학습된다.
- 대조적인 max\_margin 목적 함수를 사용함.

- 논문의 저자들은 랜덤하게 m sentences들을 샘플링한다. => 이를 negative sample로 둔다.
- negative 샘플들의 word embedding 값을 평균낸 벡터를  $\mathbf{n}_i$ 로 표현한다.

sample 들은 word embeddings들의 모음

- ABAE의 주요 목적은 재구조화(reconstructed) 된 embedding  $\mathbf{r}_s$ 를 target sentence embedding  $\mathbf{z}_s$ 와 유사하게 만들면서도 negative sample들과는 다르게 만드는 것이다.
- 따라서 목적함수 J 는  $\mathbf{r}_s$ 와  $\mathbf{z}_s$ 의 내적값은 키우고  $\mathbf{r}_s$ 와 negative sample 간의 내적은 줄이는 방향으로 hige loss 형태가 된다.
- 

$$J(\theta) = \sum_{s \in D} \sum_{i=1}^m \max(0, 1 - \mathbf{r}_s \mathbf{z}_s + \mathbf{r}_s \mathbf{n}_i)$$

- D는 training data set을 나타내고,  $\theta$ 는 {E,T,B,W,b} 모델 파라미터를 나타낸다.

## Regularization Term

- 이 논문의 주요 목적은 review dataset 에서 가장 aspect를 잘 표현한 vector 표현을 학습하는 것.  
하지만 issue 가 발생하는데 aspect embedding matrix T는 training 동안 중복 문제를 가진다.  
aspect embedding 의 다양성을 보장하기 위해서, 논문의 저자들은 각각의 aspect embedding의 희귀성을 장려하기 위해서 목적함수 J에 정규 용어를 추가한다.

$$U(\theta) = \|\mathbf{T}_n \cdot \mathbf{T}_n^T - \mathbf{I}\|$$

- 여기서  $\mathbf{I}$  는 항등행렬이다. 그리고  $\mathbf{T}_n$ 은 1의 길이를 가지고 있는 정규화된 row의 T이다.
- $\mathbf{T}_n$ 과  $\mathbf{T}_n^T$ 는 두 개의 다른 aspect embeddings 들의 내적이다.
- U는 두 개의 다른 aspect embeddings 들의 내적이 0일 때 가장 작은 값을 가진다.
- 그래서 정규화 단어는 aspect embedding matrix T의 행들 사이의 직교성을 장려하고, 다른 aspects 사이의 중복성을 별한다.
- 마지막 목적함수 L은 위에서 구한 J와 U를 더하면서 얻어진다.
- $L(\theta) = J(\theta) + \lambda U(\theta)$
- 여기서  $\lambda$ 는 정규화 용어의 weight을 조저랴는 하이퍼 파라미터다.

## Experiment Setup



## 두 개의 데이터 셋 이용

### 1. Citysearch corpus

- restaurant review
- 50,000 review
- labeled 된 aspects들의 corpus로 부터의 3400 sentences subset 을 제공.
  - 주석이 달린 문장은 aspect 식별을 평가하는데 사용된다.
  - 6개의 정해진 aspect들이 있는데 Food, Staff, Ambience, Price, Anecdotes, Miscellaneous

### 2. BeerAdvocate

- beer review
- 1.5 만개 리뷰
- 9,245 개의 sentence들이 있는 1000개의 리뷰들은 5개의 aspect label로 주석이 달렸다.
- Feel, Look, Smell, Taste, Overall

## Baseline Methods

### 1. LocLDA

- 스탠다드한 LDA 방식을 적용.
- 전역적인 topics들의 추론을 방지하고 평가 가능한 aspect로 유도하기 위해서, 각각의 sentence들은 document 와 분리된 것으로 간주된다.

### 2. k-means

- aspect matrix T를 word embedding의 k-means centroids 를 사용하여 만든다.
- ABAE와 k-means비교

### 3. SAS

- sas는 aspect와 aspect 특화된 의견들을 동시에 발견하는 하이퍼 토픽 모델이다.
- aspect 의 의미를 발견하는데 있어서 토픽 모델들 중에서 경쟁력을 보인다.

### 4. BTM

- Social 미디어나 review site의 텍스트들과 같이 짧은 텍스트들에 맞춰 디자인된 biterm 토픽 모델이다.
- 기존 LDA의 모델에 비해 BTM 의 주요 장점은 짧은 document에서 순서되어 있지 않은 단어 쌍 동시발생 행렬을 직접 모델링 하면서 data 희소성의 문제를 완화시키는 것이다.

## Experimental Settings

### ABAE 를 위해

- word embedding matrix E를 얻기 위해서 word2vec을 사용.

embedding size 200, window size = 10, negative sample size = 5

- T matrix aspect embedding 을 처음 얻기 위해서 word embedding을 k-means로 군집화한 군집들의 중심을 사용한다.
- 다른 파라미터들은 랜덤으로 지정된다.
- 학습 기간 동안
  - word embedding matrix E를 수정한다.
  - Adam 을 통해서 다른 파라미터들을 최적화 한다. (learning rate = 0.001, 15epoch 그리고 batch size = 50)
  - negative sample의 숫자를 input sample 당 20개로 설정
  - grid search 로 직교 패널리 weight  $\lambda$ 를 1로 설정
  - 모든 모델들의 보고된 결과들은 평균적으로 10번 이상이다.
- restaurant corpus 에 대한 aspects의 숫자를 14개로 설정
- beer corpus는 다른 aspect의 숫자 10 ~ 20개를 실험. => 다른 점이 없다고 판단 14개로 설정
- 논문의 저자들은 각각의 추론된 aspect를 상위 순위 대표 단어에 따라서 gold-standard aspects 중 하나에 수동으로 매핑했습니다.
- aspect 에서 가장 대표적인 단어들은 코사인 유사도 방식을 사용해서 embedding space에서 aspect 에 가장 가까운 단어들로 발견할 수 있다.

## Evaluation and Results

---

두 개의 기준을 두고 판단.

- 의미있는 고정적인 aspect를 발견할 수 있는가?
- 실제의 review world data sets 에서 aspect 발견 performance를 향상시킬 수 있는가?

## Aspect Quality Evaluation

Inferred Aspects	Representative Words	Gold Aspects
Main Dishes	beef, duck, pork, mahi, filet, veal	Food
Dessert	gelato, banana, caramel, cheesecake, pudding, vanilla	
Drink	bottle, selection, cocktail, beverage, pinot, sangria	
Ingredient	cucumber, scallion, smothered, stewed, chilli, cheddar	
General	cooking, homestyle, traditional, cuisine, authentic, freshness	Ambience
Physical Ambience	wall, lighting, ceiling, wood, lounge, floor	
Adjectives	intimate, comfy, spacious, modern, relaxing, chic	Staff
Staff	waitstaff, server, staff, waitress, bartender, waiter	
Service	unprofessional, response, condescending, aggressive, behavior, rudeness	Price
Price	charge, paid, bill, reservation, came, dollar	
Anecdotes	celebrate, anniversary, wife, fiance, recently, wedding	Anecdotes
Location	park, street, village, avenue, manhattan, brooklyn	Misc.
General	excellent, great, enjoyed, best, wonderful, fantastic	
Other	aged, reward, white, maison, mediocrity, principle	

Table 2: List of inferred aspects for restaurant reviews (left), with top representative words for each inferred aspect (middle), and the corresponding gold-standard aspect labels (right). Inferred aspect labels (left) were assigned manually.

ABAE 가 좀더 세분화하여 aspect 에 대해서 추론한 것을 알 수 있다.(Gold Aspect 와 inferred aspects 비교)

food를 main dish 와 구분해서 dessert, drink 와 같이 구분했다.

## Coherence Score

- aspect 의 품질을 측정하기 위해서 coherence score 를 평가 지표로 사용한다. 그리고 그것은 인간이 판단한 것과 관계성을 잘 보여준다.
- z\_s aspect가 주어지고, aspect 의 top words 들 N set 이 주어지면 다음과 같은 집합이 만들어진다.

$$S^z = \{w_1^z, \dots, w_N^z\}$$

$$C(z; S^z) = \sum_{n=2}^N \sum_{l=1}^{n-1} \log \frac{D_2(w_n^z, w_l^z) + 1}{D_1(w_l^z)}$$

- 여기서  $D_1(w)$  는 word  $w$ , 단어의 문서 빈도이다.
- $D_2(w_1, w_2)$  는 word  $w_1$ 과  $w_2$ 의 문서 동시 발생 빈도이다.
- 더 높은 coherence score 는 더 나은 aspect 해석 가능성을 나타낸다.
- 논문의 저자들은 각각의 모델의 coherence score의 평균을 낸다.

각각의 model 의 coherence score 는 다음과 같이 구해진다.

$$\frac{1}{K} \sum_{k=1}^K C(z_k; S^{z_k})$$

## 실험 결과

1. ABAE 는 이전의 모델을 능가한다.
2. BTM 은 LocLDA 와 SAS보다 살짝 더 좋다.
3. 이건 아마 BTM이 직접적으로 biterm의 산출을 모델링 했기 때문일 것이다. 반면에 LDA는 단지 암묵적으로 패턴들을 포착한다. document level 부터의 word 산출을 모델링 함으로써..
4. 눈여겨볼만한 것은 k-means 군집화를 word embedding 에 하는 것은 BTM을 포함한 다른 모든 topic model 들보다 더 충분히 잘 수행한다는 점이다.
5. 이는 신경망 word embedding 이 동시 발생 행렬을 사용하는 LDA보다 더 좋다는 걸 알려준다. 심지어 동시 발생 행렬을 모델링한 BTM 보다는도 좋다.

## User Evaluation

- 논문의 저자들은 사람들이 동의할만한 aspect 의 set을 발견하기 원하기 때문에, 사람들이 직접 평가하는 것이 필요하다.

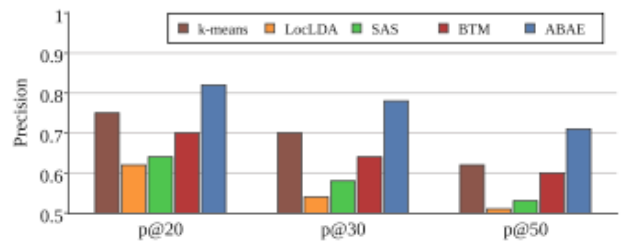
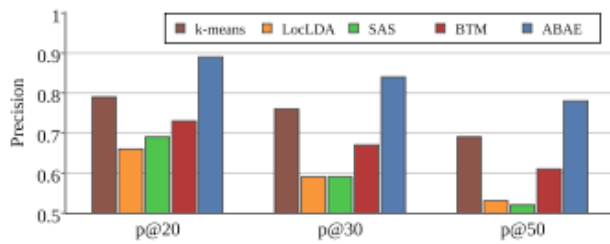
3명이 직접 해봤다.

- 만일 대부분의 심사위원들이 상위 50개의 단어들이 상품의 aspect를 일관되게 표현한다고 판단한다면 각각의 aspect는 coherent 하다고 판단된다.
- 밑에 그래프는 각각의 데이터 셋에서 각각의 방식에 따라 coherent한 aspect의 개수를 나타낸다.

	<i>k-means</i>	LocLDA	SAS	BTM	ABAE
Restaurant	11	8	9	9	11
Beer	9	8	8	9	10

- coherent aspect 를 위해서 그 aspect의 각각의 상위 단어들은 오직 대부분의 평가자들이 그 단어가 관련된 aspect를 반영한다고 판단할 때만 "correct" 로 label 되어야만 한다.
- Coherent aspect의 경우, 대다수의 평가자가 관련 aspect를 반영한다고 평가할 경우에만 각 상위 단어가 "correct" 한 걸로 분류된다.

즉, Coherent aspect 의 상위 n개의 단어를 뽑아서 그 단어들이 해당 aspect를 잘 나타낸다고 판단되면 "correct" label을 붙였다.



- ABAE 가 가장 성능이 좋다!(Precision)

## Aspect Identification

- 논문의 저자들은 이제 sentence 단계의 aspect identification 성능 주석이 달린 sentence들을 이용하여 평가한다.
- 평가 기준은 얼마나 실제 label과 매치되게 잘 예측했는지이다.

Precision, recall, f1 score로 평가된다.

- 리뷰 문장이 주어지면 ABAE 모델은 첫 번째로 가장 높은 weight의 pt와 일치하는 추론된 aspect label을 부여한다. (pt를 만들 때의 w는 그 문장이 어떤 aspect 에 속할지와 관련된 가중치이다.)

문장의 label 부여(asepct label)

- Gold-standard label을 추론된 aspects 들과 gold standard label 를 매핑한 것에 따른 문장에 부여한다.

Aspect	Method	Precision	Recall	$F_1$
Food	LocLDA	0.898	0.648	0.753
	ME-LDA	0.874	0.787	0.828
	SAS	0.867	0.772	0.817
	BTM	0.933	0.745	0.816
	SERBM	0.891	<b>0.854</b>	<b>0.872</b>
	k-means <sup>3</sup>	0.931	0.647	0.755
	ABAE	<b>0.953</b>	0.741	0.828
Staff	LocLDA	0.804	0.585	0.677
	ME-LDA	0.779	0.540	0.638
	SAS	0.774	0.556	0.647
	BTM	<b>0.828</b>	0.579	0.677
	SERBM	0.819	0.582	0.680
	k-means	0.789	0.685	0.659
	ABAE	0.802	<b>0.728</b>	<b>0.757</b>
Ambience	LocLDA	0.603	0.677	0.638
	ME-LDA	0.773	0.558	0.648
	SAS	0.780	0.542	0.640
	BTM	0.813	0.599	0.685
	SERBM	0.805	0.592	0.682
	k-means	0.730	0.637	0.677
	ABAE	<b>0.815</b>	<b>0.698</b>	<b>0.740</b>

Aspect	Method	Precision	Recall	$F_1$
Feel	<i>k</i> -means	0.720	0.815	0.737
	LocLDA	<b>0.938</b>	0.537	0.675
	SAS	0.783	0.695	0.730
	BTM	0.892	0.687	0.772
	ABAE	0.815	<b>0.824</b>	<b>0.816</b>
Taste	<i>k</i> -means	0.533	0.413	0.456
	LocLDA	0.399	<b>0.655</b>	0.487
	SAS	0.543	0.496	0.505
	BTM	0.616	0.467	<b>0.527</b>
	ABAE	<b>0.637</b>	0.358	0.456
Smell	<i>k</i> -means	<b>0.844</b>	0.295	0.422
	LocLDA	0.560	0.488	0.489
	SAS	0.336	0.673	0.404
	BTM	0.541	0.549	0.527
	ABAE	0.483	<b>0.744</b>	<b>0.575</b>
Taste+Smell	<i>k</i> -means	0.697	0.828	0.740
	LocLDA	0.651	<b>0.873</b>	0.735
	SAS	0.804	0.759	0.769
	BTM	0.885	0.760	0.815
	ABAE	<b>0.897</b>	0.853	<b>0.866</b>
Look	<i>k</i> -means	0.915	0.696	0.765
	LocLDA	0.963	0.676	0.774
	SAS	0.958	0.705	0.806
	BTM	0.953	0.854	0.872
	ABAE	<b>0.969</b>	<b>0.882</b>	<b>0.905</b>
Overall	<i>k</i> -means	0.693	0.648	0.639
	LocLDA	0.558	0.690	0.603
	SAS	0.618	0.664	0.619
	BTM	<b>0.699</b>	0.715	0.700
	ABAE	0.654	<b>0.828</b>	<b>0.725</b>

Table 5: Aspect identification results on the beer domain.

- ABAE 가 aspect `Staff` 와 `Ambience` 에서는 F1 Score 가 가장 높게 나왔다.
  - `Food` 의 f1 score 는 다른 모델에 낮게 나왔고 반면에 precision 이 높게 나왔다.
    - "The food is prepared quickly and efficiently" 는 food aspect라고 할 수 있는데, attention mechanism 을 통해서 확인해보자 "quickly" 와 "efficiently" 단어에 attention 이 집중했음을 확인했다. => 문장의 quickly와 efficiently 는 food라는 단어보다는 Staff 에 더 관련있고 더 잘 어울리는 단어이다. (sentence 가 food라는 단어가 적혀있긴 하지만, 이는 일반적으로 서비스에 대한 표현에 적합하다.)
  - 비록 *k*-means 방식과 abae 방식 둘 다 coherent aspects 들을 잘 추출하지만 abae는 실질적으로 이 task 에서 *k*-means 방식보다 더 잘 수행한다. (어텐션 덕분!)
- beer 데이터 셋에서 5개의 gold-standard-aspect label 뿐만 아니라 `Taste` 와 `Smell` 을 합성해서 하나의 aspect로 만들었다.
  - 두 aspect를 구성하는 words들이 비슷하고, 이는 사람조차 구분하기 힘들기 때문에 합쳐서 간주하는 게 더 편하다.

## Validating the Effectiveness of Attention Model

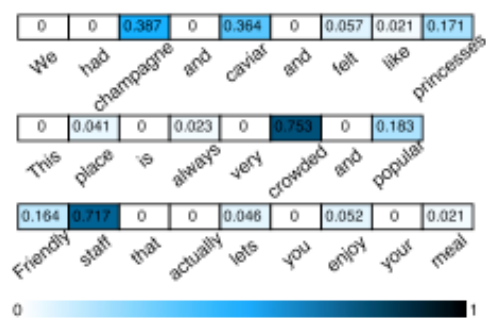


Figure 4: Visualization of the attention layer.

- attention 이 집중하고 있는 단어는 사람의 직관과 맞물려 있다.
- attention 모델이 abae 전반적인 성능에 잘 적용되고 있는지 평가하기 위해서 ABAE 모델 과 ABAE- 모델을 비교한다.

ABAE- 는 attention layer 가 꺼진 모델이고 문장 임베딩이 word embedding의 평균으로 계산된다.

$$z_s = \frac{1}{n} \sum_{i=1}^n e_{w_i}.$$

Aspect	Method	Precision	Recall	F <sub>1</sub>
Food	ABAE <sup>-</sup>	0.898	0.739	0.791
	ABAE	<b>0.953</b>	<b>0.741</b>	<b>0.828</b>
Staff	ABAE <sup>-</sup>	0.784	0.669	0.693
	ABAE	<b>0.802</b>	<b>0.728</b>	<b>0.757</b>
Ambience	ABAE <sup>-</sup>	0.782	0.660	0.703
	ABAE	<b>0.815</b>	<b>0.698</b>	<b>0.740</b>