

리뷰 딥러닝 감성 분석 및 답글 생성기

산업공학종합설계

12174762 양권민
12174764 이태우
12151911 이길연
12184810 김경엽
12184813 김지은



1.

주제 선정 배경



ifeli

Enhance your life

- 교내 스타트업(2018~)
- 재정 여건상 CS직원의 잦은 변경



주제 선정 배경



★★★★★ 5
vmf**** · 21.11.10. | 신고
종류: 저마찰 / 색상: 블랙

아직 오래 써봐야 알겠지만 팬장은 잘 같아요
사진 오른쪽은 원래 펜슬촉입니다)
펜촉 커버로 쓰면 실리온이라 손목이 아프고 구멍나는게 싫어서 사봤는데 마찰소리도 안나고 벗겨지지 않으니 팬참지 않을까 싶네요 * 배송은 2-3일 정도 걸린것같아요

ifeli 판매자 21.11.12. | 신고

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님의 감사의 인사 드립니다
고객님께서 만족하신다면 그것 또한 저희의 기쁨입니다 😊
앞으로도 아이펠리와 함께 해주시기를 바랍니다 🍀...

더보기 ▾

★★★★★ 4
grim**** · 21.11.18. | 신고
종류: 기본 / 색상: 블랙

생각했던 감촉은 아니지만 마찰력이 생기니까 미끄럽지 않아서 좋아요~

ifeli 판매자 21.11.19. | 신고

안녕하십니까, 아이펠리 입니다.
아이펠리를 이용해주시고 관심을 가져주시는 고객님께 진심으로 감사 드립니다
고객님께서 제품에 만족감을 느끼신다 하니 다행입니다 😊
앞으로도 아이펠리에 대한 큰 관심과 응원 부탁드립니다 🍀...

더보기 ▾

★★★★★ 5
aaaa**** · 21.11.28. | 신고
종류: 저마찰 / 색상: 화이트

진짜 좋아요 저는 속속 거리는 느낌을 좋아하는데 생편촉은 달아지고 그래서 저마찰 구매를 했는데 그 속속 느낌이나요 세워 쓰면 작게 소리가 나는데 높여서 쓰니까 그 질감이 나서 좋아요 저는 스x코 종이질감인데 너무 만족합니다 * *

ifeli 판매자 21.11.29. | 신고

안녕하십니까, 아이펠리 입니다.
아이펠리를 이용해주시고 관심을 가져주시는 고객님께 진심으로 감사 드립니다
고객님께서 제품에 만족감을 느끼신다 하니 다행입니다 😊
앞으로도 아이펠리에 대한 큰 관심과 응원 부탁드립니다 🍀...

더보기 ▾

★★★★★ 5
yuns**** · 21.11.27. | 신고
종류: 아이펠리 세트 할인(SALE%) / 색상: 저마찰 SET(저마찰화이트1팩+저마찰블랙1팩)

독서실에서 소음때문에 실리온 커버 사용하다가 글씨가 끊기는 것 때문에 샀어요
실리온 커버보다는 소음 있지만 필기감이 더 좋아요!
소음 공감하신분들은 영상참고하셔도 좋을 것 같아요

ifeli 판매자 21.11.29. | 신고

안녕하십니까, 아이펠리 입니다.
아이펠리를 이용해주시고 관심을 가져주시는 고객님께 진심으로 감사의 인사 드립니다.
고객님께서 제품에 만족감을 느끼신다 하니 다행입니다 😊
앞으로도 아이펠리에 대한 많은 관심과 애정 부탁드립니다 🍀

언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다 😊
행복한 하루 되십시오.

- CS담당 직원의 주요 업무인 쇼핑몰 리뷰 대응
- 대응 방식이 표준화가 되어 있지 않아 신규 직원 인수인계에 어려움

주제 선정 배경

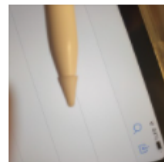


★★★★☆ 4

star**** · 21.12.01. | 신고

종류: 기본 / 색상: 화이트

실리콘이 일주일만에 찢어졌어요ㅠㅠ 무소음이어서 좋긴한데 내구성.이 아쉽습니다



판매자 21.12.01. | 신고

안녕하십니까, 아이펠리 입니다.

아이펠리에서 제품을 구매하여 주시어 감사 드립니다.

제품 이용에 불편을 드리게 되어 죄송합니다.

팁 펜촉의 겉 면은 실리콘으로 제작 된 상품인 관계로 사용자의 사용환경(필름/팔압/사용량 등)에 따라 내구성 및 사용 기한은 차이가 발생할 수 있는 점 안내 드립니다.

고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 좋은 제품 판매할 수 있도록 노력하겠습니다.

항상 고객 만족을 위해 노력하는 아이펠리 되도록 하겠습니다.

소중한 리뷰 감사드립니다😊.

접기 ^

- 평점은 높으나 내용이 부정적인 경우를 판단하고 대응 필요



안녕하십니까, 아이월리입니다.
아이월리에서 제품을 구매하여 주셔서 감사 드립니다.
팁 커버는 내부에 미세한 공기층이 있는 것이 정상 제품입니다. 올바르게 착용되어 있지 않을 경우 불만족(필기함 하라, 빠른 파손, 인식불가) 및 커버의 바른 착용을 위한 가이드 영상: <https://youtu.be/eqYfplH28Uk> 을 공유하여 드립니다.
공유하여 드린 영상을 참고하시어 제품을 다시 한번 착용하여 주신 후 제품을 사용하여 주십시오 ㉠
고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 좋은 제품 판매할 수 있도록 노력하겠습니다.
항상 고객 만족을 위해 노력하는 아이월리 되도록 하겠습니다.
소중한 리뷰 감사드립니다㉡.

안녕하십니까, 아이월리입니다.
아이월리에서 제품을 구매하여 주셔서 감사 드립니다.
공유하여 주신 영상을 확인하여 보니 제품이 잘 끼워진 채로 이용을 하신 것으로 보입니다. 팁 커버는 내부에 미세한 공기층이 있는 것이 정
범 커버의 바른 착용을 위한 가이드 영상: <https://youtu.be/eqYfplH28Uk> 을 공유하여 드립니다.
공유하여 드린 영상을 참고하시어 제품을 다시 한번 착용하여 주신 후 제품을 사용하여 주십시오 ㉠
고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 좋은 제품 판매할 수 있도록 노력하겠습니다.
항상 고객 만족을 위해 노력하는 아이월리 되도록 하겠습니다.
소중한 리뷰 감사드립니다㉡.

안녕하십니까, 아이월리입니다.
아이월리에서 제품을 구매하여 주셔서 감사 드립니다.
팁 펜촉의 끝 면은 실리콘으로 제작 된 상품인 관계로 사용자의 사용환경(결빙/활발/사용량 등)에 따라 내구성 및 사용 기간은 차이가 발생
고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 튼튼히 개선된 제품 판매할 수 있도록 노력하겠습니다.
항상 고객 만족을 위해 노력하는 아이월리 되도록 하겠습니다.
소중한 리뷰 감사드립니다㉡.

안녕하십니까, 아이월리입니다.
아이월리에서 제품을 구매하여 주셔서 감사 드립니다.
화장 사탕의 개선이 진행 될 수 있도록 고객님의 취사에 대해서 내부적으로 공유하도록 하겠습니다 ㉠
앞으로 더욱 좋은 제품 판매하여 항상 고객 만족을 위해 노력하는 아이월리 되도록 하겠습니다.
소중한 리뷰 감사드립니다㉡.
좋은 하루 되십시오.

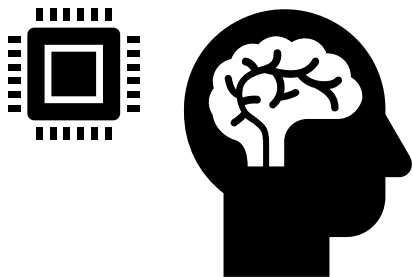
안녕하십니까, 아이월리입니다.
아이월리에서 제품을 구매하여 주셔서 감사 드립니다.
+ 서비스 제공 메뉴얼 문의개별도움 문의형틀링 문의형틀링? 4.5점 만족리뷰 2.3.4 불만족리뷰

- 지금까지 쌓아온 대응 매뉴얼 데이터를 답글 생성 기준으로 활용



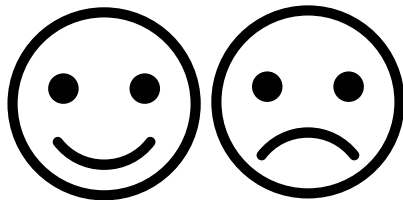
주제 선정 배경

딥러닝 모델 학습



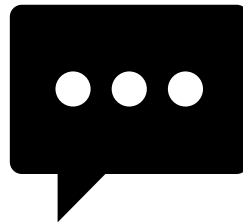
- 네이버 쇼핑 리뷰 20만개
- GRU 모델 사용

자연어 감성 분석



- 문장 입력하면 긍정 % 예측
- 90% 이상의 정확도 목표

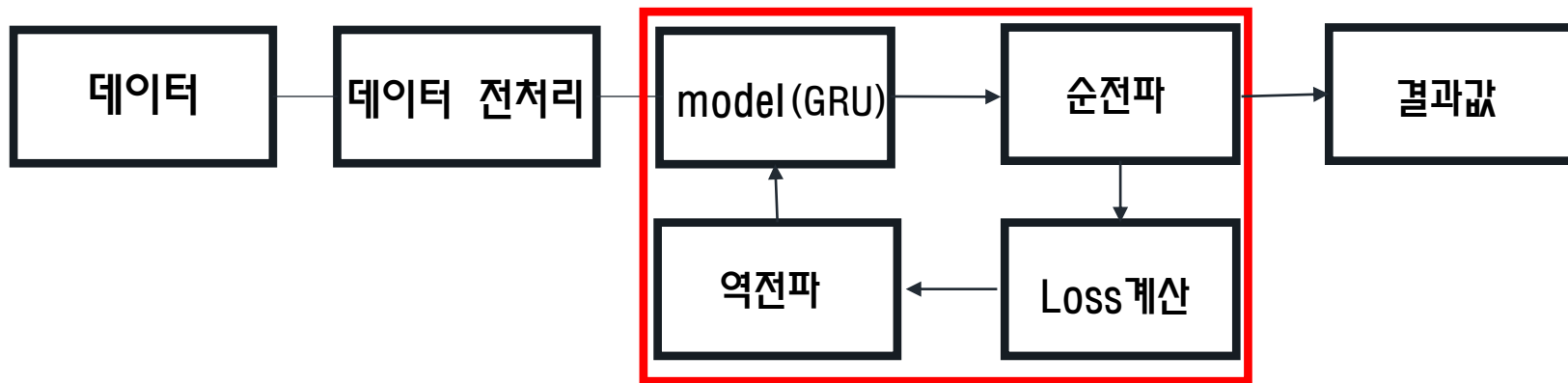
답글 자동 생성



- 문장의 감성 수치 고려
- 특정 단어 고려(ex: 배송)

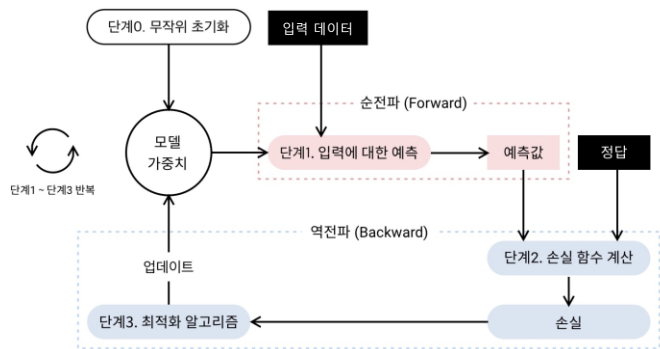
2.

분석 과정



답러닝

답러닝 과정

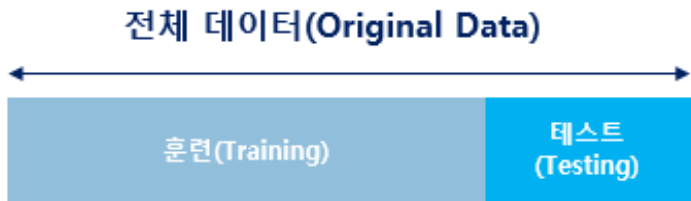


데이터 학습 과정

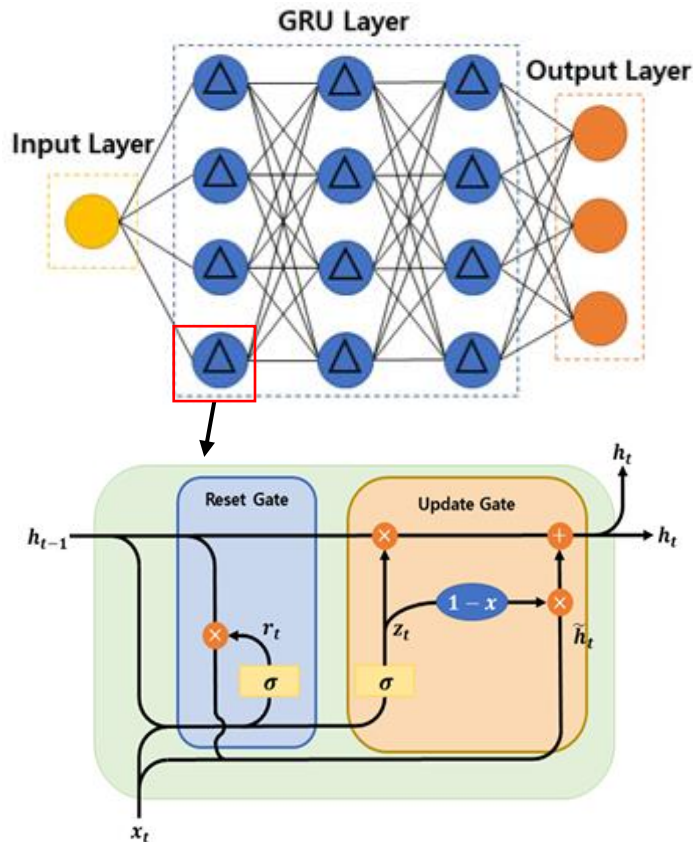
1. 딥 러닝 모델이 리뷰를 감성분석에 따라 긍정과 부정으로 분류하게 하기 위해 훈련 데이터를 통해 학습시킴
2. 테스트 데이터를 모델에 넣어 예측 값 도출하고 실제 값 사이의 오차(Loss 계산)를 구해 오차를 줄일 수 있는 가중치 업데이트
3. 모델의 예측 값과 테스트 데이터의 실제 결과 값을 비교해서 모델의 성능 테스트

데이터를 나누는 이유

1. 모델을 특정 데이터로 학습 시킴
2. 학습이 완료된 모델에 같은 데이터를 넣어 테스트를 진행
3. 데이터를 나누지 않으면 모델의 실제 성능과 관계없이 완벽한 예측 값을 보임(오버피팅 현상)



| GRU 모델



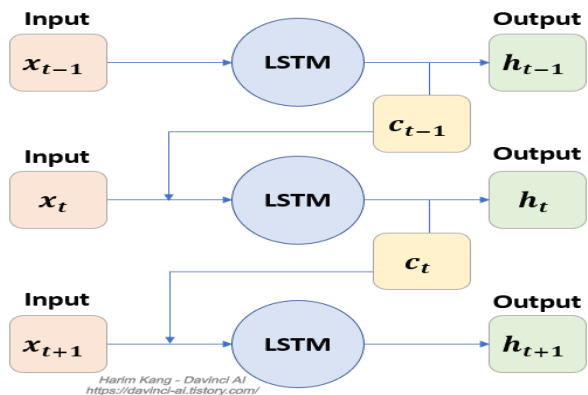
GRU 네트워크 구성

1. Input Layer에서 순차 데이터를 입력 및 정규화
2. GRU Layer에서 무작위로 설정된 가중치와 전달받은 데이터를 연산하여 예측데이터 생성 (순전파)
3. 예측 데이터와 실제 응답 데이터의 오차(loss)를 비교하고 가중치를 수정해가며 최적의 가중치 도출 (역전파)

GRU 알고리즘

1. GRU는 LSTM과 비슷하지만, 더 간단한 구조로 이루어져 있어 계산상으로 효율적
2. Reset Gate에서 과거의 데이터를 sigmoid 연산을 통해 리셋
3. Update Gate에서 현재점에서 가져갈 데이터를 결정

| LSTM과 GRU 비교

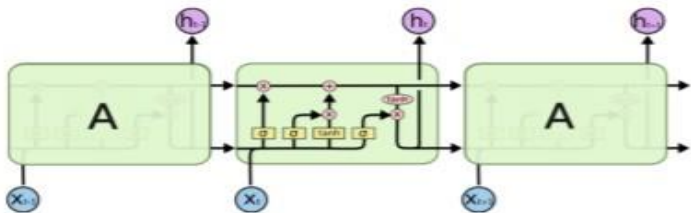


LSTM

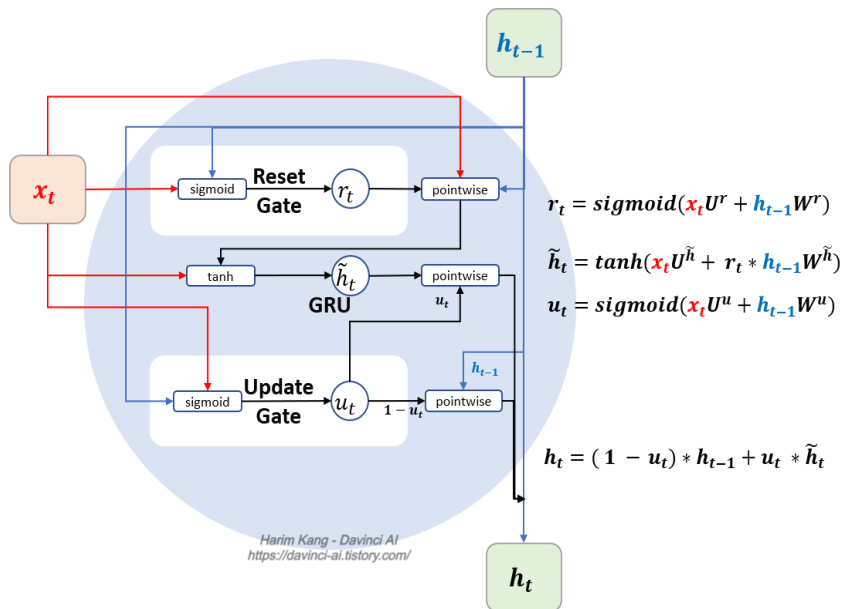
1. 출력 값 이외에, LSTM셀 사이에서 공유되는 셀의 상태(cell state)가 존재한다.
2. 복잡한 형태의 식을 가지며 은닉계층이 메모리 셀이라 불리는 구조를 지닌다.
3. 각 게이트마다 사용하는 가중치의 양이 많아 모형 학습 속도가 느리고 메모리가 많이 차지될 가능성이 있다.

GRU

1. LSTM보다 간단한 구조로 이루어져 있어 계산상 효율적이다.
2. Cell과 output gate가 없는 대신 update gate와 reset gate가 있어 메모리가 많이 차지될 가능성이 없다.



GRU



1. 각 게이트들은 sigmoid layer와 pointwise곱 연산을 통해 값을 제어
2. 과거의 출력 값을 그대로 사용하지 않고, 리셋 데이터로 출력한 값을 이용하여 pointwise 곱 연산
3. 연산 결과를 tanh 함수를 거쳐 값을 출력하여 이전의 h에서 연산한 값을 합한 값이 출력
4. 이전 정보는 되도록 무시하고 끝 부분의 정보를 최대한 유지

| 원본 데이터

[illegible]

20만개의 리뷰 데이터 사용

1 선물준사람도 발문조라도 모두 만족했어요~
2 예전고 선물해서 쓰는데, 재질이 팔고자위요, 삶에 자극 다 빼이고ㅠㅜㅜㅜ, 목감이에요. 재제도 확인안한 채 잘못이죠!! 참고로 실금제2, 2센테1 있습니다.
3 고집이 앞집기되는 하는데 많이 소근 욕진저 처음 보일까봐 걱정되는 정도 쏘았어요
4 만약이후 조금은 나아지, 카네일 물고있음
5 진작이 카네일 물고있는데 열수없어하고 참았지만 밑에 세상이 금방 안되서 AS이요고 물어뒀는데 전화도 안받고 10시미 흑물보냈는데 오후6시미 답해서 뭐하냐 물어보고하니 상담시간이요!이라고 쓰네요 발인ass를 해주세요 안되면 안원다고 답장부탁드립니다
6 좋아요 ㅎㅎㅎ 좋고 포만감 이불
7 네이비 색상이 조금 진한데 하지만 국
8 역사 기간은 보람이 있는 제품이군요!!
9 팔린다 가 줄어든다고 생각되는것 같아요
10 약간 벅줄게 생각되네요.. 반쯤 귀찮아서 그냥 씁니다.
11 미루 드릴줄 더 많이 올라오는데 용에 사용했는데 용에요 흐루지내요!!
12 전 이틀에 약속때문에 못할 컸다! 할때 열등고 그물에 너무 심한고 컨디션이 안 좋아져서 결국 불기처럼 편해요. 사람마다 다르니 사기전에 신중해 결정하세요. 참고로 전 팔발입니다.
13 아예 이틀에 후기가 사용하니까요! 후기가 많고 인기 후기 많습디다! 사이조는 다양해서 너무 좋고요 살린것과같이 관리해 줄게요!
14 사이조가 예뻐요!! 오마자차 짜 먹었는데 팔도 팔도
15 음썹썹해서 메가 좋아하길래요!
16 잘안되다 너무 작아서요 나중에 먹어먹는식물이나 넣어있었어 물론은 습습습!
17 세척하기 어렵음, 처음사올때 맨 세척기가기 앞에서 돌리고 했는데 2/3 사용했는데 너무 안헹거여, 지면살아 안헹는건지.... 다시 쓰면 고쳐줄테요 알아있습니다.
18 수습작이 너무 약해요..ㅠ 식단은 잡풀는데 이거는 메기가 자주 데요
19 제우며 환구제호는 사개로 이호는제거 자연현 경관조가 같아서요 그 색에는 기운만 열등 옅고 걸려서 다 팔렸네요.. 좋은상품주시조고 고객만약 일으으네요 ㅠ 이제 이용안하려고요
20 연후대 사것보다 만원짜리 지것보다 한달남서서 쓰네요 ㅎㅎㅎ 안되고 약한데도 안적익고 살품은 습습나 너무 늦게와서 팔도
21 제우며 만원짜리 만원짜리

| 데이터 전처리 과정

데이터 정의

- 평점 4,5점 리뷰는 긍정을 의미하는 1, 그 외는 부정의 0으로 라벨링
- 중복되는 데이터 및 Null 값과 한국어가 아닌 모든 값 제거하고 훈련 데이터와 테스트 데이터로 분리

토큰화

- Konply 라이브러리의 okt 형태소 분석기 사용
- 불용어를 지정하여 불필요한 토큰을 제거

벡터화

- 토큰을 단어의 빈도수로 정수 인코딩 후 패딩
- 워드 임베딩을 통해 각 단어들로 구성된 문장을 다 같은 차원의 벡터로 만들어 주고 각각의 단어에 수치를 부여하여 유사치를 표현

토큰화

* Tokenization

문장1 Severe storms strike the southeast Saturday.

문장2 DNA is not randomly folded inside the nucleus.

Token

Feature*Sample Table

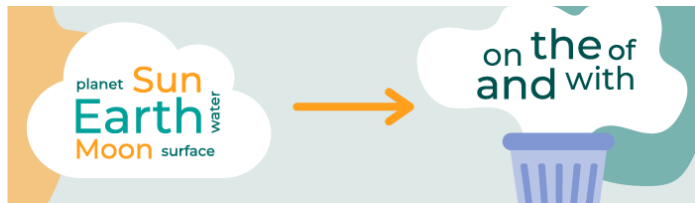
	The	Inside	Is	Not	...	DNA	Class
문장1	1	0	0	0		0	Weather forecast
문장2	0	1	1	1		1	Biology paper

단어 토큰화

1. 토큰이란 문법적으로 더 이상 나눌 수 없는 언어요소를 의미
2. 주어진 리뷰 데이터에서 토큰의 기준을 단어로 함
3. Okt 형태소 분석기를 통해 단어 토큰화 진행
4. 결과로 데이터가 단어기준으로 나누어짐

불용어

1. 갖고 있는 데이터에서 유의미한 단어 토큰만을 선별하기 위해 시행
2. 무의미를 판단하는 기준은 자주 등장하지만 분석을 하는 것에는 도움이 되지 않는 단어들
3. 불용어를 정의하여 데이터에서 제거



| 불용어 지정

1	아	
2	쥬	
3	아이구	
4	아이루	
5	아이고	
6	어	
7	나	
8	우리	
9	저희	
10	따라	
11	의해	
12	을	
13	들	
14	에	
15	의	
16	가	
17	으로	
18	로	
19	에게	
20	뿐이다	
21	의거하여	
22	근거하여	
23	일각하여	
24	기준으로	
25	예하면	
26	예를 들면	
27	예를 들자면	
28	저	
29	소인	
30	소생	
31	저희	
32	지말고	
33	하지마	
34	하지마라	
35	다른	
36	물론	
37	또한	
38	그리고	
39	비할수 없다	
40	해서는 안된다	
41	뿐만 아니라	

...

3708	격감	
3709	갯조워	
3710	개음	
3711	게시	
3712	게르	
3713	것갯	
3714	집이	
3715	걸음	
3716	걸원	
3717	건의	
3718	거지	
3719	거즈	
3720	거주	
3721	거여	
3722	거슬	
3723	거부	
3724	거구	
3725	거랑	
3726	겐자들	
3727	개취	
3728	개름	
3729	개삿는데	
3730	개살결그렸을	
3731	개보	
3732	개막	
3733	개라노	
3734	개들었어	
3735	갈습다	
3736	각습니	
3737	감습	
3738	감쥬습	
3739	감로	
3740	감긔	
3741	감겨	
3742	간해	
3743	간사이	
3744	간고	
3745	각력	
3746	가즈	
3747	가이드라인	
3748		

3700여개의 불용어 지정 후
불필요한 토큰 제거

※의미 분석에 도움이 되지 않는
불용어 직접 지정 제거※

| 단어와 길이 분포 확인

```
In [17]: negative_words = np.hstack(train_data[train_data.label == 0]['tokenized'].values)
         positive_words = np.hstack(train_data[train_data.label == 1]['tokenized'].values)

In [18]: negative_word_count = Counter(negative_words)
         print(negative_word_count.most_common(20))

[('너무', 14043), ('배송', 9890), ('그냥', 9115), ('했는데', 6030), ('별로', 5982), ('제품', 5422), ('으로',
5375), ('생각', 5326), ('사용', 5197), ('ㅠㅠ', 5138), ('하고', 5011), ('보다', 4946), ('구매', 4914), ('입
니다', 4126), ('많이', 4112), ('주문', 3876), ('가격', 3869), ('재구매', 3637), ('부분', 3579), ('해서', 352
1)]

In [19]: positive_word_count = Counter(positive_words)
         print(positive_word_count.most_common(20))

[('좋아요', 15412), ('배송', 13106), ('너무', 9690), ('재구매', 9251), ('구매', 7636), ('사용', 5748), ('가
격', 5351), ('같아요', 4749), ('으로', 4712), ('좋네요', 4427), ('보다', 4257), ('빠르고', 4256), ('제품', 4
237), ('주문', 4041), ('입니다', 3973), ('생각', 3737), ('했는데', 3558), ('좋습니다', 3537), ('했어요', 350
1), ('ㅎㅎ', 3408)]
```



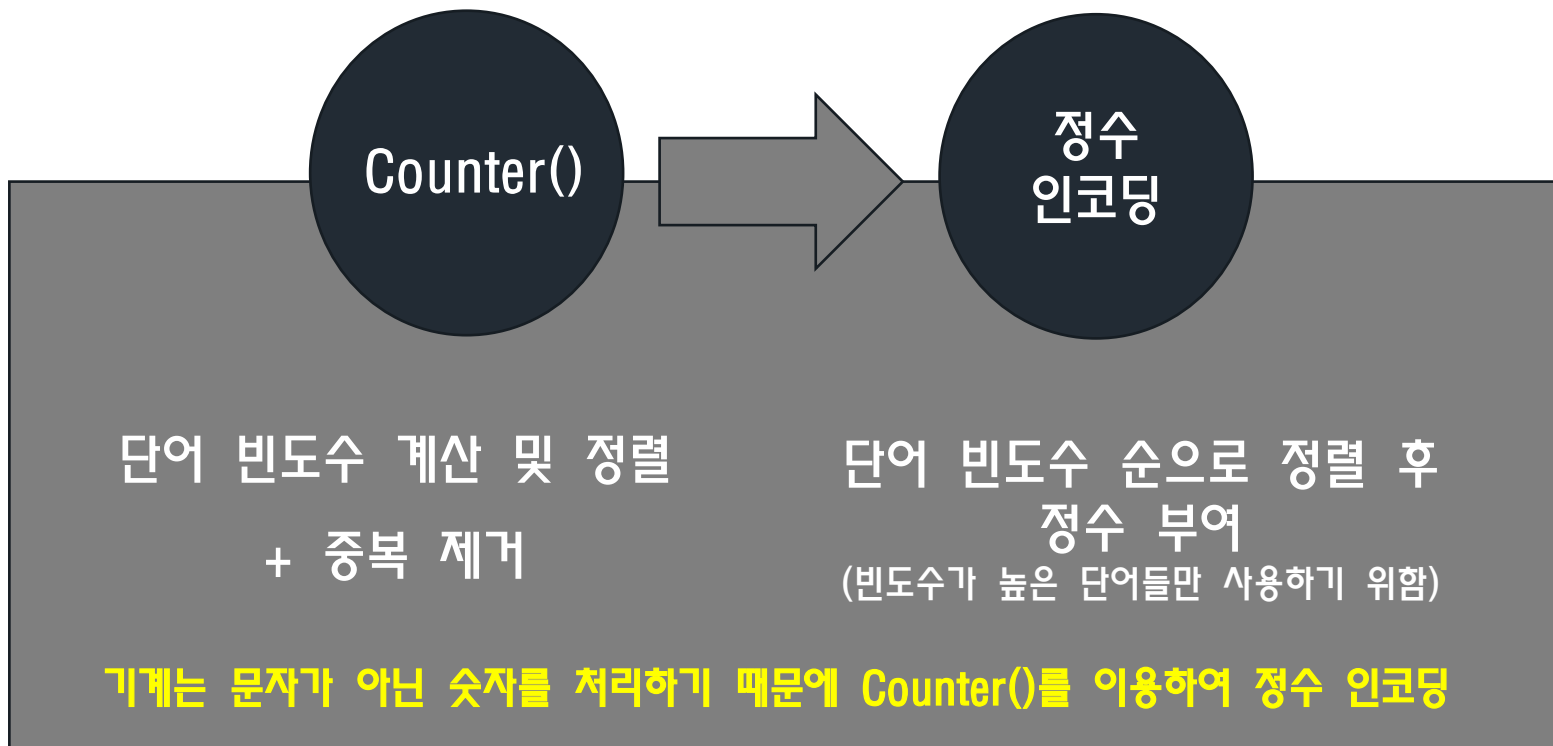
정수 인코딩을
하기위해 counter()
이용하여 각 단어
빈도수 계산

-긍정 리뷰와 부정 리뷰에 등장하는 각 단어의 빈도수 계산

-Counter()함수 사용하여 각 단어에 대한 빈도수 카운트

+긍정, 부정 리뷰에 대해서 빈도수 높은 상위 20개 단어 출력

| Counter이용한 정수 인코딩



| 정수 인코딩

```
In [23]: threshold = 2
total_cnt = len(tokenizer.word_index) # 단어의 수
rare_cnt = 0 # 등장 빈도수가 threshold보다 작은 단어의 개수를 카운트
total_freq = 0 # 훈련 데이터의 전체 단어 빈도수 총 합
rare_freq = 0 # 등장 빈도수가 threshold보다 작은 단어의 등장 빈도수의 총 합

# 단어와 빈도수의 쌍(pair)를 key와 value로 받는다.
for key, value in tokenizer.word_counts.items():
    total_freq = total_freq + value

    # 단어의 등장 빈도수가 threshold보다 작으면
    if(value < threshold):
        rare_cnt = rare_cnt + 1
        rare_freq = rare_freq + value

print('단어 집합(vocabulary)의 크기 :', total_cnt)
print('등장 빈도가 %s번 이하인 희귀 단어의 수: %s'%(threshold - 1, rare_cnt))
print('단어 집합에서 희귀 단어의 비율:', (rare_cnt / total_cnt)*100)
print('전체 등장 빈도에서 희귀 단어 등장 빈도 비율:', (rare_freq / total_freq)*100)
```

단어 집합(vocabulary)의 크기 : 95779
등장 빈도가 1번 이하인 희귀 단어의 수: 53846
단어 집합에서 희귀 단어의 비율: 56.21900416584011
전체 등장 빈도에서 희귀 단어 등장 빈도 비율: 3.4299742780594014

```
In [24]: # 전체 단어 개수 중 빈도수 2이하인 단어 개수는 제거.
# 0번 패딩 토큰과 1번 OOV 토큰을 고려하여 +2
vocab_size = total_cnt - rare_cnt + 2
print('단어 집합의 크기 :', vocab_size)
```

단어 집합의 크기 : 41935

```
In [25]: tokenizer = Tokenizer(vocab_size, oov_token = 'OOV')
tokenizer.fit_on_texts(X_train)
X_train = tokenizer.texts_to_sequences(X_train)
X_test = tokenizer.texts_to_sequences(X_test)
```

-훈련 데이터와 테스트 데이터에
정수 인코딩 수행

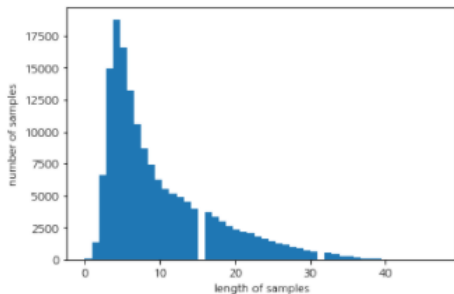
-토큰나이저는 텍스트 시퀀스를
숫자 시퀀스로 변환

-높은 빈도수를 가진 단어일수록
낮은 정수 인덱스를 부여

| 패딩

```
In [27]: print('리뷰의 최대 길이 :', max(len(l) for l in X_train))
print('리뷰의 평균 길이 :', sum(map(len, X_train))/len(X_train))
plt.hist([len(s) for s in X_train], bins=50)
plt.xlabel('length of samples')
plt.ylabel('number of samples')
plt.show()
```

리뷰의 최대 길이 : 47
리뷰의 평균 길이 : 9.8116625



```
In [28]: def below_threshold_len(max_len, nested_list):
count = 0
for sentence in nested_list:
    if len(sentence) <= max_len:
        count = count + 1
print('전체 샘플 중 길이가 %s 이하인 샘플의 비율: %s'%(max_len, (count / len(nested_list))*100))
```

```
In [29]: max_len = 80
below_threshold_len(max_len, X_train)
```

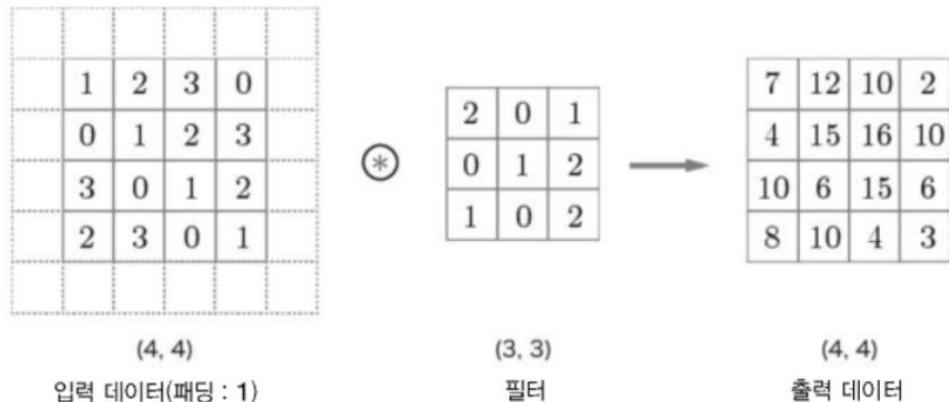
전체 샘플 중 길이가 80 이하인 샘플의 비율: 100.0

-리뷰의 최대 길이 및 평균 길이
확인 >>> 적정 패딩 확보

-80으로 패딩 시 보전 가능한
샘플 비율 100%

| 패딩

패딩 사용 이유



사용하는 데이터들의 문장의 길이가 다를 수 있다.

하지만 기계는 길이가 전부 동일한 문장들에

대해서만 하나의 행렬로 한꺼번에 묶어서 처리할 수 있다.

따라서, 여러 문장의 길이를 임의로 동일하게 맞춰주는 작업이 필요하다.

| 임베딩

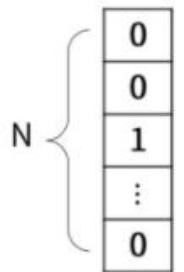
임베딩 사용 효과

	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00

사람이 쓰는 자연어를 기계가 이해할 수 있는 숫자형태로 바꿔주는 것
각 단어들로 구성된 문장을 다 같은 차원의 벡터로 만들어주고 각각의 단어에 수치를
부여한다, 그러면 위의 표에서 처럼 Gender, Royal 이라는 축을 둘 때 각각의
단어들이 유사한 단어들끼리 유사한 수치가 나오게 한다.

| 워드 임베딩

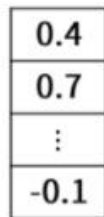
원-핫인코딩과 비교



희소 표현



Word Embedding



밀집 표현

희소 벡터의 문제점(원-핫 인코딩)

단어의 개수가 늘어나면 벡터의 차원이 한없이 커진다는 것



워드 임베딩

- 밀집 표현은 벡터의 차원을 단어 집합의 크기로 상정하지 않음
- 사용자가 설정한 값으로 모든 단어의 벡터 표현의 차원을 맞춤
- 더 이상 0과 1만 가진 값이 아니라 실수 값을 가짐

| GRU로 리뷰 감성 모델 학습 및 예측

```
in [51]: from tensorflow.keras.layers import Embedding, Dense, GRU
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.models import load_model
         from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

in [52]: embedding_dim = 100
         hidden_units = 128

         model = Sequential()
         model.add(Embedding(vocab_size, embedding_dim))
         model.add(GRU(hidden_units))
         model.add(Dense(1, activation='sigmoid'))

in [53]: es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=4)
         mc = ModelCheckpoint('best_model.h5', monitor='val_acc', mode='max', verbose=1, save_best_only=True)

in [54]: model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
         history = model.fit(x_train, y_train, epochs=15, callbacks=[es, mc], batch_size=64, validation_split=0.2)

Epoch 1/15 [-----] - ETA: 0s - loss: 0.2897 - acc: 0.8982
Epoch 00001: val_acc improved from -inf to 0.9034, saving model to best_model.h5
Epoch 2/15 [-----] - ETA: 0s - loss: 0.2897 - acc: 0.8982 - val_loss: 0.2541 - val_acc: 0.9035
Epoch 00002: val_acc improved from 0.9034 to 0.9079, saving model to best_model.h5
Epoch 3/15 [-----] - ETA: 0s - loss: 0.2334 - acc: 0.9130
Epoch 00003: val_acc improved from 0.9079 to 0.9079, saving model to best_model.h5
Epoch 4/15 [-----] - ETA: 0s - loss: 0.2334 - acc: 0.9130 - val_loss: 0.2442 - val_acc: 0.9079
Epoch 00004: val_acc improved from 0.9079 to 0.9102, saving model to best_model.h5
Epoch 5/15 [-----] - ETA: 0s - loss: 0.2143 - acc: 0.9215
Epoch 00005: val_acc improved from 0.9102 to 0.9102, saving model to best_model.h5
Epoch 6/15 [-----] - ETA: 0s - loss: 0.2143 - acc: 0.9215 - val_loss: 0.2399 - val_acc: 0.9102
Epoch 00006: val_acc improved from 0.9102 to 0.9108, saving model to best_model.h5
Epoch 7/15 [-----] - ETA: 0s - loss: 0.2011 - acc: 0.9286
Epoch 00007: val_acc improved from 0.9108 to 0.9108, saving model to best_model.h5
Epoch 8/15 [-----] - ETA: 0s - loss: 0.2011 - acc: 0.9287 - val_loss: 0.2392 - val_acc: 0.9109
Epoch 00008: val_acc did not improve from 0.9108
Epoch 9/15 [-----] - ETA: 0s - loss: 0.1907 - acc: 0.9308
Epoch 00009: val_acc did not improve from 0.9108
Epoch 10/15 [-----] - ETA: 0s - loss: 0.1907 - acc: 0.9308 - val_loss: 0.2421 - val_acc: 0.9102
Epoch 00010: val_acc did not improve from 0.9108
Epoch 11/15 [-----] - ETA: 0s - loss: 0.1800 - acc: 0.9358
Epoch 00011: val_acc did not improve from 0.9108
Epoch 12/15 [-----] - ETA: 0s - loss: 0.1800 - acc: 0.9358 - val_loss: 0.2448 - val_acc: 0.9079
Epoch 00012: val_acc did not improve from 0.9108
Epoch 13/15 [-----] - ETA: 0s - loss: 0.1699 - acc: 0.9392
Epoch 00013: val_acc did not improve from 0.9108
Epoch 14/15 [-----] - ETA: 0s - loss: 0.1699 - acc: 0.9392 - val_loss: 0.2501 - val_acc: 0.9058
Epoch 00014: val_acc did not improve from 0.9108
Epoch 15/15 [-----] - ETA: 0s - loss: 0.1591 - acc: 0.9435 - ETA: 1s -
Epoch 00015: val_acc did not improve from 0.9108
Epoch 00015: early stopping

in [55]: loaded_model = load_model('best_model.h5')
         print("원 테스트 정확도: %.4f" % (loaded_model.evaluate(x_test, y_test)[1]))

1250/1250 [-----] - 9s 7ms/step - loss: 0.2444 - acc: 0.9059

원 테스트 정확도: 0.9059
```

| 리뷰 예측

```
In [36]: def sentiment_predict(new_sentence):  
    new_sentence = re.sub(r'[^ㄱ-ㅎㅌ-ㅣ가-힣 ]', '', new_sentence)  
    new_sentence = okt.morphs(new_sentence) # 토큰화  
    new_sentence = [word for word in new_sentence if not word in stopwords] # 불용어 제거  
    encoded = tokenizer.texts_to_sequences([new_sentence]) # 정수 인코딩  
    pad_new = pad_sequences(encoded, maxlen = max_len) # 패딩  
  
    score = float(loader.predict(pad_new)) # 예측  
    return score
```

```
In [37]: sentiment_predict('이 상품 진짜 좋아요... 저는 강추합니다. 대박')
```

```
Out[37]: 0.954418957233429
```

```
In [38]: sentiment_predict('진짜 배송도 늦고 개짜증나네요. 뭐 이런 걸 상품이라고 만들?')
```

```
Out[38]: 0.007522314786911011
```

```
In [76]: sentiment_predict("애매하네")
```

```
Out[76]: 0.3676239848136902
```

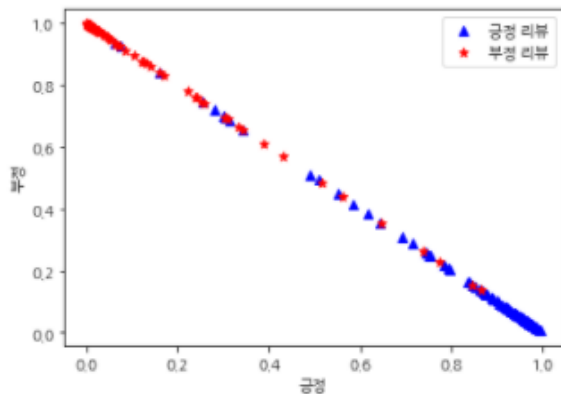
-학습된 데이터를 바탕으로 리뷰의 긍정 % 예측

| 리뷰 예측

```
In [52]: plt.scatter(V[:, 0], V[:, 1], c = 'blue', marker = '^', label = '긍정 리뷰')
plt.scatter(X[:, 0], X[:, 1], c = 'red', marker = '+', label = '부정 리뷰')

plt.xlabel("긍정")
plt.ylabel("부정")
plt.legend()
```

Out[52]: <matplotlib.legend.Legend at 0x1950c78dd30>



모델 학습에 사용한 데이터가 아닌
리뷰 데이터 긍정, 부정 각
700개씩 샘플링해 리뷰의 분포를
시각화하고 모델의 성능 확인

| 답글 생성

```
In [352]: def sentiment_bot():
    sentence = input()
    pos_score = sentiment_predict(sentence)[1]
    word = sentiment_predict(sentence)[0]

    if pos_score > 0.5:
        print('-----답글----- \n', pos_case_list[random.randrange(0,5)])

    elif pos_score < 0.5:
        if '배송' in word:
            print('-----답글----- \n', neg_case_list[0])
        elif '소리' in word:
            print('-----답글----- \n', neg_case_list[1])
        else:
            print('-----답글----- \n', neg_case_list[2])
```

```
In [348]: sentiment_bot()
```

이 상품 진짜 좋아요... 저는 감추합니다. 대박

-----답글-----

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님의 감사의 인사 드립니다
고객님께서 제품을 만족스럽게 이용하여 주시어 감사 드립니다😊.
앞으로도 아이펠리와 함께 해주시기를 바랍니다💖
언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다😊
좋은 하루 되십시오.

```
In [349]: sentiment_bot()
```

진짜 배송도 늦고 개짜증나네요. 뭐 이런 걸 상품이라고 만들?

-----답글-----

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님의 감사의 인사 드립니다
최근 택배사의 배송 지연 상황이 확인되어 실제 출고일로부터 배송은 약 3 - 5일 소요되고 있습니다😓.
고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 좋은 서비스 제공할 수 있도록 노력하겠습니다😊.
앞으로도 아이펠리와 함께 해주시기를 바랍니다💖
언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다😊
좋은 하루 되십시오.

```
In [351]: sentiment_bot()
```

아니 쓴지 이틀만에 구멍나서 버렸네요... 물론 종이질감필름이라 금방 헤진다고 생각은 하고있었지만 이건 너무 심하다고 생각해요ㅠㅠ 이틀이 뭐예요... 필기양이 많고 종이질감필름 쓰시는 문들은 절대 비추!!!!!! 진짜 그냥 바로 구멍나요ㅠㅠ

-----답글-----

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님의 감사의 인사 드립니다
팁 커버는 내부에 미세한 공기층이 있는 것이 정상 제품이며, 올바르게 착용되어 있지 않을 경우 불편함(필기감 하락, 빠른 파손, 인식불가오류 등)이 발생하실 수 있습니다.
팁 커버의 바른 착용을 위한 가이드 영상: <https://youtu.be/ed7p1H28Uk> 을 공유하여 드립니다.
공유하여 드린 영상을 참고하시어 제품을 다시 한번 착용하여 주신 후 제품을 사용하여 주십시오😊
팁 커버는 실리론으로 제작 된 상품인 관계로 사용자의 사용환경(필름/필압/사용량 등)에 따라 내구성 및 사용 기한은 차이가 발생할 수 있는 점 안내 드립니다.
만일 팁 커버를 끼우기 어려워 사용중 만족감이 낮으신 고객님들께는 팁 펜촉을 사용하여 주시기를 추천 드리고 있습니다😊
언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다😊
좋은 하루 되십시오.

- 기존의 대응 매뉴얼에 따라서 리뷰의 성격에 맞게 답글을 생성
- 부정 리뷰는 포함된 특정 단어(배송, 내구성)에 따라서 이에 맞는 케이스가 출력

3.

발전 방향

| 발전 방향

sentiment_bot()

이 상품 진짜 좋아요... 저는 강추합니다. 대박

-----답글-----

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님께 감사의 인사 드립니다
고객님께서 제품을 만족스럽게 이용하여 주시어 감사 드립니다😊.
앞으로도 아이펠리와 함께 해주시기를 바랍니다❤️
언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다😊
좋은 하루 되십시오.

사용자가 질문하면 미리 설계한 알고리즘의
시나리오 기반으로 답변하나, 이 또한 데이터의
부족 으로 인한 한계 발생

sentiment_bot()

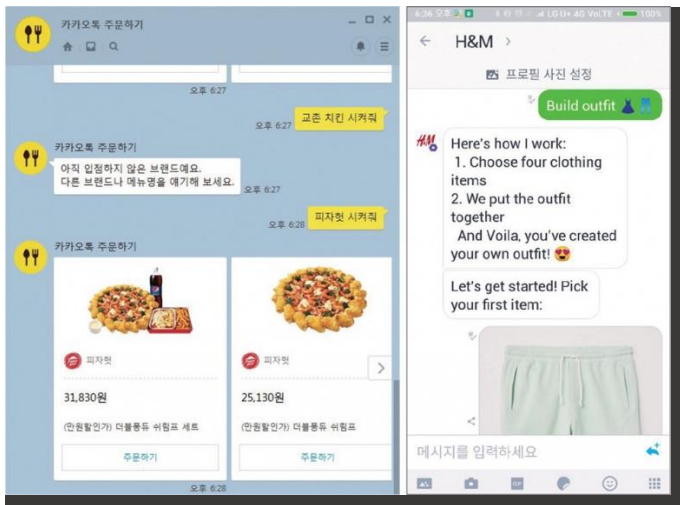
진짜 배송도 늦고 개짜증나네요. 뭐 이런 걸 상품이라고 만듦?

-----답글-----

안녕하십니까, 아이펠리 입니다.
아이펠리와 소중한 시간을 함께하여 주시는 고객님께 감사의 인사 드립니다
최근 택배사의 배송 지연 사항이 확인되어 실제 출고일로 부터 배송은 약 3 - 5일 소요되
고 있습니다😊.
고객님의 피드백은 내부적으로 전달하여 앞으로 더욱 좋은 서비스 제공할 수 있도록 노력
하겠습니다😊.
앞으로도 아이펠리와 함께 해주시기를 바랍니다❤️
언제나 고객과 함께하는 아이펠리 되겠습니다.
소중한 리뷰 감사 드립니다😊
좋은 하루 되십시오.

	현재	발전
제공방식	텍스트	텍스트,시각자료
필요기술	패턴매칭,키워드 및 연 관어 추출	딥러닝, 머신러닝, 자연어 처리
내용	-저장된 내용에 대한 질 의응답	-사용자의 개인 맞춤형 서비스 제공 -간단한 업무 처리

| 발전 방향



	발전
대화방식	쌍방향 정보 교류 : 대화 에이전트가 주도적으로 사용자와의 질문-답변 과정을 반복하여 부족한 정보를 보완
정보제공	맞춤형 정보제공 : 대화를 통해 사용자의 상황 정보를 인식하여 사용자에게 적합한 정보 제공
정보획득	대화를 통한 전문가의 도움 : 지식이 없는 사용자도 대화 에이전트의 가이드에 따라 원하는 정보에 빠르고 쉽게 접근

기대효과: 단순 지원업무 대체, 기존 인력의 타 업무 수행 및 업무효율성 향상
 주문 과정 간소화
 다른 제품 광고 통한 소비 유도