creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사 학위논문

리뷰 기반의 어텐션 메커니즘을 활용한 추천시스템의 개선

Improving Recommender Systems with Review-based Attention Mechanism

2020년 8월

서울대학교 대학원

전기정보공학부

유성욱

리뷰 기반의 어텐션 메커니즘을 활용한 추천시스템의 개선

Improving Recommender Systems with Review-based Attention Mechanism

지도교수 심 규 석

이 논문을 공학석사 학위논문으로 제출함
2020년 5월

서울대학교 대학원
전기정보공학부
유성욱

유성욱의 공학석사 학위논문을 인준함
2020년 6월

위 원 장 김태환 (인)

부위원장 심규석 (인)

위 원 홍성수 (인)

국문 초록

추천시스템은 유저의 소비를 이끌기 위해 여러 기업에서 사용하고 있다. 초기 행렬분해법을 이용한 방식은 데이터 희소성과 콜드스타트 문제로 인해 정확도 높은 추천이 어려웠다.

최근 신경망을 기반으로 리뷰를 이용하여 데이터 희소성을 보완함과 동시에 성능을 향상하는 모델들이 등장하였다. 이 중에 어텐션 메커니즘을 활용하여 유저와 아이템에서 각각 특징 벡터를 뽑아낸 후 고차원에서 유저와 아이템 특징 벡터와 평점의 관계를 찾아내는 방식이 효과적이었다. 하지만, 어텐션 스코어를 구하는데 리뷰의 단어로부터 개별적인 유저 또는 아이템의 성향을 찾아 반영한다면 단어 수준에서 더 효과적인 어텐션 스코어를 구할 수 있다. 또한 유저는 여러 아이템에 대해 리뷰들을 쓰고, 아이템도 여러 유저들에 의해 리뷰들이 쓰인다. 이를 구분하여 유저의 선호도 혹은 아이템의 특성을 추출하는데 유용한 리뷰들을 찾을 수 있다.

따라서, 본 논문에서는 단어 수준으로 기존 어텐션 스코어 계산 방식을 변형하여 유저와 아이템 아이디에 대한 개별적 잠재 요인을 적용한 어텐션 계산방식을 사용하는 모델을 제안한다. 또한 개별적 요인이 반영된 단어 수준 어텐션 방식과 리뷰 수준으로 중요한 리뷰를 반영하는 방식을 계층적으로 융합하는 또 하나의 모델을 제안한다. 3가지 실생활 데이터에 대해 제안하는 모델의 추천 성능이 기존연구보다 우수함을 검증하였다.

주요어 : 추천시스템, 컨볼루션 신경망, 어텐션 메커니즘
학 번 : 2018-26190

목 차

제 1 장 서론 ····· 1
제 1 절 연구의 배경 및 내용 ····· 1

제 2 장 관련 연구 ····· 3
제 1 절 행렬분해법 ····· 3
제 2 절 사전학습 단어 임베딩 ····· 5
제 3 절 컨볼루션 신경망 ····· 10
제 4 절 어텐션 메커니즘 ····· 12

제 3 장 제안하는 어텐션 메커니즘 ····· 15
제 1 절 개별적 정보를 반영한 어텐션 메커니즘 ····· 15
제 2 절 리뷰 수준 어텐션 메커니즘 ····· 18

제 4 장 실험 ····· 20
제 1 절 실험 환경 및 실험 데이터 ····· 20
제 2 절 성능 측정 지표 및 실험 설정 ····· 21
제 3 절 실험 결과 및 분석 ····· 22

제 5 장 결론 ····· 25

참고문헌 ····· 26
Abstract ····· 28

표 목 차

[표 1] 실험 데이터의 통계특징 ····· 20
[표 2] 기존연구와의 평균 절댓값 오차 비교 ····· 22

그 림 목 차

[그림 1] CBOW 방식 ····· 5
[그림 2] Skip-Gram 방식 ····· 5
[그림 3] 동시 등장행렬의 예 ····· 7
[그림 4] 동시 등장행렬로부터 구한 통계정보의 예 ····· 8
[그림 5] 컨볼루션 신경망의 구조 ····· 10
[그림 6] 로컬 어텐션 스코어의 계산 과정 ····· 12
[그림 7] 상호 어텐션 메커니즘의 단어 연관성 시각화 ····· 13
[그림 8] 로컬 어텐션 스코어 계산방식 ····· 17
[그림 9] 제안하는 단어 수준 어텐션 메커니즘 ····· 18
[그림 10] 리뷰 수준 어텐션 메커니즘 ····· 19
[그림 11] 어텐션 잠재 요인 수 m에 따른 성능 비교 ····· 22
[그림 12] 단어 수준 어텐션 스코어 시각화 ····· 23
[그림 13] 평점 예측과 학습된 유저의 리뷰 중요도 ····· 23
[그림 14] 평점 예측과 학습된 아이템에 쓰인 리뷰 중요도 ····· 24

제 1 장 서론

제 1 절 연구의 배경 및 내용

한 유저의 어떤 아이템에 대한 평점은 그 유저의 선호도와 선택한 아이템의 속성을 의미한다. 협업 필터링(Collaborative Filtering, CF)[1,2]의 한 방식인 행렬분해법(Matrix Factorization, MF)[3]은 초창기 추천시스템에 많이 사용되었지만, 콜드 스타트 문제(Cold Start)와 데이터 부족(Data Sparsity) 현상으로 인해 성능이 좋지 못했다. 단순히 평점만을 이용하여 유저의 선호도와 아이템의 속성에 대한 벡터를 찾기 때문에 정보가 부족하였다. 이를 극복하기 위해 리뷰를 활용하기 시작했다. 그 중 컨볼루션 신경망(Convolutional Neural Network, CNN)을 기반으로 유저와 아이템의 리뷰에서 유저와 아이템의 분리된 네트워크에서 잠재적 특징(Latent Feature)들을 추출하고, 이를 바탕으로 행렬분해법을 하는 방식[5]이 효과적이었다.

[5]에 대한 성능향상을 위해 어텐션 메커니즘(Attention Mechanism)을 활용하여 리뷰 정보에서 각 단어별로 평점에 영향을 끼치는 정도를 구하고 이를 반영하여 유저와 아이템의 특성을 뽑아내는 방식들[6-8]이 등장하였다. 특히 [8]는 리뷰에서 단어 수준(Word-Level)으로 두 종류의 어텐션 메커니즘이 사용되었다. 로컬 어텐션(Local Attention) 방식은 슬라이딩 윈도우를 옮겨가며 그 안에서의 단어 어텐션 스코어를 계산한다. 상호 어텐션(Mutual Attention)방식은 유저와 아이템의 리뷰에서 단어 사이의 상관관계를 구하고 종합하여 단어 어텐션 스코어를 계산한다. 하지만 [8]에서 단어 수준(Word-Level)으로 로컬 어텐션 스코어를 산출하는 과정은 리뷰의 각 단어가 평점에 영향을 끼치는 정도를 서로 다른 유저 또는 아이템에 대해 구분하지 않는다. 유저와 아이템의 개별적 요인(Individual Factor)을 가정하여 아이디 별로 내포된 정보를 단어 수준(Word-Level)으로 어텐션 스코어를 계산하여 반영하는 것이 가능하다.

또한 유저의 리뷰들은 아이템마다 다르고, 아이템 리뷰들은 여러 유저

에 의해 쓰인다. 각 리뷰 별로 다른 성질을 가지고 있기 때문에 유저의 선호도와 아이템 특징 추출에 영향을 끼치는 정도가 다를 수 있다. 이를 고려하여 유저의 아이템들에 대한 리뷰 모음 혹은 아이템에 쓰인 여러 유저들의 리뷰 모음에서 중요도를 찾도록 리뷰 수준(Review-Level)의 어텐션 스코어를 계산해 반영할 수 있다.

본 논문에서는 [4]의 단어 어텐션 스코어 산출방식과 달리 학습이 가능한 개별적인 아이디 특징(Id Attention Feature)을 가정하고 이를 어텐션 스코어 계산에 반영하여 더 효과적인 어텐션 스코어를 단어 수준(Word-Level)으로 적용하는 방식을 사용하는 모델을 제안한다. 또한 유저의 아이템들에 대한 리뷰 모음 혹은 아이템에 쓰인 여러 유저들의 리뷰 모음에서 리뷰 수준(Review-Level)으로 학습하여 중요도를 반영하는 또 하나의 향상된 모델을 제안한다. 2장에서는 관련 연구 및 컨볼루션 신경망, 단어 임베딩(Word Embedding)에 대해 설명하고, 베이스라인으로 사용한 논문의 어텐션 메커니즘을 소개한다. 3장에서는 본 논문에서 개별적인 아이디 특징이 반영된 단어 수준 어텐션 스코어 산출방식과 리뷰 별 중요도를 반영하는 리뷰 수준 어텐션 스코어 산출방식과 계층적으로 융합하는 방식을 제안한다. 4장에서는 실제 데이터를 이용하여 기존 연구 모델과 제시한 모델의 성능을 비교하고 실험결과를 분석한다. 마지막으로 5장에서는 내용을 정리하고 결론을 맺는다.

제 2 장 관련 연구

제 1 절 행렬분해법

행렬분해법은 협업 필터링(Collaborative Filtering, CF)[1,2]에서 사용하는 방법이다. 추천 시스템에서의 협업 필터링은 우선 평점 정보로부터 모은 유저들의 취향 또는 아이템들의 특성을 파악한다. 그 후 한 유저가 관심을 가질만한 아이템을 유추하고, 이를 바탕으로 추천하는 기술이다.

협업 필터링의 전통적인 방법은 최근접 이웃(K-Nearest Neighbor, KNN)알고리즘을 기반으로 Top-K개의 상품을 추천한다. 유저-유저간의 유사도를 기준으로 하는 유저 기반(User Based) 방식[1]과 아이템-아이템 간의 유사도를 기준으로 하는 아이템 기반(Item Based) 방식[2]이 있다. [2]에서 두 개의 상품 i와 j간의 유사도를 구하는 방식은 피어슨 상관계수(Pearson Correlation Coefficient)를 사용하였으며 식 (2-1)를 통해 계산한다.

$$\mathrm{sim}(i,j) = \frac{\sum_{u \in U}(\mathrm{R}_{u,i} - \overline{\mathrm{R}_i})(\mathrm{R}_{u,j} - \overline{\mathrm{R}_j})}{\sqrt{\sum_{u \in U}(\mathrm{R}_{u,i} - \overline{\mathrm{R}_i})^2}\sqrt{\sum_{u \in U}(\mathrm{R}_{u,j} - \overline{\mathrm{R}_j})^2}} \quad \text{식 (2-1)}$$

여기서 U는 유저 전체 집합이다. 그리고 $\mathrm{R}_{u,i}$는 유저u의 아이템i에 대한 평점이며, $\overline{\mathrm{R}_i}$는 아이템i에 대한 평점의 평균값을 나타낸다. 이 방법은 서로 다른 아이템 간의 유사도를 사전에 구해놓고, 한 사용자가 어떤 아이템에 좋은 평점을 주었다면 그 아이템과 유사도가 높은 아이템을 추천에 이용하는 방식이다.

유사도를 계산하여 유저의 선호도와 아이템의 특성을 파악하는 방법 외에도 임의의 유저가 어떤 아이템에 대해 평점을 주었을 때, 그 행동에 영향을 끼치는 잠재적 요소(Latent Factor)가 K개 있다는 가정을 바탕으로 평점을 추론하는 행렬분해법(Matrix Factorization, MF)[3]이 있다. [3]의 우수함을 Netflix Prize에서 입증하면서 널리 사용되기 시작했다.

행렬분해법은 주어진 평점 정보로부터 구성한 평점행렬 R의 랭크를 K

로 가정하면 M명의 유저와 N개의 아이템 특징 행렬을 두 잠재행렬(Latent Matrix) $\mathbf{P} \in \mathbb{R}^{M \times N}$와 $\mathbf{Q} \in \mathbb{R}^{N \times K}$ 로 나타낼 수 있다. 위와 같은 가정을 바탕으로 $\mathbf{P}$와 $\mathbf{Q}^{\mathrm{T}}$의 곱이 평점 행렬 $\mathbf{R}$을 가장 잘 표현하도록 분해한 뒤, 다시 복원하면 식 (2-2)을 통해 평점 $r_{u,i}$을 두 잠재 벡터(Latent Vector) $\mathbf{p}_u \in \mathbf{P}$와 $\mathbf{q}_i \in \mathbf{Q}$ 사이의 내적을 통해 예측할 수 있다.

$$\widehat{r_{u,i}} = \mathbf{p}_u \mathbf{q}_i = \sum_{k=1}^{K} p_{u,k} q_{i,k} \qquad \text{식 (2-2)}$$

추천시스템의 성능을 높이기 위해서는 $\mathbf{P}$와 $\mathbf{Q}^{\mathrm{T}}$의 곱이 평점 행렬 $\mathbf{R}$을 가장 잘 표현하도록 하는 것이 중요하다. 평점 행렬과 잠재행렬과의 복잡한 연관성을 찾는 딥러닝(Deep Learning, DL) 기술을 기반으로 평점행렬 $\mathbf{R}$을 가장 잘 표현하는 $\mathbf{P}$와 $\mathbf{Q}^{\mathrm{T}}$를 찾는 연구가 이루어졌다[9]. 그런데 주어진 행렬 $\mathbf{R}$의 대부분의 평점 $r_{u,i}$는 0인 값으로 주어지는 데이터 부족 현상(Data Sparsity)이 있다. 이런 이유로 평점 정보만을 가지고 평점을 예측하는 것은 한계가 있었다.

리뷰를 바탕으로 유저가 평점을 매긴 이유를 설명할 수 있었기 때문에 데이터 부족 현상을 보완함과 동시에 추천 성능을 향상하는 방법으로 리뷰 정보를 활용하는 연구가 이루어졌다.[5-8]. 특히 [5]에서 리뷰 정보를 바탕으로 유저의 선호도와 아이템에 특성을 나타내는 벡터를 효과적으로 얻는 딥러닝 모델 연구가 이루어졌다. 한 유저의 선호도는 한 유저가 어떤 아이템에 대한 리뷰를 작성하면서 발생한 행동을 통해 파악할 수 있다. 또한, 아이템에 대한 특성 정보는 한 아이템에 쓰인 여러 유저의 리뷰 정보들로부터 얻을 수 있다. 이 가정을 바탕으로 병렬 네트워크를 사용하여 유저의 선호도와 아이템의 특성을 추출하였다. [5]를 시작으로 유저의 선호도와 아이템의 특성 벡터를 추출하고 두 네트워크의 사이에서 나오는 평점을 예측하는 방식의 연구가 활발히 이루어졌다[6-8].

제 2 절 사전 학습된 단어 임베딩

[5]에서 리뷰 정보에 Word2Vec 알고리즘[10]에 의해 사전 학습된 단어 임베딩을 사용하였다. [10]에서 신경망은 그림 1, 2와 같은 구조를 지니며 문맥적인 의미를 고려하여 말뭉치(Corpus) 내에 존재하는 여기서 말뭉치에 존재하는 원 핫 인코딩(One-hot Encoding)된 단어 벡터 x_i를 임베딩 레이어를 거쳐 d차원의 벡터 $\mathbf{w}_i \in \mathbb{R}^d$로 표현하는 기술이다. 여기서 말뭉치 안에 존재하는 모든 단어의 수는 V로 나타낸다.

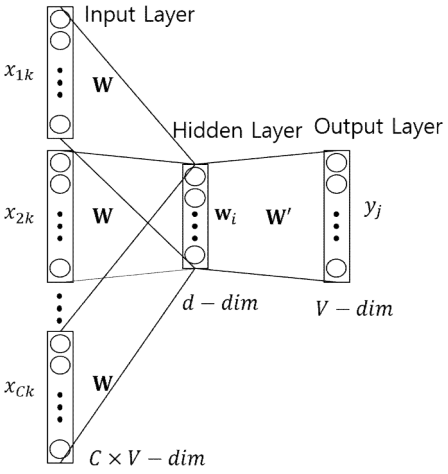

그림 1 CBOW 방식

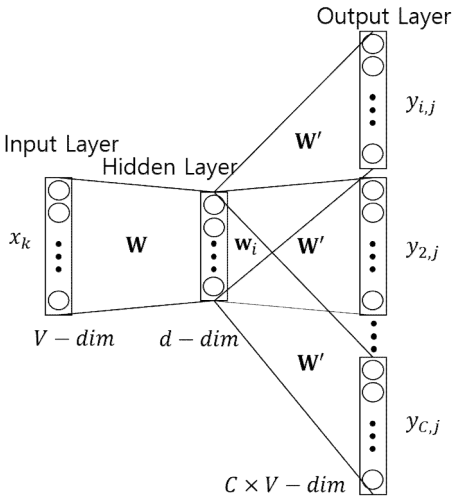

그림 2 Skip-Gram 방식

그림 1, 2의 신경망 네트워크는 주어진 말뭉치를 바탕으로 문맥적 의미가 고려되도록 학습 파라미터 $\mathbf{W} \in \mathbb{R}^{V \times d}$, $\mathbf{W}' \in \mathbb{R}^{d \times V}$가 학습되며 학습 후에는 $\mathbf{W}$와 $\mathbf{W}'$가운데 어느 것을 사용해도 무방하지만 $\mathbf{W}$를 사용하여 단어 x_i를 식 (2-3)을 통해 벡터 $\mathbf{w}_i$로 선형 변환하여 표현 한다.

$$\mathbf{w}_i = \mathbf{W}\,\mathrm{x}_i \qquad \text{식 (2-3)}$$

학습되는 과정은 윈도우 크기 w안에서 C개의 주변단어로 중심단어를 맞추는 Continuous Bag of Words, CBOW 방식과 중심단어로 C개의 주변단어를 맞추는 Skip-Gram방식이 있다. 그림 2와 같이 Skip-Gram 방식으로 학습 데이터를 구성하면 말뭉치 크기가 동일하더라도 각 단어 당 학습할 데이터의 수가 더 많기 때문에 주로 이 방식을 사용한다.

Skip-Gram에 의한 Word2Vec 알고리즘은 윈도우 크기 w에서 중심단어 x와 C개의 주변단어 $\{y_k\}_{k=1,2,..,C}$를 바탕으로 아래의 식 (2-4)을 최대화하는 방향으로 학습을 진행한다.

$$P(y_1, y_2, ..., y_C \mid x) = \prod_{k=1}^{C} P(y_k|x) \qquad \text{식 (2-4)}$$

$$P(y_k|x) = \frac{\exp(\mathbf{w}_{y_k}\mathbf{w}_x)}{\sum_{j=1}^{V}\exp(\mathbf{w}_{y_j}\mathbf{w}_x)} \qquad \text{식 (2-5)}$$

식 (2-5)는 중심단어 x가 주어졌을 때 주변단어 y_k가 나타날 확률을 뜻하며, y_k와 x의 임베딩 벡터 $\mathbf{w}_{y_k}$와 $\mathbf{w}_x$사이의 $softmax$값을 의미한다. 식 (2-5)을 최대화하는 방향으로 학습하면 임베딩 된 단어 벡터 $\mathbf{w}_x$와 $\mathbf{w}_{y_k}$간의 유사도를 높이게 되므로 문맥적인 의미가 유사한 단어의 관계를 학습할 수 있으며 중심단어 x로 예측한 주변 단어를 $\hat{y_k}$로 두면 손실 함수 J는 다음 식 (2-6)과 같다.

$$J = -\sum_{k=1}^{C}\log P(y_k|x) = -\sum_{k=1}^{C} y_k \log\hat{y_k} \qquad \text{식 (2-6)}$$

여기서 $\mathbf{W}$와 $\mathbf{W}'$는 크기가 큰 행렬이다. 단어의 수가 증가할수록 계산량이 폭증하므로 두 가지 휴리스틱(Heuristic)을 사용하였다. 첫 번째는 말뭉치에서 등장하는 단어는 학습량을 확률적인 방식으로 줄이는 방법인 서브 샘플링(Sub-Sampling Frequent Words)기법을 사용하였다. 식 (2-7)처럼 단어 빈도수에 의한 샘플링을 통해 학습량을 줄인다. 여기서 $f(y_i)$는 해당 단어 y_i가 말뭉치에 등장한 비율(해당 단어빈도/전체 단어수)이며 t는 실험적으로 얻은 값 0.00001이다.

$$P(y_i) = 1 - \sqrt{\frac{t}{f(y_i)}} \qquad \text{식 (2-7)}$$

전체 단어를 대상으로 확률을 구하는 식 (2-5)는 전체 단어의 수 V가 크면 학습 속도가 느리다. 속도 향상을 위한 휴리스틱(Heuristic)으로써 윈도우 내 단어를 포지티브 샘플(Positive Sample)로 두고 윈도우 밖 일부의 단어를 네거티브 샘플(Negative Sample)로 뽑아 이 범위 안에서만 예측을 하는 네거티브 샘플링(Negative Sampling) 기법 사용한다. 여기

서 식 (2-8)을 네거티브 샘플이 뽑힐 확률에 대한 분포로 사용하였으며, $f(y_i)$는 식 (2-7)에서와 동일한 함수이다.

$$P(y_i) = \frac{f(y_i)^{3/4}}{\sum_{j=0}^{n} f(y_i)^{3/4}} \qquad \text{식 (2-8)}$$

[19]에서 예측 기반으로 문맥적 의미를 단어 간 유사도로 유추할 수 있었다. 하지만 전체적인 통계정보를 고려하지 않았다는 단점이 있어 Glove 알고리즘[20]에서 이 단점을 개선하였다. Glove 방식은 단어의 카운트 기반으로 구한 통계정보를 이용하였다. 말뭉치의 전체적인 통계정보를 고려하기 위해 그림 3과 같이 전체 말뭉치에 대한 동시 등장행렬(Co-occurrence Matrix) $\mathbf{X}$를 정의하였다. 여기서 $\mathbf{X}_{ij}$는 i번째 단어와 j번째 단어의 특정 윈도우 내에서 동시 등장 수이며 $\mathbf{X}_i$는 $\sum_j \mathbf{X}_{ij}$를 의미한다. 여기서 $i = 1,2,...,V,\ j = 1,2,...,V$ 이며 $\mathbf{X}$는 대칭 행렬(Symmetric Matrix)이다.

카운트	I	like	enjoy	deep	learning	NLP	flying
I	0	2	1	0	0	0	0
like	2	0	0	1	0	1	0
enjoy	1	0	0	0	0	0	1
deep	0	1	0	0	1	0	0
learning	0	0	0	1	0	0	0
NLP	0	1	0	0	0	0	0
flying	0	0	1	0	0	0	0

그림 3 동시 등장행렬의 예

앞서 구한 동시 등장행렬을 이용하여 i번째 단어의 특정 윈도우 내에 k번째에 단어가 등장할 확률 $P(k|i)$는 다음 식 (2-9)을 통해 계산된다.

$$P(k|i) = \frac{\mathbf{X}_{ik}}{\mathbf{X}_i} \qquad \text{식 (2-9)}$$

예를 들면 그림 4의 통계정보를 구할 수 있다. 그림 4의 첫 번째 열을 보면 ice가 주어졌을 때 solid가 등장할 확률이 steam보다 높아 ice와 solid의 관련성이 더 높다고 볼 수 있다. 그림 4의 마지막 행의 첫 번째 열

은 ice와 steam의 동시 등장행렬 확률 간 비율이며 ice와 solid의 관련성이 steam과의 관련성보다 월등히 높음을 알 수 있다.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

그림 4 동시 등장행렬로부터 구한 통계정보의 예

Glove 알고리즘은 이러한 단어 사이의 관계를 알 수 있는 통계정보가 반영되도록 동시 등장 확률의 크기 관계 비율 $\frac{P(k|i)}{P(k|j)}$을 벡터 공간에 인코딩하고자 임베딩 된 중심단어로 쓰인 i번째와 j번째 단어의 벡터 $\mathbf{w}_i$, $\mathbf{w}_j$와 주변 단어로 쓰인 k번째 단어의 벡터 $\widetilde{\mathbf{w}}_k$가 주어질 때, 다음 식(2-10)을 만족하는 함수 F를 찾도록 하였다.

$$F(\mathbf{w}_i, \mathbf{w}_j, \widetilde{\mathbf{w}}_k) = \frac{P(k|i)}{P(k|j)} \qquad \text{식 (2-10)}$$

식 (2-10)에서 선형 공간(linear Space)에서 단어 의미 관계를 표현하기 위해 다음 식 (2-11)과 같이 $\mathbf{w}_i - \mathbf{w}_j$와 $\widetilde{\mathbf{w}}_k$를 F함수의 인자로 받아 내적 하도록 하였다.

$$F(\mathbf{w}_i - \mathbf{w}_j, \widetilde{\mathbf{w}}_k) = F((\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}}\widetilde{\mathbf{w}}_k) = \frac{P(k|i)}{P(k|j)} \qquad \text{식 (2-11)}$$

식 (2 -11)에서 동시 등장행렬 $\mathbf{X}$는 $\mathbf{X}_{ik}$와 $\mathbf{X}_{ki}$가 같은 대칭 행렬이므로 F함수에서 $\mathbf{w}_i$와 $\widetilde{\mathbf{w}}_k$ 사이의 교환이 자유롭도록 가정한다. 이와 같은 성질을 만족하는 함수는 지수함수이므로 다음 식 (2-12)이 된다.

$$\exp((\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}}\widetilde{\mathbf{w}}_k) = \frac{\exp(\mathbf{w}_i^{\mathrm{T}}\widetilde{\mathbf{w}}_k)}{\exp(\mathbf{w}_j^{\mathrm{T}}\widetilde{\mathbf{w}}_k)} = \frac{P(k|i)}{P(k|j)} \qquad \text{식 (2-12)}$$

위의 결과로부터 정리하면 다음 식(2-13)과 같이 $\mathbf{w}_i$와 $\mathbf{w}_k$의 내적이 전체 말뭉치에서의 두 단어의 동시 등장 확률(Co-occurrence Probability)의 로그 값 $\log P(k|i)$이 되도록 하면 된다.

$$\mathbf{w}_i^{\mathrm{T}}\widetilde{\mathbf{w}}_k = \log P(k|i) = \log\frac{X_{ik}}{X_i} = \log\mathbf{X}_{ik} - \log\mathbf{X}_i \qquad \text{식 (2-13)}$$

그런데 식 (2-13)에서 $\mathbf{w}_i$와 $\widetilde{\mathbf{w}}_k$사이에 교환법칙이 성립해야 한다는 가정이 맞지 않는다. 따라서, $\mathbf{X}_i$를 바이어스 b_i와 $\tilde{b}_k$로 대체하는 휴리스틱(heuritic)을 사용하여 다음 식 (2-14)을 만족하도록 하였다.

$$\mathbf{w}_i^{\mathrm{T}}\widetilde{\mathbf{w}}_k = \log\mathbf{X}_{ik} - b_i - \tilde{b}_k \qquad \text{식 (2-14)}$$

식 (2-14)에서 좌변과 우변의 차이를 최소로 하도록 다음 식 (2-15)에서 손실 함수 J를 정의한다. 이때, $\mathbf{X}_{ik}$가 특정 값 이상으로 튀는 경우를 방지하고자 $f(x)$를 손실 함수에 추가하여 다음 식 (2-16)과 같이 J를 재정의한다. 이를 최소화하는 방향으로 학습 파라미터 $\mathbf{w}_i$,$\widetilde{\mathbf{w}}_k$, b_i, $\tilde{b}_k$를 학습한다.

$$J = \sum_{i,k=1}^{V} (\mathbf{w}_i^{\mathrm{T}}\widetilde{\mathbf{w}}_k + b_i + \tilde{b}_k - \log\mathbf{X}_{ik})^2 \qquad \text{식 (2-15)}$$

$$J = \sum_{i,k=1}^{V} f(\mathbf{X}_{ik})(\mathbf{w}_i^{\mathrm{T}}\widetilde{\mathbf{w}}_k + b_i + \tilde{b}_k - \log\mathbf{X}_{ik})^2 \qquad \text{식 (2-16)}$$

$$where\ f(x) = \begin{cases} (\frac{x}{x_{\max}})^{\alpha} & \text{if } x < x_{\max} \\ 1 & o.w \end{cases}$$

J를 최소화하는 방향으로 학습하여 구한 다음 식 (2-17)을 통해 $\mathbf{w}_i \in \mathbb{R}^d$와 $\widetilde{\mathbf{w}}_i \in \mathbb{R}^d$를 더하여 i번째 단어에 대한 임베딩 벡터$\mathbf{w}_i^*$를 얻는다.

$$\mathbf{w}_i^* = \mathbf{w}_i + \widetilde{\mathbf{w}}_i \qquad \text{식 (2-17)}$$

제 3 절 컨볼루션 신경망

컨볼루션 신경망(Convolution Neural Network, CNN)은 이미지 분야에서 주로 사용되는 인공 신경망 구조이다. 컨볼루션 신경망은 공유되는 학습 파라미터를 가진 필터를 가지고 최근 컨볼루션 신경망을 사용하여 리뷰의 단어들로부터 문맥적 특징(Contextual Feature)을 효과적으로 구하는 연구가 진행되었다[5-8]. 본 논문에서는 텍스트 데이터에 대한 1차원 컨볼루션 신경망을 기준으로 소개한다.

컨볼루션 신경망은 그림 5와 같이 컨볼루션 레이어(Convolutional Layer)와 풀링레이어(Pooling Layer)로 구성된다.

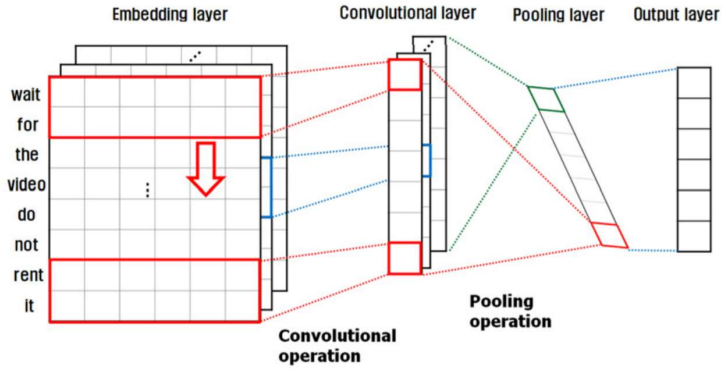

그림 5 컨볼루션 신경망의 구조

컨볼루션 레이어의 입력은 식 (2-18)처럼 각 원소는 d차원 벡터 $\mathbf{w}_i \in \mathbb{R}^d$이고 l길이의 입력 벡터 시퀀스 $\mathbf{W} \in \mathbb{R}^{l \times d}$가 주어진다.

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_l] \quad where \quad \mathbf{w}_i \in \mathbb{R}^d \qquad \text{식 (2-18)}$$

컨볼루션 연산을 위한 필터의 너비는 1차원 연산이므로 $\mathbf{w}_i$의 차원 d과 같으며 필터의 높이는 윈도우 크기w와 같은 값으로 사용한다. j번째 채널에서 생성된 문맥적 특징 벡터 $c_i^j \in \mathbb{R}$를 구하는 과정은 식 (2-19)와 같이 윈도우 크기가 w일 경우 길이의 i번째부터 $i+w-1$번째 입력 벡터 시퀀스 $\mathbf{W}_{(i:i+w-1)}$와 j번째 필터 $\mathbf{W}_c^j \in \mathbb{R}^{w \times d}$에 대해 컨볼루션 연산을 한 후 j번째 바이어스 b_c^j를 더하고 비선형 함수 $\sigma(.)$를 취함으로써 계산된다. 여기서 $j = 1,2,...,f$ 이다.

$$c_i^j = \sigma(\mathbf{W}_{(i:i+w-1)} * \mathbf{W}_c^j + b_c^j) \qquad \text{식 (2-19)}$$

식 (2-19)를 통해 윈도우 w크기의 컨볼루션 연산을 하면 윈도우 내에서의 특징 값을 구하기 때문에 1번째부터 $l-w+1$번째까지 특징 값 필터의 개수 f만큼 생성된다. 이때, 주어진 입력 시퀀스의 길이 원하는 길이의 특징 시퀀스를 만들기 위해서 패딩(Padding) 처리가 필요하다. 슬라이딩 하는 윈도우의 스트라이드(stride) 크기가 S라면 P만큼 입력 시퀀스의 앞과 뒤에 패딩 처리하여 컨볼루션 연산을 하면 $\frac{(l-w+1+2P)}{S}$ 개의 특징 값이 생성 된다. l만큼 특징 벡터를 만들기 위한 P값은 식 (2-20)과 같이 계산된다.

$$P = \frac{(S-1)l + w - 1}{2} \qquad \text{식 (2-20)}$$

풀링 레이어는 그림 5와 같이 컨볼루션 연산에 의해 생성된 문맥적 특징 벡터에 대한 차원을 줄임으로써 과적합(Overfitting)을 방지하는 역할을 한다. 차원을 줄이는 방법은 f개의 채널에 대해 다음 식 (2-21)처럼 i번째 특징 벡터를 평균을 내는 평균 풀링(Average Pooling), 식 (2-22)과 같이 f개의 채널 중 가장 큰 값을 추출하는 최대 풀링(Max Pooling) 등의 방법이 있다. 여기서 $i = 1, 2, ..., l-w+1$ 이다.

$$z_i = mean(c_i^1, c_i^2, ..., c_i^f) \qquad \text{식 (2-21)}$$

$$z_i = \max(c_i^1, c_i^2, ..., c_i^f) \qquad \text{식 (2-22)}$$

제 4 절 어텐션 메커니즘

자연어처리(Natural Language Process, NLP)분야에서 어텐션 메커니즘은 기계번역(Machine Translation)을 위한 딥러닝 모델의 성능향상을 위해 처음 사용되었다[12]. 최근 추천시스템 분야에서 어텐션 메커니즘을 바탕으로 리뷰에서 평점에 영향을 끼치는 요소를 집중하도록 하여 성능을 향상하는 연구가 진행되었다[6-8].

[6-8]에서 컨볼루션 신경망(Convolution Neural Network, CNN)과 어텐션 메커니즘을 사용하여 추천시스템 모델의 성능향상과 동시에 분석성능이 향상되었다. [8]의 어텐션 메커니즘은 어텐션(Local Attention) 방식과 상호 어텐션(Mutual Attention) 방식 두 단계로 나뉜다.

로컬 어텐션(Local Attention) 방식은 [그림]과 같이 윈도우를 슬라이딩 하면서 윈도우 내 단어들 중에 평점에 영향을 주는 유익한 단어를 추려내는데 도움을 준다.

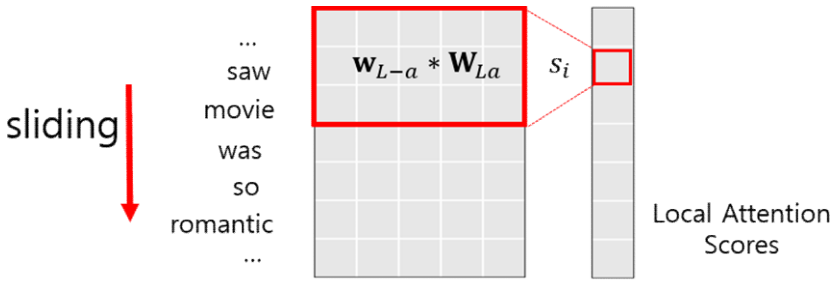

그림 6 로컬 어텐션 스코어의 계산 과정

컨볼루션 연산은 윈도우 크기를 w로 가정하면 필터 $\mathbf{W}_{La} \in \mathbb{R}^{d \times w}$와 d차원의 l개의 단어로 구성된 임베딩 매트릭스 $\mathbf{W} \in \mathbb{R}^{d \times l}$에서 필터와 대응되는 영역 $\mathbf{w}_{L-a} \in \mathbb{R}^{d \times w}$을 바탕으로 계산된다. 여기서 $\mathbf{w}_i \in \mathbb{R}^d$를 $\mathbf{W}$의 i번째 벡터라고 하면 $\mathbf{w}_{L-a}$는 식 (2-23)으로 표현된다. 로컬 어텐션 스코어는 (2-23)과 같이 필터의 윈도우 크기 w내에 단어들에 대하여 평점에 영향을 주는 정도를 나타내는 0과 1사이의 값이 되도록 식 (2-24)을 통해 계산한다. 여기서 *는 컨볼루션 연산이고, $b_{La} \in \mathbb{R}$는 바이어스 값이다.

$$\mathbf{w}_{L-a} = [\mathbf{w}_{i+\frac{-w+1}{2}} \ \dots \ \mathbf{w}_i \ \dots \ \mathbf{w}_{i+\frac{w+1}{2}}] \qquad \text{식 (2-23)}$$

$$s(i) = \delta(\mathbf{w}_{L-a} * \mathbf{W}_{La} + b_{La}) \qquad \text{식 (2-24)}$$

식 (2-24)에서 구한 로컬 어텐션 스코어를 식 (2-25)를 통해 슬라이딩 윈도우 내에서 단어들의 중요도가 반영된 단어 임베딩 벡터 $\hat{\mathbf{w}}_i \in \mathbb{R}^d$를 얻는다.

$$\hat{\mathbf{w}}_i = s(i)\mathbf{w}_i \qquad \text{식 (2-25)}$$

상호 어텐션은 유저와 아이템 사이의 단어 연관성을 파악하여 평점 추측에 도움을 주는 방식이다. [그림]과 같이 유저의 리뷰와 아이템에 관한 리뷰에서 단어들 사이의 관련도를 계산하여 리뷰 특징 벡터에 반영한다.

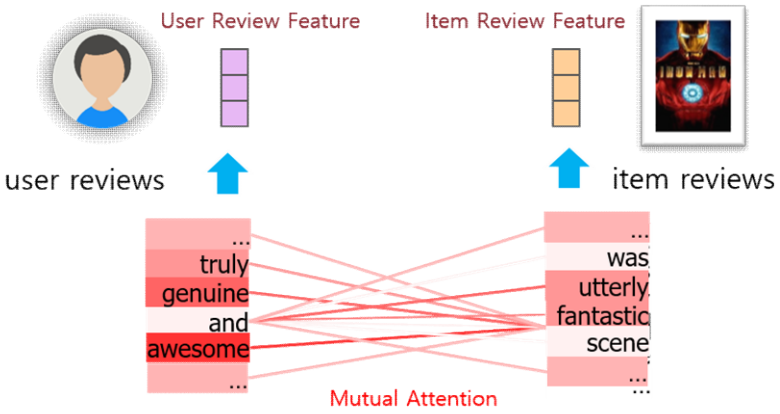

그림 7 상호 어텐션 메커니즘의 단어 연관성 시각화

u번째 유저의 리뷰에서 k번째 단어와 i번째 아이템에 쓰인 리뷰에서 j번째 단어 사이의 관련도 $\mathbf{A}_{kj}$는 식 (2-26)를 통해 두 임베딩 벡터 사이의 유클리디안 거리(Euclidean distance)를 나누는 방식으로 계산한다. 여기서 $\mathbf{w}_k^u \in \mathbb{R}^d$는 유저의 리뷰에서 k번째 단어에 대한 임베딩 벡터 이고, $\mathbf{w}_j^i \in \mathbb{R}^d$는 아이템에 쓰인 리뷰에서 j번째 단어에 대한 임베딩 벡터이다. 그리고 분모가 0이 됨을 방지하기 위해 1을 더하였다.

$$\mathbf{A}_{kj} = \frac{1}{1 + |\mathbf{w}_k^u - \mathbf{w}_j^i|} \qquad \text{식 (2-26)}$$

식 (2-26)에서 구한 단어 사이의 관련도를 바탕으로 u번째 유저가 쓴 리뷰에서 k번째 단어와 u번째 유저가 평점을 매긴 아이템에 쓰인 리뷰의 모든 단어와의 관련도 $g_k^u \in \mathbb{R}$를 식 (2-27)을 통해 구한다. i번째 아이템에 쓰인 리뷰에서 j번째 단어의 관련도 g_j^i도 비슷한 방식으로 구한다.

$$\begin{cases} g_k^u = \sum_k \mathbf{A}[k,:] \\ g_j^i = \sum_j \mathbf{A}[:,j] \end{cases} \quad \text{식 (2-27)}$$

위의 상호 어텐션 스코어를 바탕으로 문맥적인 요소를 반영하기 위해 식 (2-28)을 통해 윈도우 크기 w안에서 단어 임베딩에 적용하여 u번째 유저의 k번째 문맥적 단어 특징 벡터(Contextual Feature Vectors) $\mathbf{c}_k^u$와 i번째 아이템의 j번째 문맥적 단어 특징 벡터 $\mathbf{c}_j^i$를 구한다. 여기서 $\bullet$는 실수 사이의 곱셈을 의미한다.

$$\begin{cases} \mathbf{c}_k^u = \sum_{k=k:k+w} g_k^u \bullet w_k^u \\ \mathbf{c}_j^i = \sum_{j=j:j+w} g_j^i \bullet w_j^i \end{cases} \quad \text{식 (2-28)}$$

[7]은 유저는 여러 아이템에 대해 리뷰를 쓰고 아이템도 여러 유저에 의해 쓰인다. 이를 구분하여 컨볼루션과 풀링을 통해 리뷰 별 특징 벡터를 구하고 리뷰 수준(Review-Level) 어텐션 메커니즘을 통해 리뷰의 중요도를 학습하여 반영하였다.

제 3 장 제안하는 어텐션 메커니즘

[5-8]에서 유저에 대한 신경망과 아이템에 대한 신경망을 각각 두고 유저의 선호도와 아이템에 대한 특성을 나타내는 특징 벡터(Feature Vector)를 찾고자 하였다. 본 논문의 제안하는 방식은 [8]에서 사용된 모델을 기반으로 하였으며 기존 연구[8]는 크게 두 모듈이 있다. 먼저, 리뷰들로부터 유저와 아이템에 대한 특징(Feature)을 각각 뽑아내는 것을 배우는 특징 학습(Feature Learning) 모듈이 있다. 그리고 평점 정보에 의한 유저의 선호도와 아이템의 속성을 고려하여 상호작용 사이에서 서로 다른 잠재적 특징을 융합하여 평점을 예측하는 특징 상호작용(Feature Interaction) 모듈이 있다. [8]의 특징 학습 모듈 안에서 두 가지 어텐션 방식이 사용된다. 특징 학습 모듈 안에서 두 가지 방식의 어텐션 레이어가 사용된다. 로컬 어텐션(Local Attention)방식은 슬라이딩 윈도우(sliding)를 옮겨가며 그 안에서의 평점에 영향을 끼치는 단어의 중요도(Attention score)를 구하는 방식이다. 상호 어텐션(Mutual Attention)방식은 유저와 아이템의 리뷰 단어 사이의 상관관계에 대한 스코어를 구하고 종합하여 단어 어텐션 스코어를 구한다.

본 논문에서는 [8]의 로컬 어텐션 메커니즘이 유저 혹은 아이템마다 다른 어텐션 정보를 반영하도록 개별적 아이디를 적용하여 단어들의 문맥적 특징 벡터를 구하였다.

제 1 절 개별적 정보를 반영한 어텐션 메커니즘

본 논문에서는 [8]의 로컬 어텐션(Local Attention)방식에 집중하였다. [8]의 로컬 어텐션 방식에서 서로 다른 유저 또는 아이템을 구분하지 않고 동일한 단어 중요도를 사용하였다. 기존의 방식과 달리 유저와 아이템에 아이디에 대한 개별적인 단어 중요도를 반영하고자 유저 또는 아이템에 대한 정보를 반영한 특징 벡터(Id Attention Feature Vector)를 두고 이를 학습하여 단어의 중요도에 반영하였다. 제안하는 방법을 통해 구한 유저와 아이템의 단어수준(Word-Level) 문맥적 특징 벡터를 구하

고 유저의 선호도와 아이템의 특성 벡터 추출에 반영하였다. 본 논문에 적용된 각 층에 대해서는 상세히 나누어 설명하도록 하겠다.

임베딩 레이어(Embedding Layer): 임베딩 레이어는 길이 l의 단어의 시퀀스로 이루어진 리뷰를 식 (3-1)와 같이 리뷰 행렬 $\mathbf{D}\in\mathbb{R}^{d\times l}$로 바꾼다.

$$\mathbf{D}=[\mathbf{w}_1,\mathbf{w}_2,...,\mathbf{w}_i,...,\mathbf{w}_l] \qquad \text{식 (3-2)}$$

본 논문에서는 학습 변수의 초깃값을 Glove[11] 알고리즘에 의해 미리 학습한 값을 사용하였다.

컨볼루션 신경망(Convolution Neural Network): 컨볼루션 신경망은 이미지의 특징 추출에서 시작하여 텍스트 정보인 리뷰에서도 문맥적인 단어 특징을 효과적으로 뽑아내었다[5]. 구조는 각 단어를 문맥적 의미를 추출하는 컨볼루션 레이어, 특징 벡터의 차원을 낮추고, 필요한 성분을 추출하는 풀링 레이어로 구성된다.

어텐션 메커니즘(Attention Mechanism): 리뷰에서 각 단어는 유저와 아이템의 특성을 나타내는 요소이다. 이 단어들이 특징이 평점에 영향을 끼치는 정도에 대한 스코어를 계산하여 더 효과적으로 리뷰 특징을 추출할 수 있다. 예를 들면, 한 유저의 리뷰 행렬 $\mathbf{D}\in\mathbb{R}^{d\times l}$은 길이$l$의 단어 벡터 $\mathbf{w}_{k=1,2,...,l}\in\mathbb{R}^d$들이 있을 때, 각 단어에 해당하는 어텐션 스코어 $\mathbf{s}_{k=1,2,...,l}\in\mathbb{R}$로 구성된 벡터 $\mathbf{S}\in\mathbb{R}^l$를 어텐션 레이어를 거쳐 구하고, 그 스코어를 각 단어 벡터에 곱하면 어텐션이 적용된 단어 벡터들 $\hat{\mathbf{w}}_{k=1,2,...,l}\in\mathbb{R}^d$로 구성된 리뷰 행렬 $\hat{\mathbf{D}}\in\mathbb{R}^{d\times l}$을 다음 식 (3-3, 4)과 같이 얻는다.

$$\mathbf{S}=[s_1,s_2,...,s_i,...,s_l] \qquad \text{식 (3-3)}$$

$$\hat{\mathbf{w}}_i=s_i\mathbf{w}_i \qquad \text{식 (3-4)}$$

로컬 어텐션 스코어 계산방식에 개별적인 유저 혹은 아이템의 측면에서

단어 중요도를 효과적으로 학습하기 위해 다음 그림 8과 같은 로컬 어텐션 스코어 방식을 제안한다.

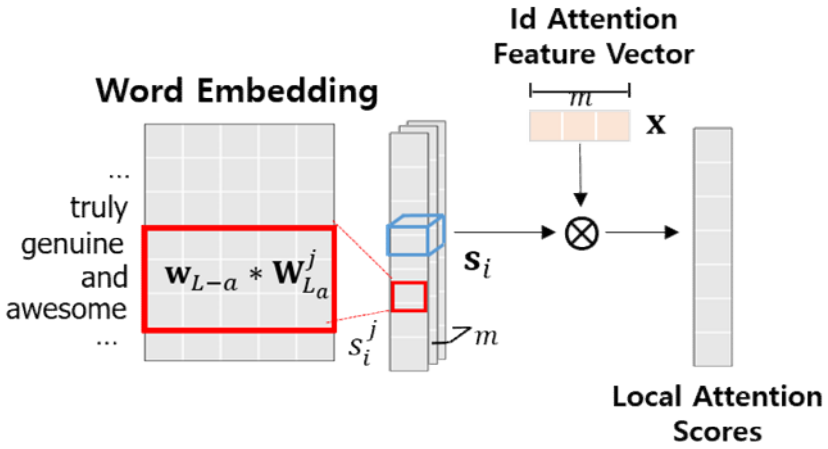

그림 8 로컬 어텐션 스코어 계산방식

필터 $\mathbf{W}^j_{La} \in \mathbb{R}^{w \times d}$로 컨볼루션 연산을 통해 w_i에 대한 스코어를 다음 식과 식 (3-5, 6, 7)와 같이 구한다. ($\mathbf{x} \in \mathbb{R}^m$, σ는 sigmoid 함수)

$$\mathbf{w}_{L-a} = [\mathbf{w}_{i+\frac{-w+1}{2}}, \ldots, \mathbf{w}_i, \ldots, \mathbf{w}_{i+\frac{w+1}{2}}] \qquad \text{식 (3-5)}$$

$$s_i^j = \mathbf{w}_{L-a} * \mathbf{W}^j_{La} + b^j_{La}, \ \mathbf{s}_i = [s_i^1, s_i^2, \ldots, s_i^m] \qquad \text{식 (3-6)}$$

$$s_i = \sigma(\mathbf{s}_i \mathbf{x}^T) = \sigma(\sum_{j=1}^{m} s_i^j x_j) \qquad \text{식 (3-7)}$$

제 2 절 리뷰 수준 어텐션 메커니즘

제 1절의 단어 수준(Word-Level)의 어텐션 메커니즘을 적용한 후 풀링 레이어를 통해 리뷰 별 특징 벡터(Review Feature Vector)로 종합한다. 종합된 리뷰 별 특징 벡터에 [7]의 리뷰 수준(Review-Level) 어텐션 메커니즘을 적용하는 모델을 제안한다. 유저와 아이템 네트워크가 대칭적이므로 유저가 쓴 리뷰를 기준으로 설명하겠다.

단어 수준 어텐션 메커니즘: [5, 6, 8]의 어텐션 스코어 계산방식은 유저가 쓴 리뷰들 또는 아이템에 쓰인 모든 리뷰들을 연결하여 사용하였다. 이와 달리 본 논문에서는 유저가 쓴 리뷰들은 아이템별로 나누고 아이템에 쓰인 리뷰들은 유저별로 구분 지어 단어 수준 어텐션 메커니즘이 적용된 리뷰 특징 벡터를 얻도록 그림 9와 같은 방식을 제안한다.

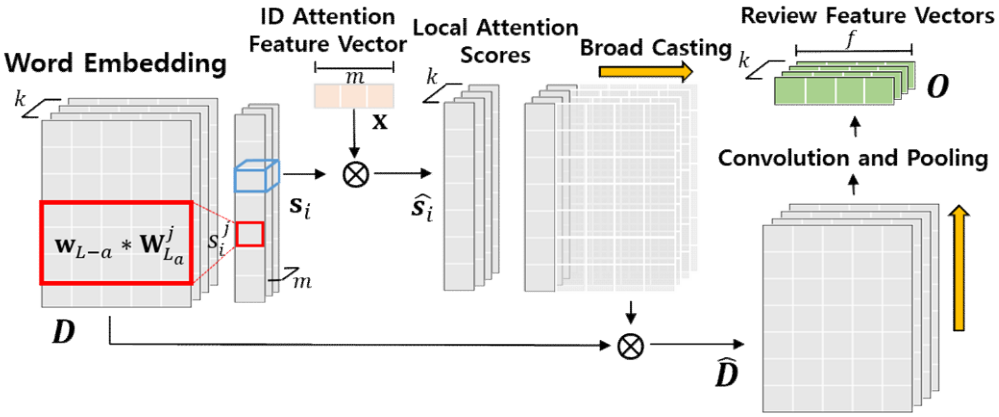

그림 9 제안하는 단어 수준 어텐션 메커니즘

유저가 쓴 아이템 리뷰의 수를 k로 두고 각 리뷰의 시퀀스 길이를 T로 가정하면 1절에서 제안하는 방식을 통해 리뷰 행렬 $\hat{\mathbf{D}}=[\widehat{\mathbf{w}_1},\widehat{\mathbf{w}_2},\ldots,\widehat{\mathbf{w}_T}]$를 얻는다. 이후 단어 수준의 특징 벡터를 리뷰 수준으로 종합하기 위해 윈도우 w크기의 f개 필터$\mathbf{K}^{j=1,2,\ldots,f}\in\mathbb{R}^{w\times d}$로 컨볼루션과 평균 풀링 레이어를 거쳐 리뷰 특징 벡터$\mathbf{O}\in\mathbb{R}^{k\times f}$를 다음 식 (3-8, 9, 10)과 같이 구한다.

$$\hat{\mathbf{w}}_{L-a} = [\hat{\mathbf{w}}_{i+\frac{-w+1}{2}}, \ldots, \hat{\mathbf{w}}_i, \ldots, \hat{\mathbf{w}}_{i+\frac{w+1}{2}}] \quad \text{식 (3-8)}$$

$$z_i^j = \delta(\hat{\mathbf{w}}_{L-a} * \mathbf{K}_j + b_j) \quad \text{식 (3-9)}$$

$$o_n^j = mean(z_1^j z_2^j \ldots, z_T^j) \quad \text{식 (3-10)}$$

리뷰 수준 어텐션 메커니즘 적용: 유저의 아이템들에 대한 리뷰 모음 혹은 아이템에 쓰인 여러 유저들의 리뷰 모음에서 중요한 리뷰를 학습을 통해 반영하도록 단어 수준 어텐션 메커니즘으로부터 구한 리뷰 특징 벡터들에 대해 다음 그림 10과 같은 방식을 사용한다.

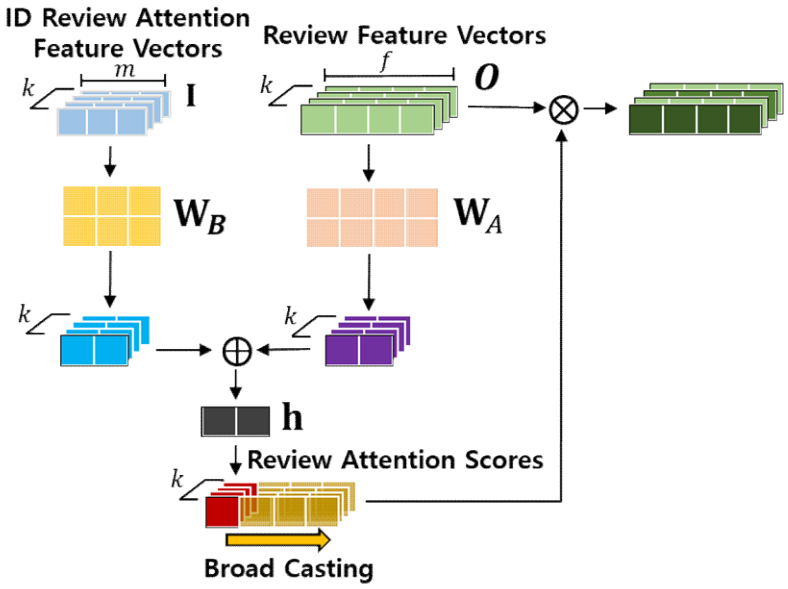

그림 10 리뷰 수준 어텐션 메커니즘

유저의 리뷰 특징 벡터$\mathbf{O} \in \mathbb{R}^{k \times f}$와 유저가 쓴 아이템들의 아이디 벡터 $\mathbf{I} \in \mathbb{R}^{k \times m}$(개별적 요인의 수 m)를 바탕으로 리뷰 수준 어텐션 스코어 $\mathbf{a} \in \mathbb{R}^k$는 다음 식 (3-11, 12)과 같이 구한다.

($\mathbf{h} \in \mathbb{R}^t, \mathbf{W}_A \in \mathbb{R}^{f \times t}, \mathbf{W}_B \in \mathbb{R}^{m \times t}, \mathbf{b}_1 \in \mathbb{R}^{k \times t}, \mathbf{b}_2 \in \mathbb{R}^{k \times 1}$)

$$\mathbf{a}^* = \mathbf{h}^{\mathrm{T}} \delta(\mathbf{O}\mathbf{W}_A + \mathbf{I}\mathbf{W}_B + \mathbf{b}_1) + \mathbf{b}_2 \quad \text{식 (3-11)}$$

$$a_i = \mathbf{a}_i^* / \sum_{i=1}^{k} \mathbf{a}_i^*, \ \mathbf{a} = [a_1, a_2, \ldots, a_k] \quad \text{식 (3-12)}$$

제 4 장 실험

제 1 절 실험 환경 및 실험 데이터

1. 실험 환경

Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz에 GeForce GTX 1080 Ti 11GB, 메인 메모리 32GB머신에서 Pytorch 1.4.0을 이용하여 실험을 수행하였다.

2. 실험 데이터

데이터는 Amazon 5-core[1)]의 Musical Instruments (Amazon1), Grocery and Gourmet Food (Amazon2), Sports and Outdoors (Amazon3) 이다. 표 1에 각각의 실험 데이터의 통계특징을 나타내었다. 학습 데이터에 유저와 아이템마다 평점 혹은 리뷰가 적어도 하나 이상 있도록 전처리하였다. 전체 데이터 중 학습, 검증, 시험 비율을 8:1:1로 나누었다.

Dataset	#users	#items	#ratings	density(%)
Amazon1	1,429	900	10,261	0.798
Amazon2	14,681	8,713	151,254	0.118
Amazon3	35,598	18,357	296,337	0.045

표 1 실험 데이터의 통계특징

1) http://jmcauley.ucsd.edu/data/amazon/

제 2 절 성능 측정 지표 및 세부 사항

모델 성능을 평가로 평균 절댓값 오차(MAE)를 사용하였다. 실제 유저가 매긴 평점의 수 N, 실제 평점 $r_{u,i} \in \mathrm{R}$, 추정값 $\hat{r}_{u,i}$을 바탕으로 다음 식 (4-1)을 통해 계산한다.

$$MAE = \frac{1}{N} \sum_{(u,i) \in R} |r_{u,i} - \hat{r}_{u,i}| \qquad \text{식 (4-1)}$$

모델의 하이퍼파라미터 d, f, 배치사이즈는 100이고, w는 3, 개별적 요인 m은 32, t는 16이다. 학습 비율(learning rate)은 [4]를 참조하였다. 정규화 감쇄 상수는 [0.001 0.1]에서 그리디 탐색하였다. δ는 tanh, LeakyRelu 함수를 사용하였다.

제 3 절 실험 결과 및 분석

평균 절댓값 오차 비교: 표 2와 그림 11에서 성능은 실험을 3번 반복하였고, 학습 과정 중 검증데이터의 성능이 가장 좋은 모델로 시험데이터에 대한 성능의 평균을 측정하였다. 표 4에서 실험결과를 기존 논문 성능과 비교하였다.

Dataset	DAML	DAML+	DAML++	△%
Amazon1	0.651	0.6258	**0.5787**	11.1
Amazon2	0.7354	0.708	**0.6933**	5.72
Amazon3	0.6676	0.6424	**0.639**	4.284

표 2 기존연구와의 평균 절댓값 오차 비교

개별적 요인이 반영된 단어 수준 어텐션 메커니즘: 그림 11에서 DAML+ 모델에 대해 아이디 어텐션 잠재 요인 수 m은 $8, 16, 32$로 바꾸어가며 성능을 비교하였다. 제안하는 모델은 잠재 요인에 따라 크게 변동 없이 안정적인 성능을 보였다.

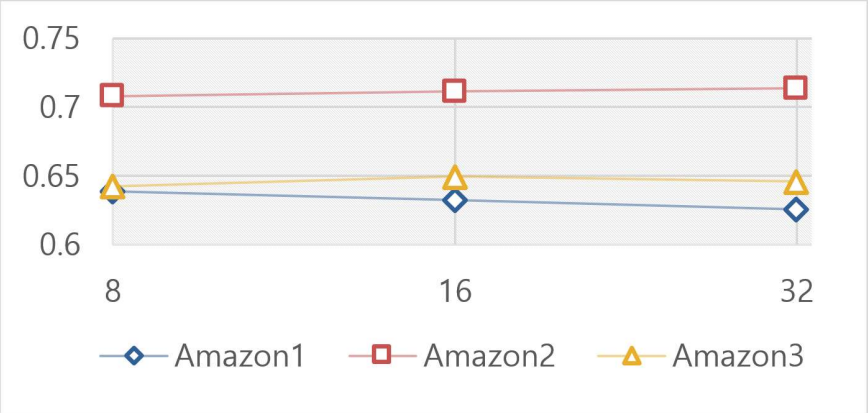

그림 11 어텐션 잠재 요인 수 m에 따른 성능 비교

그림 12에서는 Amazon3 데이터에서 유저 아이디 27380번과 31798번을 뽑아 리뷰에 대한 로컬 어텐션 스코어를 시각화하였다. 어텐션 스코어가

높을수록 초록색을 진하게 표시하였으며, 스코어가 낮을수록 빨간색을 진하게 표시하였다. 27380번 유저의 리뷰에서 단어 “well”에 대한 어텐션 스코어가 높아 선호도에 강한 영향을 주는 반면, 31798번 유저의 리뷰에서는 스코어가 낮았다. 단어 “very”는 27380번 유저에게는 스코어가 낮았지만 31798번 유저의 리뷰에서 높았다. 이 결과는 유저의 선호도에 영향을 주는 단어는 개별적인 특성이 있음을 시사한다.

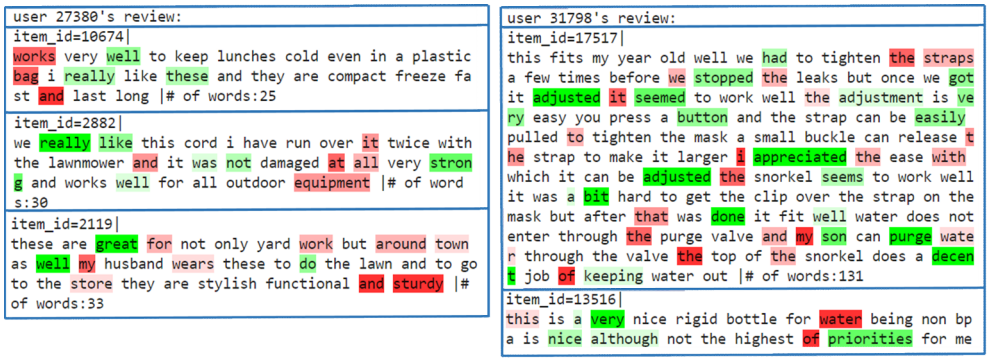

그림 12 단어 수준 어텐션 스코어 시각화

리뷰 수준 어텐션 메커니즘: 그림 13과 14는 DAML++ 모델의 리뷰 수준 어텐션 메커니즘을 통해 각 리뷰의 중요도를 학습한 결과를 표로 나타내었다. 각 그림에서 Au와 Ai는 학습된 리뷰의어텐션 스코어이며 첫 번째 줄에서 시험데이터에 대한 정보와 평점 예측 결과를 보였다.

user_id	item_id	ratings	pred	reviews
22489	7437	5.0	4.863518	You just cannot beat Lee products

user_id	item_id	ratings	reviews	Au
22489	13023	5.0	This is a great little punch set. Would buy it again	0.646877
22489	5437	5.0	Good service and you can't beat Lee products	0.252813
22489	7600	5.0	It's ok	0.055850
22489	245	5.0	Hey, it's Remington	0.044460

그림 13 평점 예측과 학습된 유저의 리뷰 중요도

그림 13에서 22489번 유저의 선호도를 파악하기에 중요한 정보가 많을

수록 높은 어텐션 스코어를 얻었다. “it’s ok”와 같이 일반적인 표현은 유저의 선호도를 파악하는데 큰 도움을 주지 못한다는 직관과 일치하였다.

item_id	user_id	ratings	pred	reviews
17865	22604	3.0	3.657427	Wore this once in the gym, washed it, and it shrunk to the point where I can't wear it anymore. Should probably go up a size with this one.

item_id	user_id	ratings	reviews	Ai
17865	3240	1.0	all in all this is a pretty cheap t-shirt. returned because the quality was poor and the size (mens large) was pretty short. Would not order again	0.455895
17865	16487	5.0	My son will be attending this Fall and looks forward to showing it off! Love the colors and the fit.	0.280128
17865	15565	5.0	Good shirt and better color for the price. I needed a good Husky shirt for a reasonable price and this shirt met all expectations.	0.263977

그림 14 평점 예측과 학습된 아이템에 쓰인 리뷰 중요도

그림 14는 17865번 아이템의 특성을 파악하는데 여러 유저가 쓴 리뷰의 학습된 어텐션 점수를 보였다. 첫 번째 리뷰는 평점을 낮은 점수를 주었지만 티셔츠의 질이나 사이즈 정보가 묘사되어 있으므로 높은 점수를 얻었다. 이 결과를 통해 추천하고자 하는 아이템에 대한 이유를 어텐션 스코어가 높은 리뷰로부터 추론할 수 있다.

제 5 장 결론

본 논문에서 [8]의 로컬 어텐션 레이어 구조를 개선하여 유저와 아이템에 아이디에 대한 개별적인 단어 중요도를 반영하는 개선된 방법을 제시하였다. 이 방법에 더하여 [7]의 리뷰 수준 어텐션 메커니즘을 융합하는 방식을 제시하였다. 그리고 제시된 두 가지의 모델에 대해 파생되는 효과를 분석하였다. 첫 번째로 제시된 단어 수준의 개별적 요인이 반영되었을 때 나타나는 효과로서 서로 다른 유저의 선호도에 영향을 분석하였다. 두 번째로 학습된 리뷰 별 어텐션 스코어를 바탕으로 학습된 중요도를 보이고 이를 통해 추천하는 이유에 대한 설명이 가능함을 보였다. 실생활 데이터를 이용하여 [8]의 방법보다 성능이 최대 11.1% 향상된 것을 실험적으로 확인하였다.

리뷰를 이용한 맞춤형 추천시스템에 본 논문에서 제안된 기법들을 이용하면 리뷰에 쓰이는 단어에 유저 혹은 아이템의 개별적인 요인을 반영할 수 있다. 그리고 각 유저의 아이템에 따른 리뷰의 중요도 혹은 아이템의 여러 유저의 리뷰의 중요도를 반영하여 효과적으로 유저의 선호도와 아이템의 속성을 파악할 수 있다.

참 고 문 헌

[1] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. Item-based collaborative filtering recommendation algorithms. WWW, 2001

[2] Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet computing, 7(1), 76-80.

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173-182).

[5] Zheng, L., Noroozi, V., & Yu, P. S. (2017, February). Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 425-434).

[6] Seo, S., Huang, J., Yang, H., & Liu, Y. (2017, August). Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the eleventh ACM conference on recommender systems (pp. 297-305).

[7] Chen, C., Zhang, M., Liu, Y., & Ma, S. (2018, April). Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference (pp. 1583-1592).

[8] Liu, D., Li, J., Du, B., Chang, J., & Gao, R. (2019, July). DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 344-352).

[9] Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017, August). Deep Matrix Factorization Models for Recommender Systems. In IJCAI (pp. 3203-3209).

[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[11] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[12] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Abstract

Improving Recommender Systems with Review-based Attention Mechanism

Sungwook Yoo

Electrical and Computer Engineering

The Graduate School

Seoul National University

The Recommender system is used by many companies to generate revenue by driving user consumption across various platforms. Matrix factorization method used earlier was difficult to recommend with high accuracy due to data sparseness and cold-start problems. Recent studies have attempted to incorporate review information into rating predictions. Among them, it was efficient to apply the attention mechanism to text information. However, the recent studies can be improved by considering two things. First, current word-level attention mechanism cannot reflect individual factors by each user, or item. Second, review-level attention mechanism is possible by assigning the usefulness of each review. This is because a user writes reviews on several items, and the items are also reviewed by

several users. By discriminating this, you can find the reviews that are useful for extracting user preferences or item properties.

In this paper, we propose two models that improve the baseline model. First model reflects the individual factors for each user or each item by revising the calculation process of word-level attention mechanism. Another model hierarchically combines the word-level attention mechanism which considers individual factors and the review-level attention mechanism which reflects usefulness of reviews. We show the effectiveness of our proposed model by conducting experiments with three real-life data sets.

keywords : recommender system, convolutional neural network, attention mechanism, individual factor
Student Number : 2018-26190

감사의 글

지난 2년 동안 무사히 석사과정을 마칠 수 있도록 도움 주신 분들께 감사의 인사를 드리고 싶습니다. 먼저 바쁘신 와중에도 성실하게 신입생 시절부터 꼼꼼한 지도로 성장할 수 있도록 도와주셨던 지도 교수님이신 심규석 교수님께 감사드립니다. 지도 교수님 덕에 컴퓨터과학에 대해 배울 수 있는 소중한 시간이었습니다. 귀중한 시간을 내어 심사위원을 맡아주신 김태환 교수님과 홍성수 교수님께도 감사의 인사를 드립니다.

연구실 분들께도 감사의 인사를 드립니다. 연구 방향이 올바른 길로 가도록 정말 필요한 조언을 자주 해주시던 우환 선배, 탁구 치며 적응하도록 도와주신 광호 선배, 신뢰감 있고 다정한 성웅 선배, 매사에 열정적으로 도움을 주신 한준 선배, 어려움이 있을 때마다 차근차근 해결하도록 도와주신 수용 선배, 연구실에서 재밌는 역사 이야기를 많이 들려주신 장혁 선배, 바쁜 와중에도 항상 친절히 도와주신 대영 선배, 동갑이지만 선배로서 많은 도움을 준 영준이, 동기로서 옆에서 힘이 되어준 영빈이, 짧은 기간이었지만 친절히 맞아주신 김진현 박사님과 희원 선배 모두 연구실 적응에 도움을 주신 분들이었습니다. 2년 동안 컴퓨터과학 분야의 지식부터 논문 발표 연습까지 많은 것들을 배울 수 있었습니다.

주변 친구들에게도 감사의 인사를 드립니다. 지난 2년 동안 지치고 힘들 때 술 한잔하며 용기를 준 동엽, 동현, 상진, 한용, 병윤, 우진, 태훈, 재욱, 찬호, 병욱, 충현, 곁에서 힘을 준 지영씨 모두 감사합니다.

묵묵히 연구실 생활을 위해 지원을 아끼지 않았던 가족들에게도 감사드립니다. 부모님의 든든한 지원 덕에 부족함 없이 연구실 생활을 하며 석사과정을 무사히 마칠 수 있었습니다. 그리고 비슷한 분야의 연구를 하는 동생도 든든한 지원군이었습니다. 도움 주신 분들 모두 평생 잊지 않겠습니다.

2020. 5. 유성욱 드림.