

심층적 의미 매칭을 이용한 cQA 시스템 질문 검색

김선훈[○], 장현석, 강인호

네이버

{seonhoon.kim, heonseok.jang, once.ihkang}@navercorp.com

Question Retrieval using Deep Semantic Matching

for Community Question Answering

Seon-Hoon Kim[○], Heon-Seok Jang, In-Ho Kang
Naver Corporation

요약

cQA(Community-based Question Answering) 시스템은 온라인 커뮤니티를 통해 사용자들이 질문을 남기고 답변을 작성할 수 있도록 만들어진 시스템이다. 신규 질문이 인입되면, 기존에 축적된 cQA 저장소에서 해당 질문과 가장 유사한 질문을 검색하고, 그 질문에 대한 답변을 신규 질문에 대한 답변으로 대체할 수 있다. 하지만, 키워드 매칭을 사용하는 전통적인 검색 방식으로는 문장에 내재된 의미들을 이용할 수 없다는 한계가 있다. 이를 극복하기 위해서는 의미적으로 동일한 문장들로 학습이 되어야 하지만, 이러한 데이터를 대량으로 확보하기에는 어려움이 있다. 본 논문에서는 질문이 제목과 내용으로 분리되어 있는 대량의 cQA 셋에서, 질문 제목과 내용을 의미 벡터 공간으로 사상하고 두 벡터의 상대적 거리가 가깝게 되도록 학습함으로써 의사(pseudo) 유사 의미의 성질을 내재화 하였다. 또한, 질문 제목과 내용의 의미 벡터 표현(representation)을 위하여, semi-training word embedding 과 CNN(Convolutional Neural Network) 을 이용한 딥러닝 기법을 제안하였다. 유사 질문 검색 실험 결과, 제안 모델을 이용한 검색이 키워드 매칭 기반 검색보다 좋은 성능을 보였다.

주제어: cQA, 질문 검색, 의미 매칭, 딥러닝

1. 서론

cQA(Community-based Question Answering) 시스템은 사용자들이 특정한 정보를 얻기 위해 질문을 등록하고, 해당 질문 분야에 대해 잘 알고 있는 사용자들이나 전문가들이 답변을 할 수 있는 시스템이다. 대표적인 cQA 시스템으로는 네이버 지식iN¹, Yahoo! Answers², Quora³ 등이 있다.

웹 환경의 발전으로 많은 사람들이 cQA 시스템을 통해 정보 커뮤니케이션 활동을 하게 되었고, 새로운 질문에도 비교적 짧은 시간 내에 답변을 받을 수 있게 되었다. 또한, 이렇게 축적된 질문-답변 쌍들은 Database 화 되어 동일한 정보 요구가 있는 또 다른 사용자들이 검색을 통해 정보 획득을 할 수 있도록 도움을 준다.

하지만, 시스템 사용 빈도가 높아지면서 새로운 질문의 인입량이 늘어나게 되고, 이에 따라 우선순위에 밀려 답변을 받지 못한 채 뒤로 밀려나는 질문들이 생기기도 한다. 또한, 검색을 통해 관련 정보를 얻으려 해도 키워드 매칭 기반의 검색으로는 단어는 다르지만 의미적으로

동일한 질문을 검색 하기가 쉽지 않다.

특정 질문이 작성되었을 때, 기존에 저장된 질문-답변 DB 에서 유사한 질문을 찾아 그에 해당하는 답변을 바로 줄 수 있다면, 답변 받지 못한 채로 남아있는 질문 비율은 많이 줄어들 것이다. 또한, 유사 질문 검색에 있어서, 단순 키워드 매칭을 넘어 의미적인 유사도도 판단할 수 있게 된다면 더 많은 질문에 대해 답변을 찾아 줄 수 있게 된다. 그리고, 특정 정보를 얻기 위해 단순 검색을 하는 사용자들도 키워드의 나열이 아닌 자연스러운 질문을 통해 원하는 답변을 얻을 수 있게 된다.

의미적으로 유사한 성질을 학습하기 위해서는 의미적으로 동일한 문장 쌍의 형태로 학습 데이터가 구성되어야 하지만, 이러한 데이터를 대량으로 확보하기는 쉽지 않다. 본 논문에서는 네이버의 대표적인 cQA 서비스인 지식iN에서 질문 제목과 질문 내용 데이터를 이용함으로써 이러한 한계를 극복하였다. 사용자들이 질문을 작성할 때, 질문 내용을 대표할 수 있는 문장으로 질문 제목을 작성한다. 질문 제목과 내용은 완벽하게 동일한 의미는 아니지만 제목이 내용에 대한 대표성을 띠고 있기 때문에 의사(pseudo) 유사 문장으로 간주하고, 이들의 유사도를 높게 학습함으로써 의미적인 성질을 부여하였다. 예를 들어, 그림 1과 같이 질문 제목에는 “하복부 통증”이라는 문구가 질문 내용에서 “아랫배가 아픈데” 라는 문구와 연결이 되어 있는 것을 볼 수 있다. 제목과 내용을 semi-training word embedding 과 CNN(Convolutional Neural Network) [1, 2] 을 이용하여 의미적 벡터 공간

¹ <http://kin.naver.com>

² <https://answers.yahoo.com>

³ <https://www.quora.com>

으로 사상하고, 이들을 벡터 공간상의 가까운 곳에 위치하게 함으로써 이러한 의미적 연결고리를 부여하였다. 새로운 질문이 주어지면 이를 DB에 저장된 질문들과 함께 의미적 벡터 공간으로 사상하여 벡터 유사도를 계산함으로써 질문 검색을 수행하였다. 본 논문의 실험 결과 키워드 매칭 기반 검색보다 딥러닝을 통해 의미 벡터 공간으로 사상 후 검색하는 방법이 더 좋은 결과를 얻을 수 있었다.

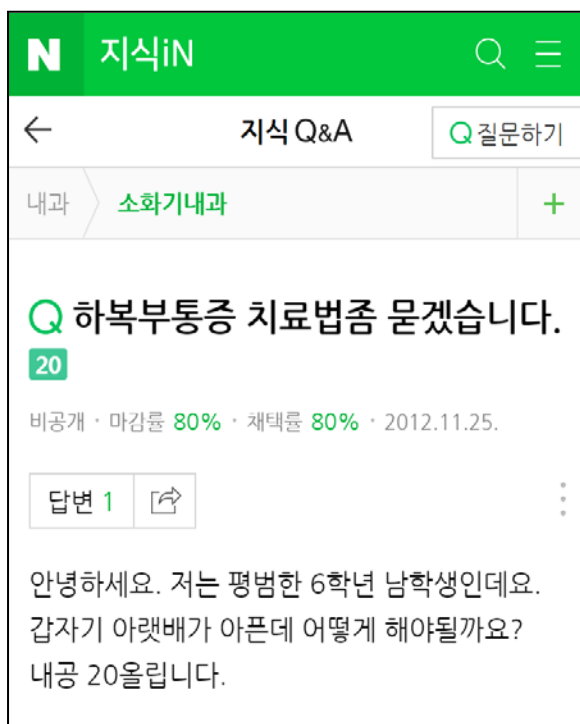


그림 1. 네이버 지식iN 질문 구성 예

2. 관련 연구

cQA 시스템은 질문-답변 DB에서 사용자의 신규 질문과 유사한 질문을 찾는 것이 중요하다. 유사한 질문을 찾는다면 해당 질문에 달린 답변을 신규 질문에 대한 답으로 제공할 수 있다.

질문간의 유사도를 구하는 방법으로는 TF-IDF, BM25, vector space model 등의 방법들이 연구되었다. 또한 질문간의 번역 확률로서 유사도를 계산하는 번역 기반 모델과 질문들 간의 topic 정보를 사용하는 topic model 방법들이 연구되었다. [3, 4]는 knowledge base로 Wordnet [5] 사전과 TF-IDF를 활용한 vector space model을 이용해 질문과 질문간의 유사도를 계산하였다. [6]에서는 단어와 단어 단위, [7]에서는 구문과 구문 단위 번역 모델을 이용하여 질문간의 유사도를 계산하였다. [8]은 단어 단위의 매칭에 topic model을 추가하여 질문 검색을 수행하였다. [9]에서는 topic과 질문에서의 핵심 부분(focus)을 추출한 뒤 이를 language model과 조합하여 질문을 검색하였다.

최근에는 이미지, 음성, 자연어 등 여러 도메인에서

각광을 받고 있는 딥러닝을 이용한 방법이 많이 연구되고 있다. [10]은 단어의 워드 임베딩 벡터와 카테고리 정보를 이용하여 유사도 계산의 feature로 사용하였다.

[11]은 CNN과 Bag-of-words의 도합으로 유사 질문을 검색하였다. 질의와 질문을 CNN을 통해 feature를 추출하고, 이들을 결합한 후 추가적인 MLP(Multi Layer Perceptron)를 통해 유사도를 계산하였다. [12]에서는 RNN(Recurrent Neural Network)의 기법중 하나인 LSTM(Long Short-Term Memory)을 통해 질문과 답변의 feature를 추출하고, 전통적인 자연어 기반 feature를 결합하여 추가적인 MLP를 통해 유사도를 계산하였다.

하지만 이런 방법은 질문과 질문간의 feature를 결합한 후 또다시 MLP 연산을 해야 하므로, 대량의 질문 셋에서 실시간으로 유사도를 추정하기가 쉽지는 않다.

본 논문에서는 질문 자체에 대한 벡터 표현(representation)을 추출함으로써 각 질문간의 벡터 거리 계산만으로 최종 유사도를 결정할 수 있도록 하였다. 또한, pre-trained된 word vector의 일부만 학습하고 일부는 고정된 채 사용하는 semi-training word embedding과 CNN을 결합하여 의미적인 연결을 더 강화할 수 있도록 하였다.

3. 의미 매칭을 이용한 질문 검색 모델

본 논문에서는 네이버의 대표적인 커뮤니티 기반의 질의응답 시스템인 지식iN 데이터를 이용하여 의미적 질문 검색 모델을 구성하였다. 질문을 의미 공간으로 사상하기 위하여 Semi-training Word Embedding과 CNN(SWECNN)을 이용하였고, 질문 제목과 질문 내용의 벡터 유사도를 높게 학습함으로써 의미적인 연결고리를 부여하였다.

3.1. 학습 데이터

문장간의 의미적 유사도를 판단하기 위해서는 유사 의미의 문장 데이터 셋이 존재해야 한다. 하지만, 이러한 데이터를 구축하기 위해서는 상당한 시간과 비용이 발생한다. 본 논문에서는 질문이 제목과 내용으로 구성된 지식iN 데이터를 이용하여 이러한 한계점을 극복하였다. 그림 1과 같이 질문 제목은 질문 내용에 대해 어느 정도의 대표성을 지닌다고 볼 수 있기 때문에, 이들을 의사 유사 문장 쌍으로 간주하였다. 그리고, 이들의 의미 공간 내 거리를, 질문 제목과 다른 임의의 질문 내용간의 거리보다 상대적으로 더 가깝게 되도록 학습함으로써 의미적인 연결고리를 부여하였다. 즉, 질문 제목에 대해 같은 쌍인 질문 내용을 positive sample로 설정하였고, 같은 쌍이 아닌 임의의 질문 내용을 negative sample로 설정하였다. negative sample은 질문 제목 당 4개를 사용하였다. 질문 제목과 질문 내용이 연관이 없는 내용을 가지고 있는 경우를 필터링하기 위하여 질문 제목과 질문 내용의 교집합 단어의 개수가 질문 제목 단어 개수의 절반이 안 되는 샘플들은 제거하였다.

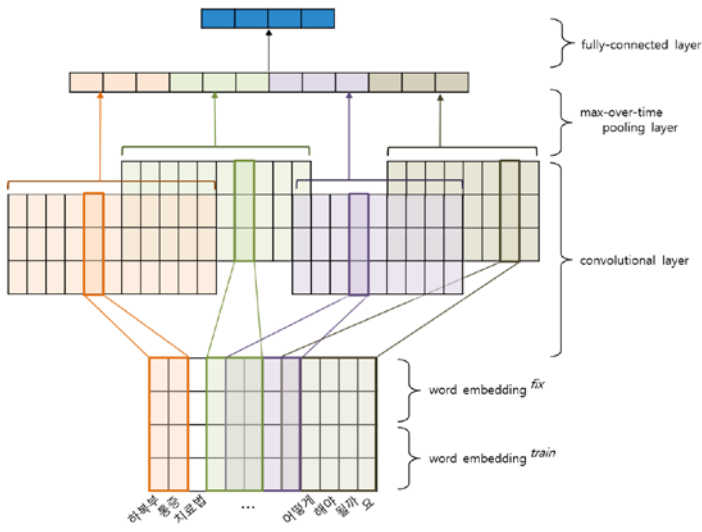


그림 2. 의미 매칭 모델(SWECNN). Pre-trained word embedding 두 개 중 하나는 고정하고(fix), 다른 하나는 fine-tuning(train) 하였다. Word embedding 된 값을 convolutional 연산을 이용하여 feature map 을 추출하고, 이를 max-over-time pooling 을 이용하여, 가장 높은 특징 값들을 추출하였다. 마지막으로 fully-connected layer를 통해 의미 벡터를 추출하였다.

3.2. word representation layer

Word embedding 을 이용하기 위해 별도의 코퍼스를 구축하였고, GloVe [13] 방법을 이용하여 300차원의 word embedding vector 를 학습하였다. 이렇게 학습된 동일한 word embedding vector 두 개를 식 (1) 과 같이 연결하여 semantic matching model 의 입력으로 사용하였다. 식 (1) 에서 \oplus 는 연결 연산이다.

$$x_i = x_i^{fix} \oplus x_i^{train} \quad (1)$$

i 번째 단어의 word vector $x_i^{fix} \in \mathbb{R}^{300}$ 는 학습이 끝날 때까지 vector 값을 고정하였고, i 번째 단어의 또 다른 word vector $x_i^{train} \in \mathbb{R}^{300}$ 은 의미 매칭 모델 학습 시에 fine-tuning 하였다. x_i^{fix} 와 x_i^{train} 을 연결해 구성된 최종 word vector $x_i \in \mathbb{R}^{600}$ 를 통해 기존의 워드 벡터가 가지고 있는 구조적, 의미적 특징은 물론, 본 논문의 데이터와 모델에 맞게 학습되는 문장 단위의 의미적인 정보도 내재될 수 있도록 하였다.

3.3. 의미 매칭 모델(SWECNN)

그림 2와 같이 Semi-training Word Embedding 과 CNN 을 이용한 SWECNN 의미 매칭 모델을 구성하였다. CNN 의 입력으로 질문의 각 단어 별 embedding vector x_i 를 사용하였고, convolution 연산을 위해 filter $w \in \mathbb{R}^{h \times 600}$ 를 사용하였다. h 는 convolution 연산의 window 사이즈

이고, window 만큼의 word embedding vector 가 filter 와의 연산을 통해 한 개의 feature 로 추출된다. 본 논문에서는 2, 3, 4, 5 의 window 크기를 갖는 4 종류의 filter 를 각 300개씩 사용하였다. Convolution 연산 후에는 vanishing gradient 에 좋은 효과를 보이는 Rectified linear unit (ReLU) f_{ReLU} 를 non-linear 함수로 사용하였다[15].

$$f_{ReLU}(x) = \max(0, x) \quad (2)$$

ReLU 함수 적용 후 max-over-time pooling 을 통해 가장 중요한 활성화 값(activation value) 만 남기도록 하였고, 4 종류의 filter 별로 300차원의 feature 를 추출하였다. 추출된 각 feature 들을 모두 인접시키고 1개의 fully-connected layer 를 연결하여 최종적인 300 차원의 semantic 질문 벡터를 추출하였다 (그림 2).

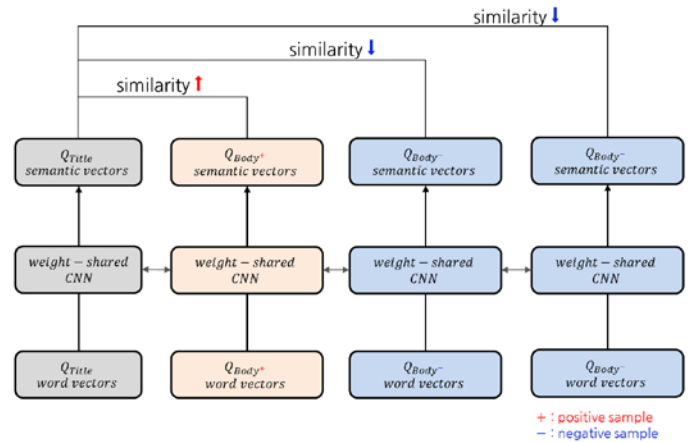


그림 3. Weight-shared network 을 이용한 시맨틱 매칭 모델의 학습. 질문 제목과 질문 내용의 word vector 를 weight 를 공유하는 CNN 의 입력으로 하여, 각 질문의 semantic vector 를 추출한다. 그리고, Q_{Title} 과 positive Q_{Body+} 의 벡터 유사도는 높게, Q_{Title} 과 negative Q_{Body-} 의 벡터 유사도는 낮게 학습함으로써, 의미 정보를 내재화하였다.

3.4. weight-shared network 를 이용한 학습

본 논문에서는 질문 제목 Q_{Title} 과 같은 쌍인 질문 내용 Q_{Body+} 를 positive sample 로, 질문 제목과 같은 쌍이 아닌 임의의 질문 내용 Q_{Body-} 를 negative sample 로 사용하였다. 그리고 각 질문들을 weight 를 공유[15] 하는 CNN 의 입력으로 사용하여 각 질문의 의미 벡터 y_{Title} 과 $y_{Body+/-}$ 를 추출하였다. 그리고 질문 제목과 positive 질문 내용의 유사도 $S(Q_{Title}, Q_{Body+})$ 를 높게, 질문 제목과 negative 질문 내용의 유사도 $S(Q_{Title}, Q_{Body-})$ 를 낮게 학습하였다 (그림 3). 유사도 S 는 식 (3) 과 같이 코사인 유사도를 이용하였다.

$$S(Q_{Title}, Q_{Body}) = \cos(y_{Title}, y_{Body}) = \frac{y_{Title}^T y_{Body}}{\|y_{Title}\| \|y_{Body}\|} \quad (3)$$

[16]과 같이, 질문 제목과 positive 질문 내용의 상대적 유사도를 식 (4)의 softmax 함수를 이용하여 posterior 확률로 간주하였다.

$$P(Q_{Body+} | Q_{Title}) = \frac{\exp(\gamma S(Q_{Title}, Q_{Body+}))}{\sum_{Body' \in Body} \exp(\gamma S(Q_{Title}, Q_{Body'}))} \quad (4)$$

식 (4)에서 **Body**는 모든 질문 내용의 집합이고, γ 는 softmax 함수의 smoothing factor이다. Loss 함수는 식 (5)와 같이 cross entropy 함수를 이용하였다.

$$Loss = -\log P(Q_{Body+} | Q_{Title}) \quad (5)$$

Loss 함수에 대한 gradient를 구하기 위해 back-propagation [17]을 이용했고, Stochastic Gradient Descent 방법 중 하나인 RMSProp [18]으로 모델 파라미터들을 학습하였다.

4. 실험 및 평가

본 논문에서 사용한 모델의 효과를 검증하기 위하여 약 230만건의 지식iN 질문-답변 문서를 추출하고, 실제 사용자가 수행하였던 질문들 중 임의의 200건에 대하여 얼마나 의미적으로 동일한 질문을 가진 문서들이 검색되는지 실험하였다. 의미 벡터 유사도를 이용한 검색 방법으로는 Approximate Nearest Neighbor [19] 검색을 이용하였다. 검색 결과 평가를 위하여 서로 상의하지 않은 세 명의 평가자가 평가 질문과 검색된 질문이 의미적으로 유사한지 아닌지 각각 판단을 하였다. 각 샘플은 최소 두 명의 동의를 받은 결과를 최종 결과로 간주하였다. 예를 들어, 특정 샘플에 대해 두 명의 평가자가 ‘유사하다’고 판단하고, 다른 한 명의 평가자는 ‘유사하지 않다’라고 판단했을 경우, 해당 샘플은 ‘유사하다’로 최종 판단 하였다. 모델 별 평가 기준으로는 P@K (Precision at K)를 사용하였다. 본 논문에서 사용한 신경망 모델과의 비교를 위하여 BM25와 pre-trained word vector들의 평균값(avg-emb)을 baseline으로 이용하였다. 추가적으로 제안 모델에서 사용한 semi-training word embedding 방법에 대한 효과 검증을 위하여 emb_300^{fix}, emb_300^{train}, emb_600의 방법과 비교를 하였다. emb_300^{fix}, emb_300^{train} 그리고 emb_600의 경우 제안 모델에 대한 word embedding layer의 변형된 모델이고 word embedding 이후의 CNN 및 fully-connected layer의 아키텍처는 동일하다.

- BM25 : 학습셋으로 BM25 모델을 구축하고, 평가 질문과 지식iN 질문간의 BM25 score를 계산한 후 가장 높은 질문들을 추출하였다.
- avg-emb : pre-trained 된 word vector에 대해, 평가 질문의 단어 벡터 평균과 지식iN 질문의 단어 벡

터 평균의 유사도를 비교하여 가장 score가 높은 질문들을 추출하였다.

- emb_300^{fix}+CNN: pre-trained 된 word vector를 사용하고 의미 매칭 모델을 학습할 때 word vector를 고정하였다.
- emb_300^{train}+CNN : pre-trained 된 word vector를 사용하고 의미 매칭 모델 학습시에 fine-tuning 하였다.
- emb_600+CNN : 제안 모델과 동일한 파라미터 개수로 비교하기 위하여 600 차원의 word vector를 설정하여 임의의 값으로 초기화 한 후 의미 매칭 모델 학습시에 같이 학습하였다.

의미 매칭 모델의 비교 평가 결과는 표 1과 같다. 기존 키워드 매칭 검색 방법인 BM25의 경우 가장 좋지 않은 성능을 보였고, 본 논문의 제안 모델과는 큰 성능 차이가 있었다. word embedding과 CNN을 통해 의미 벡터 공간으로 질문을 사상한 후 검색하는 방법이 키워드 매칭 방법보다 탁월함을 알 수 있었다. 또한 word embedding의 단순 평균만을 이용했던 avg-emb도 BM25보다는 좋은 결과를 얻을 수 있었다. 이는 단순 word embedding만으로도 각 단어가 내포하고 있는 의미를 잘 표현해줄 수 있다고 볼 수 있다. 그리고 word embedding을 처음부터 학습하는 emb_600보다는 pre-trained word embedding과 CNN을 결합한 방법들이 더 좋은 성능을 보였다. 이는 pre-trained word embedding이 의미 매칭 학습 데이터에서 얻지 못한 단어 별 의미 정보를 보유하고 있기 때문으로 분석이 된다. emb_300^{fix}와 emb_300^{train}의 비교 결과는 큰 차이는 없으나 emb_300^{train}이 근소하게 높은 성능을 보였다. 본 논문에서 제안한 SWECNN 모델이, P@1 기준 51.5%로 다른 비교 모델들보다 가장 우수한 결과를 보임을 확인할 수 있었다.

표 1. 유사 질문 검색 실험에 대한 결과

모델	P@1 (%)	P@3 (%)	P@5 (%)
BM25	32.5	25.3	20.9
avg-emb	42	34	28.9
emb_300 ^{fix} +CNN	47	35.8	30
emb_300 ^{train} +CNN	48	37.3	31.8
emb_600+CNN	43.5	33.8	27.9
SWECNN (proposed model)	51.5	40	33.2

표 2. 제안 모델과 다른 비교 모델간의 샘플 비교 결과

평가 질문 1. 발바닥 아픈데 어느 병원으로 가야 하나요?	
BM25	아랫배가 너무 아픈데 어느 병원으로 가야 하나요? 양쪽어깨가 아프고 종아리와 발바닥 통증이 있다면 어느과 병원을 가야 하나요?
SWECNN	발바닥 아플때는 어느병원에 가야 되나요 발바닥 통증...어디병원을 가야 할까요.
평가 질문 2. 겨드랑이 털은 보통 언제 나나요?	
word2vec	겨드랑이 레이저 제모 후 언제쯤 털이 안나나요? 팔 털 제모하면 털이 더 나나요?
SWECNN	겨드랑이 털나는시기 겨드랑이 털이 나기 시작하는데요...
평가 질문 3. 일본 동경에서 가볼만한곳은?	
emb_300 ^{fix}	도쿄 가볼만한곳 오사카 저녁시간에 가볼만한 곳
SWECNN	도쿄 가볼만한곳 일본 도쿄 가볼만한곳 추천좀해주세요
평가 질문 4. 햇빛 많이 받으면 머리아파요?	
emb_300 ^{train}	햇빛을 보면 머리가 아파요 요즘 머리가아파요
SWECNN	햇빛을 보면 머리가 아파요 여름에만 머리가 아파요
평가 질문 6. 비타민디가 들어있는 음식?	
emb_600	비타민 c가 들어있는 음식 무엇이 있는가요 비타민씨 풍부한 음식??
SWECNN	비타민 d 많은 음식 비타민D 가득한음식!

표 2는 제안된 모델과 다른 모델간의 비교 샘플이다. 평가 질문 1에서는 BM25를 이용한 키워드 매칭 방법과의 비교이다. BM25는 '아픈데', '어느 병원으로 가야 하나요' 등의 단순 키워드를 이용하여 검색을 하였지만, 검색된 질문은 평가 질문과 의미적으로 큰 차이를 보였다. 제안 모델은 '발바닥'에 대하여 '아픈데'와 '통증'을 의미적으로 연결하였다. 평가 질문 2에서는 avg-emb 모델과 비교하였다. avg-emb는 단순 word embedding만을 사용했기 때문에, 신체 일부인 '겨드랑이'와 '팔'이 가깝게 매칭되었고 검색된 질문은 평가 질문과 의미가 일치하지 않는다. 제안 모델은 '언제 나나요'와 '시기'에 대하여 의미적으로 바르게 연결이 되었다. 평가 질문 3에서는 emb_300^{fix}의 경우 '동경'과 '도쿄'가 잘 연결이 되기도 하였지만, pre-trained word embedding의 고정된 값만 사용해서 '도쿄'와 '오사카'가 유사하게 검색되기도 하였다. 평가 질문 4에서는 제안 모델의 경우 '여름에만 머리가 아파요'라는 질문이 검색되었다. 단순히 보면 의미적으로 일치하지 않지만 햇빛이 많은 계절이 여름이라는 것을 보면 '햇빛 많이'와 '여름'의 경우 어느 정도는 상호간에 의미가 연결되었음을 확인할 수 있었다. 평가 질문 5에서는 제안 모델의 경우 '비타민디'가 '비타민 d'로 연

결이 되었다. '디'와 'd'의 경우 pre-trained word embedding의 효과로 상호간 연결이 되었지만, 학습된 word embedding을 사용하지 않은 emb_600의 경우는 '디'와 'd'의 연결점을 찾지 못하였다.

모델 비교 실험 결과, 질문 제목과 내용을 이용하는 의사 의미 학습에는 pre-trained word embedding이 중요하며, 그 중 제안 모델에서 이용하는 semi-training word embedding의 성능이 탁월함을 확인할 수 있었다.

5. 결론

본 논문에서는 의미 매칭 모델링을 위한 대량의 학습 데이터를 구축하기 위하여 지식인 cQA 셋에서 질문 제목과 내용을 이용하는 방법을 제안하였다. 질문 제목과 내용을 각각 의미 벡터 공간으로 사상한 후, 이들의 거리를 가깝게 되도록 학습함으로써 의사 의미 연결고리를 부여하였다. 의미 벡터 추출로는 Semi-training Word Embedding과 CNN(SWECNN)을 이용하는 딥러닝 모델을 제안하였다. 모델 별 비교 실험 결과 SWECNN이 가장 좋은 성능을 보임을 확인할 수 있었다.

향후 연구로는, 단어 단위 데이터와 음절 또는 자소 단위의 데이터도 함께 이용함으로써 Out-of-vocabulary

의 한계를 극복하고자 한다. 또한, 카테고리과 같은 부가 정보를 추가적으로 이용한다면 더 정밀한 질문 검색을 할 수 있을 것이다.

참고문헌

- [1] LeCun, Y., Kavukcuoglu, K., & Farabet, C. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pp. 253-256. IEEE, 2010.
- [2] Kim, Y. Convolutional neural networks for sentence classification. In EMNLP, 2014.
- [3] Burke, R. D., Hammond, K. J., Kulyukin, V., Lytinen, S. L., Tomuro, N., & Schoenberg, S. Question answering from frequently asked question files: Experiences with the faq finder system. AI magazine, 18(2), 57. 1997.
- [4] Song, W., Feng, M., Gu, N., & Wenyin, L. Question similarity calculation for FAQ answering. In Semantics, Knowledge and Grid, Third International Conference on, pp. 298-301. IEEE, 2007.
- [5] Fellbaum, C. WordNet. John Wiley & Sons, Inc., 1998.
- [6] Jeon, J., Croft, W. B., & Lee, J. H. Finding similar questions in large question and answer archives. In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 84-90. ACM, 2005.
- [7] Zhou, G., Cai, L., Zhao, J., & Liu, K. Phrase-based translation model for question retrieval in community question answer archives. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 653-662. 2011.
- [8] Zhang, K., Wu, W., Wu, H., Li, Z., & Zhou, M. Question retrieval with high quality answers in community question answering. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management, pp. 371-380. ACM, 2014.
- [9] Duan, H., Cao, Y., Lin, C. Y., & Yu, Y. Searching Questions by Identifying Question Topic and Question Focus. In ACL, Vol. 8, pp. 156-164. 2008.
- [10] Zhou, G., He, T., Zhao, J., & Hu, P. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering. In ACL (1), pp. 250-259. 2015.
- [11] Dos Santos, C., Barbosa, L., Bogdanova, D., & Zadrozny, B. Learning hybrid representations to retrieve semantically equivalent questions. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 2, pp. 694-699. 2015.
- [12] Nassif, H., Mohtarami, M., & Glass, J. Learning semantic relatedness in community question answering using neural models. ACL, 137. 2016.
- [13] Pennington, J., Socher, R., & Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
- [14] Nair, V., & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807-814. 2010.
- [15] Chopra, S., Hadsell, R., & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 1, pp. 539-546. IEEE, 2005.
- [16] Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pp. 2333-2338. ACM, 2013.
- [17] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. Learning representations by back-propagating errors. Cognitive modeling, 5(3), 1. 1988.
- [18] Hinton, G., Srivastava, N., & Swersky, K. RMSProp: Divide the gradient by a running average of its recent magnitude. Neural networks for machine learning, Coursera lecture 6e. 2012.
- [19] Kalantidis, Y. and Avrithis, Y. Locally optimized product quantization for approximate nearest neighbor search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2321-2328. 2014.