1. 인사 – 안녕하세요 울산대학교 챗봇조 발표를 맡은 이태관 학부생입니다

2. 목차 - 순서는 제가 사용한 사전 훈련된 모델의 간단한 설명 후 상세 요구사항들을 발표 하겠습니다

3. 먼저 pretrain model의 설명입니다

4. 기존의 많은 챗봇 연구는 BERT위주의 사전훈련 된 모델 위주로 진행되었습니다. BERT는 MLM 방식으로 학습되었습니다. MLM은 각각의 입력 단어 토큰들을 지정된 확률로 손상 시킨 후 원래 토큰으로 수정해가지면서 학습을 합니다. 여기 아래 그림을 보시면 단순함을 얻기란 복잡함을 얻기보다 어렵다라는 문장에서 얻기란과 어렵다가 마스킹 처리가 되었고 MLM 모델에서 마스킹된 부분이 무슨 단어인지 추측하는 것을 볼 수 있습니다. MLM은 pretrain 된거에 finetuning 을 하는 관점에서 효과적인 성능을 발휘하지만 전체토큰 중에서 통상의 손상 확률인 15%만 학습 되기에 로우 컴퓨팅에 취약한 면이 있습니다

5. RTD 방식은 제가 소개 드리고 싶은 훈련 방식입니다. RTD는 문장이 미리 학습해둔 어떤 AI 모모델 거쳤을 때 생성된 유사한 단어토큰에 대해서 각각 모든 토큰들이 다르게 변했는지 아닌지 이진분류 하는 방식입니다

6. 저희가 사용한 koElectra라는 pretrain 된 model 또한 RTD 방식으로 만들어진 모델입니다 위의 그림을 보시면 koelectra는 두개의 신경망 Generator와 discriminator로 구성되어 있습니다. Generator는 MLM방식으로 학습되었고 Discriminator는 RTD로 학습 되었습니다

7. genertator의 학습 메커니즘은 앞서 설명한 MLM과 유사하게 원래 토큰에서 마스킹을 주고 원래토큰을 예측 하면서 학습합니다

8. Discriminator의 학습 메커니즘은 Generator에서 마스킹할 위치의 토큰을 마스킹 하지말고 해당 위치의 단어토큰을 분포값에 근거해서 다른 단어로 바꾸고, 그 단어들이 다른걸로 변경이 됐는지 안됐는지 이진분류를 하면서 학습합니다

9. MLM과 RTD의 성능은 오른쪽 그림에서 보시는 것처럼 모든 상황에서 RTD가 좋은 것으로 나타났습니다

10. 다음은 요구사항 상세이고

11. 가장 먼저 사용자 요구사항부터 설명 드리겠습니다

12. 사용자는 울산대 관련한 각자 궁금한 사항들을 질문 할 수 있습니다. 또한 사용자 질문은 간결하고 명확할수록 좋습니다. 한시적으로 질문의 영역은 등록과 휴복학만 할 수 있습니다. 사용자는 질문을 하면 답변을 받을 수가 있는데 그 답변이 적절한 답변인지 아닌지 평가 할 수 있게 됩니다

13. 시스템의 진행은 총 5단계로 이루어집니다 사용자가 시스템에게 질문을하는 1단계, 시스템이 사용자에게 답변을 해주는 2단계, 답변이 정확했는지 사용자가 피드백을 주는 3단계, 원하는 답변이 나올때까지 반복하는 4단계, 원하는 답변을 얻었거나 적절한 답변을 줄수 없을 때 종료하는 5단계입니다

14. 저희가 개발한 챗봇의 우수성 및 차별성은 크게 3가지로 첫번째로 유지보수와 업데이트가 용이 합니다. 외주 챗봇은 새로 학습되는 것이 제한됩니다. 대부분 안되거나 매우 비싼 프리미엄 챗봇을 구매했을시 업데이트 횟수제한이 주어지는 정도가 다입니다. Q&A 데이터가 축적을 함으로서 더 좋은 성능을 뽑아 낼 수 있는게 자명한 현 상황에서 적절하지 않습니다.

15. 그리고 기반 기술이 비교적 최신이고 기존 챗봇의 기반기술과 비교 했을 때 비교적 더 높은 성능을 보유하고 있습니다

16. 또한 월 지속 비용이 계속 들고 초기의뢰비용은 견적마다 다 달라지는 업계의 현상황을 극복하여 비용을 절감 할 수 있습니다

17. 다음은 시스템 요구사항입니다

18. 시스템의 모듈은 크게 3가지로 이루어졌으며 DB와 AI모델, 그리고 UI가 있습니다

19. 챗봇 AI 모델은 핵심적인 챗봇의 구동 알고리즘을 기능하고 DB는 울산대학교 Q&A 광장에서 얻은 데이터를 재가공해서 만들었습니다. UI는 말그대로 유저 인터페이스 입니다

20. 챗봇 시스템의 모듈 결합과정은 첫번째로 AI 학습 알고리즘 설계 및 구현 후 두번째로 데이터를 크롤링 후 재가공을 거친 다음 첫번째의 알고리즘으로 두번째의 데이터를 사용해서 학습을 합니다. 마지막으로 학습이 완료된 모델과 웹 인터페이스를 결합해서 완성 시켰습니다

21. 첫번째로 챗봇 딥러닝 모델의 flow chart입니다 기본적으로 단일레이블 다중분류 형태이며 질문 text를 입력하고 입력 text로 레이블 분류 하고 적절한 답변을 선택 한 후 답변을 출력합니다. 답변 선택의 기준은 출력 softmax tensor에서 가장 높은 값의 인덱스를 선택하는 겁니다

22. flow chat 두번째에 해당하는 레이블 분류모델 설계도 입니다. 데이터를 재가공하여 pretrain된 koElectra model에 집어준 후

23. 그 위에다가 fine tuning 된 layer를 거쳐서 레이블을 예측합니다. Fine tuning은 4개의 Dense layer와 3개의 drop out을 추가하여 제작하였습니다

24. 모델 제작할 때 많은 문제들이 있었지만 특히 어렵게 다가웠던 문제들로는 크게 두가지가 있었습니다. Imbalanced Data 현상으로 학습 불균형의 문제와 데이터 수 부족 현상으로 overfitting이 되는 문제였습니다.

25. 대학생들의 평균 나이 특성상 창업휴학이나 출산휴학 관련 데이터들은 다른 데이터들에 비해서 현저히 수가 적었습니다. 이때 cross entropy를 사용했을 때 수가 적은 레이블은 훈련받는 시간이 너무 낮아서 모델의 정확도를 크게 낮추는 현상이 있었습니다

26. 저희는 이 문제를 pytorch 내장함수는 아니지만 논문을 참조해서 Focal loss를 구현해서 해결했습니다. Focal loss는 그림에서 보시는 것처럼 Cross Entropy와 구조는 거의 비슷하지만 각각의 레이블에게 가중치를 둬서 가중치가 높으면 Loss를 훅 떨어뜨립니다.

27. 구현한 Focal loss 함수의 테스트를 위해서 에브리타임이라는 울산대 온라인 커뮤니티에 1주일동안 설문을 실시하여 테스트 데이터를 확보하였고

28. Cross Entropy Loss와 Focal Loss간에 소수 데이터 정확도가 유의미한 차이가 있음을 확인 했습니다

29. 다음으로 데이터가 부족했는데 Data augmentation 기법을 사용해서 해결했고 사용자 질문 데이터 수집 기능을 추가 함으로서 성장가능성을 향상시켰습니다. Data augmentation은 이미 컴퓨터 비전에선 이미지를 돌려도 보고 노이즈를 줘보기도 해서 했지만 자연어처리에선 단어 하나만 바뀌어도 문장의 의미가 달라지기 때문에 이 메커니즘을 그대로 받아들이기엔 무리가 있었습니다.

30. 저희는 이문제를 번역기를 사용함으로서 해결했습니다. 아이디어는 번역을 2번 하면 문장이 비슷한 의미긴 한데 좀 이상하게 바뀌니까 어떻게 되지 않을까?에서 시작했습니다. 번역을 2번 하고 sbert의 문장 유사도 평가 인공지능 모델을 사용해서 50%가 넘으면 데이터를 추가 하는 과정을 거쳤고

31. (--대충 설명--) 추가 시켰습니다

32. 사용자 질문 데이터 수집 기능은 그림과 같은 과정을 거칩니다. 시스템이 답변되면 답변이 마음에 드냐고 물어 보고 yes라고 한다면 사용자의 질의와 분류된 레이블을 데이터에 저장을 하고 no가 3번째 누적된다면 답변을 찾아줄수 없다는 안내와 함께 사용자 질의 데이터만 저장을 합니다

33. yes라고 피드백을 준 데이터가 어느정도 쌓였을 경우 새로 저장된 데이터를 기반으로 새로 훈련 시킵니다

34. 사진과 함께 보시겠습니다. 시스템이 답변하고 마음에 드냐고 물어 봅니다. 사용자는 네와 아니오 버튼을 선택 할 수 있고

35. 만약 3번의 추가 질의에도 원하는 답변이 안나왔을 때 그림과 같은 안내를 해준 후 이 사이트를 소개 해주며 전화 문의를 유도하고

36. 3번이 되기전 사용자가 답변이 마음에 안든다 했을 때 시스템은 softmax tensor에서 다음으로 높은 확률의 레이블을 출력합니다.

37. 이제 시스템의 다음 모듈인 데이터 베이스 입니다 data는 울산대학교 UWINS Q&A광장에서 직접 크롤링을 해서 얻었습니다.

38. 전처리가 필요한 데이터들의 대표적인 특징들은 크게 3가지가 있었습니다

39. 한 질문에 여러 개의 질문과 여러 개의 답변이 모호하게 분리 되어있을 경우 각각 따로 직접 분류하였고

40. 한 질문에 여러 개의 질문과 여러 개의 답변이 명확하게 분리되어있을때도 각각 따로 직접 분류 하였습니다

41. 한 질문에 여러 개의 질문이 있지만 답변이 일부분만 되었을 경우 질문을 적절하게 삭제했습니다

42. 다음으로 시스템의 다음 모듈인 UI 입니다. 화면을 구성하는 기능은 화면 하단에 사용자가 질문을 입력하는 칸 그리고 오른쪽에 사용자가 무슨 질문을 했는지 볼 수 있는 영역 그리고 시스템의 답변을 볼 수 있는 영역이 있습니다

예상 질문

1. 왜 등록과 휴복학만 제작했냐?

* 울산대학교 정보통신원에게 자문을했습니다. 등록금관련과 휴학 어떻게 하냐고 복학 어떻게 하냐는 질문이 시시때때로 많기 때문에 챗봇을 만든다면 이쪽이 가장 우선순위가 되면 좋겠다라는 답변을 받았습니다

1. 번역기 쓰는거 근거가 있는거냐?

* 사실 처음엔 관련 논문을 참조하지 않고 현재 개발자로 근무하고 계시는 분에게 자문을 해서 아이디어를 냈습니다. 그러나 후에 서칭하다가 이게 코스트가 많이 들긴 하지만 EDA라는 Data augmentation기법이 자리 잡기전 고전적으로 해왔던 방식이라는걸 뒤늦게 알았습니다

1. Focal loss에 대해서 자세한 설명

* Focal loss는 원래 컴퓨터 비전에서 쓰던 loss 법입니다. Object detection에서 크게 one stage detector와 two stage detector가 있습니다 둘의 차이점은 이미지의 localization과 classification을 동시에 하면 one stage detector, 따로따로 하면 two stage detector입니다.
* One stage detector에선 모든 이미지를 동일한 오브젝트 후보에 놓고 학습을 합니다. 속도면에서 에서 좋지만 정확성에서 떨어집니다
* 예를들어 하늘색 배경에 사람들이 있는 사진으로 사람을 훈련 시켰을 때 배경은 너무 넓고 사람은 너무 좁아서 사람 영역에 대한 학습이 배경이 학습 하는 것에 비해서 너무 짧게 학습 되는 문제가 있었는데 이걸 focal loss로 해결 할 수 있습니다
* Focal loss ppt로 돌아가보시면 이 그림과 식은 focal loss를 나타냅니다 이 식을 보면 이 둘의 차이는 이 부분밖에 없는 것을 알 수 있습니다. 여기서 r이 0이라면 둘은 똑 같은 식이 됩니다
* 이 부분의 역할은 쉽게 학습이 되는 label의 loss 가중치를 줄이기 위함입니다
* Focal loss의 시나리오는 크게 2가지가 있습니다
* 만약 잘못 분류되면 pt값이 작아지게 됩니다. 그럼 앞에 부분도 1에 가까워지고 log(pt)도 커져서 loss에 반영이 됩니다
* Pt가 1에 가까워지면 앞에부분은 0에 가까워지고 cross entropy가 비슷하게 log(pt)또한 줄어듭니다