Dense Retrieval에 기반한 유의어 추출 개선

Cognitive Computing Lab.

이정두°[1], 나승훈[1], 김관우[2], 이성민[2], 한상민[2]
[1]전북대학교, [2]삼성중공업
bo0od12@naver.com, nash@jbnu.ac.kr, {kwanwoo.kim, smvd.lee, sm5.han}@samsung.com

Ⅰ. 서론

유의어 추출(Synonym Extraction)이란 단어의 의미는 같지만 형태가 다른 단어를 추출하는 태스크이다. 질의 응답, 텍스트 요약, 생성 등 많은 자연어 처리 태스트에 적용될 수 있는 중요한 태스크이다. 본 논문에서는 중공업 특화 언어 모델을 기반으로 정보 검색에서 사용되는 방식인 MIPS와 Rank module 을 사용하여 유의어 추출하는 모델을 제안한다.

Ⅱ. 데이터

본 실험에는 삼성중공업에서 제공해준 용어들과 corpus 그리고 유의어 세트를 사용하였다.

표 1. 유의어 세트 예시

Set 1	Set 2	Set 3	Set 4
Accommodation	Zero Defect	WOL	자동복귀
Cabin	ZD	Weldolet	저전압 해방
Living Quarter	무결점 운동	용접식 돌기물	
거주구			

표 2. 데이터 통계

	Total	Train	Dev	Test
유의어 세트 수	6973	5805	-	-
Corpus에 나타나는 단어 수	6405	5473	335	597
총 단어 수	13943	11851	697	1395

Ⅲ. 제안 방법

Definition

$\mathcal{C}$: 중공업 전체 문장 집합
w: 단어
$S(w) \subseteq \mathcal{C}$: 단어 w 가 등장하는 문장 집합
$s \in S(w)$: 단어 w 가 등장하는 문장
$\boldsymbol{w}^c$: contextualized 벡터
$\boldsymbol{w}^d$: decontextualized 벡터
$\boldsymbol{w}$: 단어 벡터

Synonym extraction Module

Synonym extraction module 은 $\boldsymbol{w}$ 간의 Maximum Inner Product Search 를 통해 유의어를 찾는다. $\boldsymbol{w}$ 는 그림 1 과 같이 w 가 나타난 문장들 그리고 w 를 사전학습 언어모델을 이용하여 인코딩 후 단어의 각 토큰들을 average pooling 을 통해 하나의 벡터로 만든 후 $\boldsymbol{w}^c$ 와 $\boldsymbol{w}^d$ 이 두 벡터를 결합하여 얻는다.

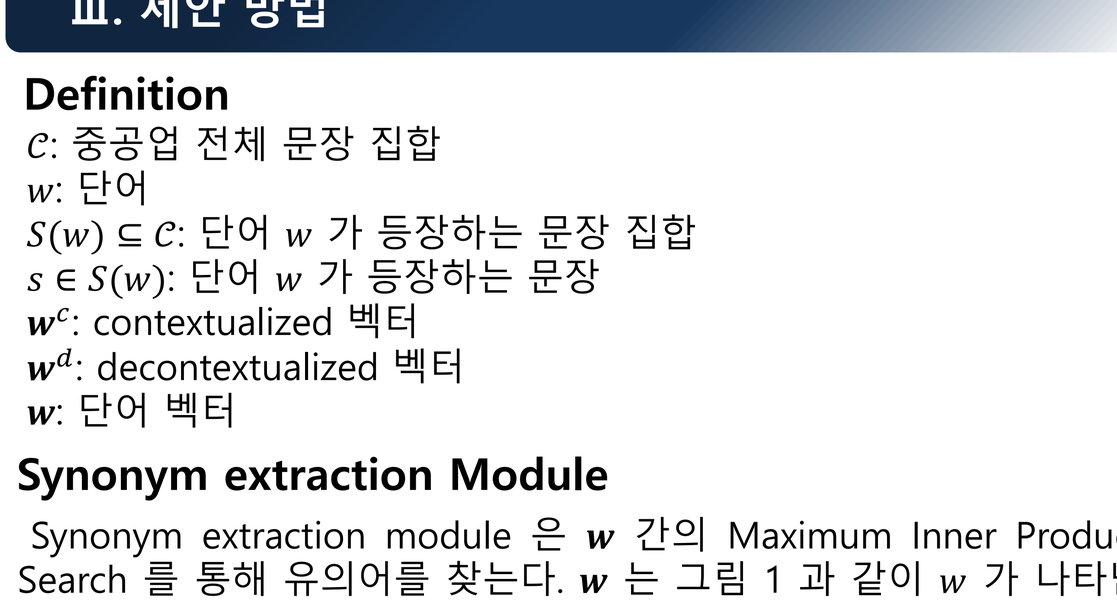

그림 1. Synonym extraction module 구조

$\boldsymbol{w}$ 를 얻는 과정을 수식으로 나타내면 아래와 같다.

$$\boldsymbol{w} = concat(\boldsymbol{w}^c, \boldsymbol{w}^d)$$

$$\boldsymbol{w}^c = \frac{1}{|S(w)|} \sum_{s \in S(w)} MLP(HeavyRoBERTa_{span_w}(s))$$

$$\boldsymbol{w}^d = MLP(HeavyRoBERTa(w))$$

여기서 $concat(,)$는 벡터간의 결합, MLP는 Multi-Layer Perceptron, $HeavyRoBERTa$ 는 인코더로 사용된 사전학습 언어모델, $span_w$는 문장 내에 단어의 위치를 의미한다.
위 모듈을 통한 두 단어의 유의어 점수는 아래 수식과 같다.

$$score_s = \sigma(\boldsymbol{w}^T \boldsymbol{w}')$$

여기서 $\boldsymbol{w}'$은 w가 아닌 다른 단어의 벡터, σ 는 sigmoid function 을 의미한다.

Rank Module

Rank module 은 synonym extraction module 을 통해 얻은 Topk 개의 유의어 후보를 대상으로 재순위화를 진행한다. 이 모듈은 입력 $\mathrm{x} = [\mathrm{SOS}]w[\mathrm{EOS}][\mathrm{EOS}]w'[\mathrm{EOS}]$ 를 입력으로 받고 [SOS] 토큰의 벡터를 이용하여 score를 계산한다. 수식은 아래와 같다.

$$score_r = \sigma(MLP\left(HeavyRoBERTa_{[SOS]}(\mathrm{x})\right))$$

두 단어의 유의어 최종 score 는 synonym extraction module 과 rank module 의 두 score 를 선형 결합을 통해 계산된다. 식은 다음과 같다.

$$score = \lambda\, score_s + (1-\lambda) score_r$$

여기서 λ 는 dev set 으로 결정되는 하이퍼 파라미터이다.

Ⅳ. 실험 결과

본 논문 평가 지표는 F1 score 를 사용하고, 평가 시 test 데이터의 단어와 유의어인 단어를 기존 전체 단어 집합에서 추출한다.

표 3. 유의어 추출 성능

	Synonym Extraction Module	+Rank Module
λ	-	0.1
Threshold	0.97	0.701
Precision	26.28%	**54.34%**
Recall	31.15%	**40.10%**
F1	28.51%	**46.15%**

본 실험에서는 계산 비용을 줄이기 위해 $\boldsymbol{w}^c$ 계산 시 사용되는 문장은 $S(w)$ 에서 무작위로 5개를 선택하여 사용하였고, Synonym Extraction module 에서 Top300개를 재순위화 하였다. 이때 TopK 의 개수와 λ, 그리고 threshold 값은 모두 dev 상에서 결정한다.
표 3에서 보여주듯이 최종 유의어 추출 시 rank module 에 더 의존하지만 rank module 의 경우 새로운 단어에 대한 score 를 계산 시에 기존 모든 단어들과 조합으로 cross-encoding 을 수행하기 때문에 시간 복잡도가 $\mathcal{O}(n^2)$으로 높지만 synonym extraction module 은 $\mathcal{O}(n)$ 으로 훨씬 빠르다는 장점이 있다. 따라서 synonym extraction module 을 통해 topk 개를 먼저 가져오고 rank module 을 적용하므로써 성능은 향상 시키고 속도 또한 합리적인 시간 내에 계산하도록 하였다.

Ⅴ. 결론

본 논문에서는 유의어를 추출하기 위해 정보 검색에서 활용되는 MIPs 와 Rank module 을 적용한다. 단순히 MIPs 만을 이용할 때 보다 rank module 을 추가 적용했을 때 큰 성능 향상을 보였다.