넥슨_ 딥러닝으로 욕설 탐지하기
- https://www.youtube.com/watch?v=K4nU7yXy7R8
- 게임마다 존재하는 욕설 탐지와 제재 시스템

왜 굳이 딥러닝으로 ?
- 기존: 금칙어 기반
    - 해서는 안 되는 단어가 포함될 경우 제재 및 마스킹
    - 문제1: 우회가 쉽다. (ㅅ1발)
        -> 금칙어를 늘리면 되지 않나? 문제2가 생김
    - 문제2: 오탐이 잦다
        - 18채널(게임속 쉬운 단어)>>**채널로 마스킹
        - 몇학년 몇반이야 >> 몇 학 *
        - 스페이스바 >>스페이**
            - 일반적인 대화도 합쳐짐
    - 근본적인 한계: 비속어와 공격적인 표현 구분이 어려움
        - 공격적이지 않는 자연스러운 욕설 제재
            : “제가 병신이었네요”
        - 비속어는 아니지만 공격적인 표현 제재하지 못함
            : “배를 확 따서 그냥 회를 쳐 먹을ㄲ”
    - 욕설 신고를 받고 운영자가 직접 읽고 판단
        >> 운영자의 시간이 소요되는 부담+ 욕설 읽음에 따른 부담

** 딥러닝으로 언어를 이해하고 욕설을 탐지하고자 프로젝트 진행
- 텍스트 분류의 큰 케잇으로 욕설 탐지를 진행.

 # 발표
- 부족한 데이터를 극복하려는 노력
- 텍스트 분류 모델 고도화
- 모델 해석
- 딥러닝 모델로 게임 운영에 도움을 주는 방법
- 역전파, 경사하강법은 시간상 발표 못함

 # 발표순서
- 프로젝트 목표, 프로토타입, 고도화, 서비스화

 # 프로젝트 목표:운영자의 수고를 덜게하기
- 현재는 모든 신고를 운영자가 수동으로 시별하는 방식
- 욕설 신고 중에서 욕인것과 아닌 것을 일일이 추리고 했음
- 지향점: 1차 자동 분류후에 운영자가 수동 식벼하는 식
    - 번역: 초벌번역 후 제대로 번역을 하는 것처럼

#프로토타입

-1. 데이터 : 라벨링 (안녕하세요 즐겜하세요: 오케이)(게임 w같이 하네:욕설) 태깅

- 크롤링+자체제작(노가다)

-2. 모델링 : 1DCNN

- CNN: 이미지 처리 분야에서 일반적으로 쓰이는 알고리즘. 지역적인 필터를 만들어서 특성ㄴ을 추출하고 이를 통해 예측. 예)고양이 분류를 한다고 하면 고양이 입, 귀, 수염 인식하게 하는 필터를 만들고 새로운 고양이 사진이 들어오면 특성을 기반으로 판단하는식. 이 필터는 데이터를 통해 학습
- 1DCNN: 자연어 처리에 CNN 도입하자
    - 입력:자모. 원래 1dcnn은 단어 기반 작동이나 자모를 입력으로 받은 것은 채팅 데이터가 띄어쓰기가 없고 오타 많아서.
- 임베딩: 텍스트를 숫자로 표현하는 과정 진행. 이 경우 비슷한 텍스트를 비슷한 숫자로 가지게 함. 이는 비슷한 욕에 대해서 비슷하게 욕으로 탐지할 수 있게, 즉 일반화를 하기 위해서. 이 임베딩도 데이터를 통해 학습
- 컨볼루션: 필터를 상뇽하여 특성을 추출. 필터 여러개를 사용하여 다양한 특성 추출.
    - 시발/지랄/시발+지랄 등이 필터가 되어 들어갈 수 있음.
- 풀링 레이어: 가장 특징적인것만 남겨두고 버림
    - 하는 이유: 1.불필요한 부분 버려서 노이즈 줄이기
    - 2. 텍스트의 길이와 상관없이 같은 개수의 특성을 뽑게 하기 위해서
- 출력 레이어:특성을 바탕으로 텍스트가 욕설일 확률을 계산

-3. 해석 : 어떤 부분 때문에 그렇게 해석을 했는가

- 필요이유1. **신뢰성 확보**

- 모델의 판단 근거가 믿을만한지를 알기 위해서

>> 예)의사가 주사를 맞으세요 보다는 이 증상은 이 병에서 기인했기에 ~점을 완화해주는게 좋습니다가 더 신뢰도가 있는것처럼

-필요이유2. 디버깅:모델이 왜 잘못된 판단을 했는지 알아보기 위해서 진행

- 필요이유3. 마스킹: 야 이 병신아=>> 야 이 **아. 이렇게 ** 처리하는게 어딘지 알려고

 # 어떻게 해석을 하나?

- 딥러닝 모델은 흔히 블랙박스라고 하죠. 값은 잘 나오는데 왜 그렇게 나오는지 알기힘들곤해요.
- 가장 무식하지만 가장 간단한 방법: 텍스트에서 글자 빼보기
    - 예) 게임 / 참/ 좇 같이/ 하네.
        - 하나씩 빼보면서 욕설여부를 판단

- 이 과정을 조금더 빠르고 효율적으로하는 알고리즘:LIME

- 텍스트가 아닌 이미지 예시 들긴하나 같음. 모델이 개구리라고 할 때 이곳 저곳 삭제해서 모델에 넣어보고 삭제한 부분과 모델을 비교하여 모델이 사진을 개구리라고 인식하는데 영향을 준 부분을 찾아냅니다
- 이 알고리즘은 오픈소스. 그래서 욕설 탐지기에 적용.

예)=> 게임 참 좇같이 하네>> 라임을 돌려서 왜 욕설로 보았나 보면

- 욕설일 확률 98프로
- 이게 왜 욕설이냐에 대해서 'ㅈ 같' 이 가장 크게 기여했따고 함

예)=> 뭐하고 있냐 개년아 >> 욕설로 89퍼센트로

<> 경제발전 5개년 계획은 욕설이 0.24퍼센트로 나타냄. 문맥 구분됨. (근데 오탐이 없진 않음 있긴 함)

- 시111발 미친놈인가 진짜

--> 정리: 1.입력 2.임베딩 3.컨볼루션 4. 풀링 5. 출력 레이어로 모델 생성

 # 핵심: 금칙어 사전없이 모델이 욕설을 학습 할 수 있음

- 정확도 비교: 기존 금칙어 사전과 비교하면 금칙어 기반의 정확도는 56퍼센트이나 1DCNN 같은 경우는 88퍼센트.

- 사전기반:보수적으로 탐지. 완전히 동일한 경우만 필터링

 # 프로토타입의 결과

- 데이터야 놀자에서 욕성 탐지기 데모를 보였고, (2017년 10월 13일)
- 욕설탐지기가 탐지 못하는 욕하는 게임을 진행했다고 하며 많은 분들이 참여해주셨다고 합니다.

 # 프로토타입 고도화

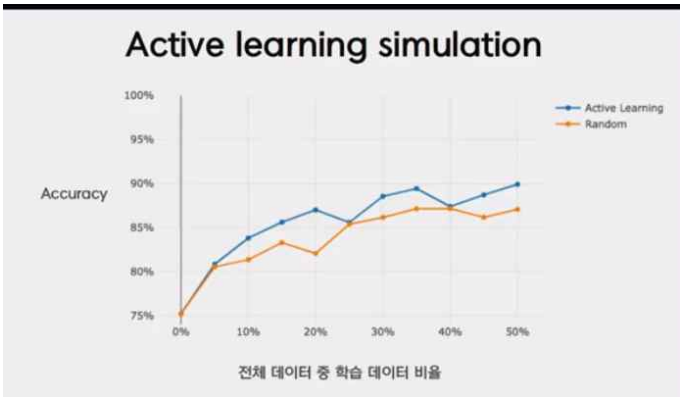
-1.데이터 : 더 효율적인 노가다

- 데이터는 빙산의 일각 같아요. 세상에는 unlabeled data가 수많아요. 그래서 labeled 데이터를 스마트하게 선택하는 접근방식이 필요하고 이것이 'active learning' . 스마트 하다는 것은 모델을 가장 잘 학습시킬 수 있게하는 것으로 적은 데이터로도 좋은 성능이 모델학습하게 함.
- 액티브 러닝: 처름 사람이 노가다로 데이터 만들어 학습시키고 그거로 다음 노가다 데이터를 선태하고. 이 과정을 반복되는 구조.

>> 핵심:다음 노가다할 데이터를 어뜧게 선택할것이냐이고, '모델이 가장 헷갈려하는 데이터를 뽑자'가 핵심.

>> 즉 50퍼센트로 탐지되는 욕설을 수작업으로 레이블링 하자고 하는겁니다.

- 시뮬레이션을 돌렸는데 데이터 다 쓰지 않고 50프로 및 10프로를 뽑을 때 액티브 러닝과 랜덤으로 뽑은 경우의 정확도 비교



-

-2. 모델링 : 더 깊고 강력한 모델링

- VDCNN : Very Deep CNN.
- 왜 깊어야할ㄲ요? 2012년 알렉스 넷이 8레이어>2014년 vggr 16레이어 >2015년에 ResNet이 152 레이어로 정확도 올림

>> 깊을수록 강력하다

>> 근데 깊을수록 문제가 생김.

- 모델이 얕으면 입력>레이어>레이어>예측을 진행하고, 이 오차를 최소화하려고 파라미터를 업데이트 하면서 진행.이 오차가 뒤에 있는 레이어ㄲ지 전달을 하는데 모델이 얕으면 학습이 쉬움. 그러나 모델이 깊다면 오차를 통해서 학습을 하려고 하는데 뒤에 있는 레이어ㄲ지 전달이 어려워서 학습이 어려워지는 문제가 생김.

>> 모델이 깊어셔도 해결되려고 나온: shortcut connection(skip connection) :레이어 2개를 레이어 건너 뛰어서 가는 것, 즉 지름길을 만들어서 모델을 깊게 쌓아요.

- VDCNN은 이 아이디어를 자연어처리에 가져온 것으로 사실 간단

>> [텍스트> 임베딩>컨볼루션> 컨볼류션 블록(컨볼류션 2겹 쌓은 것)> 풀링]>과정 반복> 출력 레이어를 쌓고 중간에 shortcut connection을 만들어줍니다

 # VDCNN: 90프로 정확성

- 데이터 셋이 클수록 힘을 받는데 넥슨 데이터 셋이 크지 않아서 원디씨앤과 큰 차이가 없음

-3. 해석: 더 빠르고 직관적인 해석

- 라임의 단점: 느리다.

>> 텍스트 하나 해석하려고 수십개의 변형을 만들어야 해석을 할 수 있음. 그래서 수천만개의 텍스트 해석에 있어서 계산적으로 어려움

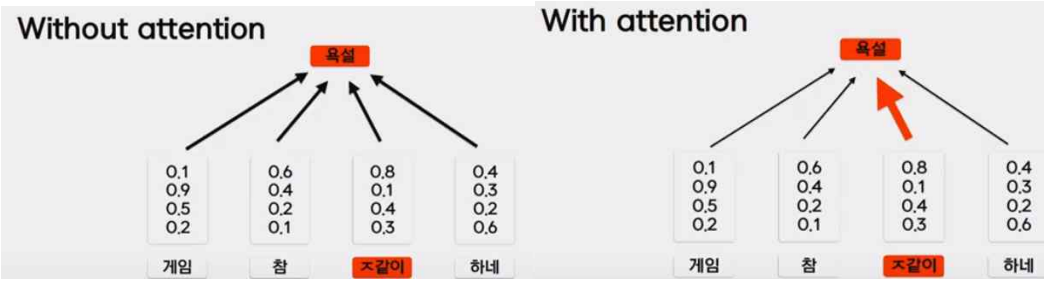
 # 모델 실행과 해석이 동시에 이뤄지게 할 수는 없을ㄲ?

- 어텐션.

>> 모델 어디에 집중할 것인가를 넣어주는 것. 즉 집중하는 곳이 결과에 영햐응ㄹ 많이 주는 곳이라 그곳이 결과의 근거다.

>>어텐션이 없으면 게임창에서 자/소단위로 했지만 단어를 숫자로 바꾸고 조합해서 판단을 했음.

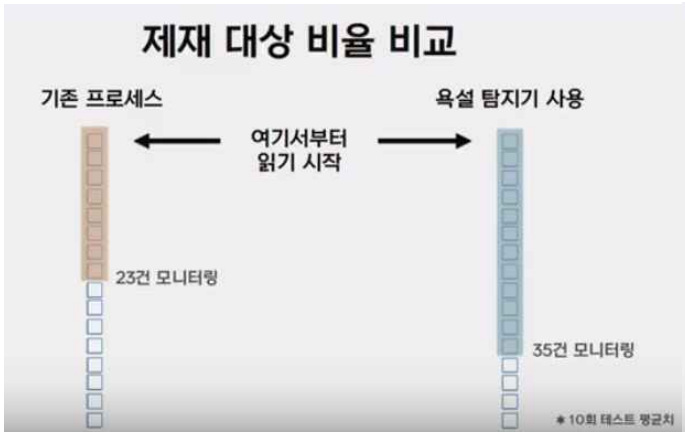
>> 어텐션이 있다면 가장 욕이라고 생각하는 부분의 데이터를 가져와서 욕설을 판단.



--->> 여러 문장을 동시에 탐지 및 해석할 수 있음. ***. 결과는 라임과 비슷하나 더 ᄲᆞ르다

 # 현재 넥슨 욕설탐지는 프로토타입으로 아직 실무에 반영된 것이 아니라고함

- 아래는 테스트 결과. 제재에 대한 부분



테스트 결과

테스트 회차	욕설탐지기 사용 결과		기존 프로세스 결과	
	1분 당 모니터링 건수	제재 대상 비율	1분 당 모니터링 건수	제재 대상 비율
1회차	34.3	97%	20.8	41%
2회차	35.0	96%	22	45%
3회차	32.3	97%	23.5	35%
4회차	31.2	95%	22.7	38%
5회차	36.2	96%	25.5	37%
6회차	38.4	96%	23	41%
7회차	35.7	95%	23.6	41%
8회차	38.9	95%	23.9	44%
9회차	36.3	94%	22.7	45%
10회차	36.2	97%	23.9	43%

<사이버래피티와 넥슨의 차이점>

(1) 수집한 데이터의 종류

- 우리는 실제 데이터를 긁어옴(채팅 및 자막)
- 그러나 넥슨에서 생성한 욕설은 프로젝트 참가자들이 뱉은 말을 기반으로 한다.

(2) 필터링 수준

- 우리는 비속어 및 욕설이 어린이 청소년에게 유해하면 모두 필터링
- 넥슨은 공격적이지 않는 자연스러운 욕설은 제재하는 한계점을 개선하기 위해 기존의 금칙어 기반에서 현재의 딥러닝 기반으로 시스템을 변경하게 되었다고 함. (예, 제가 병신이었네요)

(3) 정확도

- 넥슨은 기존에 금칙어 기반으로 필터링 및 마스킹을 하고 , 신고가 들어온 경우 운영자가 일일이 제재하는 식이었습니다.
    - 이때 금칙어기반으로는 accuracy가 56퍼센트, 1d cnn으로는 88%가 나왔씁니다
    <>사이버 래피티의 정확도는 90퍼센트(찬이말로는)라고 하였기에 넥슨의 것보다 좋음
- 그리고 금칙어의 경우에도 넥슨은 정확하에 일치하는 경우에만 보수적으로 탐지했으나, 우리 사이버래피티는 유사 욕에 대해서도 사전 기반(레이블링)이 탐지할 수 있음

(4) 확장성

- 넥슨은 넥슨의 게임 데이터에 대해서만 욕설 필터링이 적용 가능
- 사이버래피티는 트위치, 유튜브 아프리카 티비 등 범용적으로 사용할 수 있음