발표 대본

슬라이드 1: 표지

발표자: 안녕하세요. 저희는 팀 '깡마이'입니다. 오늘 'CLIP 모델을 활용한 VQA 태스크 분석 및 성능 개선 방안'에 대해 발표하게 되어 기쁩니다. 저는 AI Research를 담당한 [본인 이름]입니다.

슬라이드 2: 과제 정의 및 접근 전략

발표자: 먼저, 저희가 해결하고자 했던 과제에 대해 말씀드리겠습니다. 저희는 주어진 이미지와 질문을 이해하고, 네 개의 선택지 중에서 가장 적절한 답변을 선택하는 VQA, 즉 시각적 질의응답 문제를 다루었습니다. 핵심 목표는 '일상 사진' 도메인에 대한 모델의 이해도를 극대화하여 높은 정확도를 달성하는 것이었습니다. 주요 도전 과제는 이미지의 시각적 정보와 질문 및 선택지의 텍스트 정보를 효과적으로 결합하여 의미적 관계를 정확히 추론하는 것이었습니다.

이러한 과제를 해결하기 위해 저희 팀은 CLIP 모델의 제로샷 이미지-텍스트 매칭 능력을 기반으로 단계적인 접근법을 채택했습니다. 첫째, CLIP 모델을 기반 모델로 선정하고, 둘째, 프롬프트 엔지니어링을 통해 모델의 이해를 돕고, 셋째, 모델 미세 조정을 통해 VQA 태스크에 최적화하며, 마지막으로 견고한 추론 전략을 수립했습니다.

슬라이드 3: CLIP 모델의 구조와 특징

발표자: 저희 솔루션의 핵심인 CLIP 모델에 대해 자세히 살펴보겠습니다. CLIP은 이미지 인코더와 텍스트 인코더로 구성되어 이미지와 텍스트를 동일한 통합적 표현 공간에 매핑합니다. 이를 통해 이미지와 텍스트 간의 유사도를 직접 측정할 수 있습니다.

CLIP의 가장 큰 특징 중 하나는 제로샷 능력입니다. 이는 학습 시 보지 못한 이미지-텍스트 쌍에 대해서도 높은 예측 성능을 보이는 것을 의미합니다. 또한, 비대칭적 유사도 측정 방식을 통해 VQA와 같이 이미지와 텍스트 간의 복합적인 관계를 추론하는 태스크에 특화된 방식으로 활용될 수 있습니다.

슬라이드 4: 데이터 처리 및 증강 기법

발표자: 데이터 처리 및 증강 기법에 대해 설명드리겠습니다. 저희는 PyTorch의 `Dataset` 클래스를 상속받아 `VQADataset`을 구현했습니다. 이 클래스는 이미지 로딩, 질문 및 선택지 추출, 그리고 이미지 증강을 담당합니다. 특히, `CLIPProcessor`가 내부적으로 이미지 전처리를 처리하므로, 기존에 사용되던 `ToTensor`와 `Normalize` 변환은 제거하여 효율성을 높였습니다.

슬라이드 5: 핵심 아이디어: 프롬프트 엔지니어링

발표자: 저희 솔루션의 핵심 아이디어 중 하나는 프롬프트 엔지니어링입니다. 일반적인 VQA 프롬프트는 단순히 질문을 던지는 형태이지만, CLIP은 이미지와 텍스트 쌍의 유사도를 측정하는 데 특화되어 있습니다. 이 점을 활용하여 저희는 프롬프트를 최적화했습니다.

기존 방식인 "질문: {question}, 선택지: {choice}" 대신, "A daily life photo: {q}. The correct answer is: {c}"와 같이 프롬프트를 재구성했습니다. 이 방식은 두 가지 기대 효과를 가져옵니다. 첫째, "일상 사진"이라는 문구를 추가하여 대회 데이터의 도메인 특성을 모델에 명시적으로 알려줍니다. 둘째, 질문-답변 형식을 '가장 적합한 설명 찾기' 문제로 재구성하여, CLIP이 이미지와 가장 유사한 (질문+정답) 텍스트를 효과적으로 찾도록 유도합니다.

슬라이드 6: 모델 미세 조정 (Fine-tuning) 전략

발표자: 다음으로 모델 미세 조정 전략입니다. 저희는 CLIP 모델의 출력인 `logits_per_image`가 (배치 크기, 배치 크기 * 선택지 수) 형태의 유사도 행렬임을 파악했습니다. 여기서 핵심 로직은 각 이미지에 해당하는 4개의 선택지 로짓만을 정확히 추출하여 (배치 크기, 4) 형태로 재구성하는 것입니다. 이렇게 재구성된 로짓에 손실 함수를 적용하여 모델을 학습시켰습니다.

학습 프로세스에서는 `AdamW` 옵티마이저와 `CrossEntropyLoss`를 사용했으며, 혼합 정밀도 학습을 통해 GPU 메모리를 효율적으로 사용하고 학습 속도를 높였습니다. 이 과정을 통해 모델은 VQA 태스크에 더욱 특화된 성능을 발휘하게 됩니다.

슬라이드 7: 추론 전략 및 성능 극대화

발표자: 단일 예측의 불안정성을 최소화하고 최종 제출 점수를 극대화하기 위해 다각적인 추론 전략을 적용했습니다. 첫째, 테스트 시점 증강, 즉 TTA를 적용했습니다. 테스트 데이터에

`RandomRotation`과 `RandomHorizontalFlip`과 같은 서로 다른 두 가지 증강을 적용하여 총 2회 예측을 수행했습니다.

둘째, 앙상블 기법을 사용했습니다. TTA를 통해 얻은 각 예측 확률을 산술 평균하여 최종 확률을 계산했습니다. 이 과정을 통해 단일 예측의 오류나 편향을 완화하고 더 안정적인 결과를 도출할 수 있었습니다.

셋째, 온도 스케일링을 적용했습니다. 로짓을 소프트맥스 함수에 통과시키기 전에 `temperature` 값인 0.8로 나누어 주었습니다. 1보다 작은 온도는 모델이 예측한 확률 분포를 더 뾰족하게 만들어, 가장 높게 예측한 선택지에 대한 신뢰도를 증폭시키는 효과를 줍니다.

슬라이드 8: 코드 구조 및 주요 함수 분석

발표자: 저희 프로젝트의 코드 구조와 주요 함수에 대해 설명드리겠습니다. 저희는 `train.py`와 `inference.py`로 코드를 분리하여 모델 학습과 추론을 명확히 구분했습니다. `train.py`에서는 모델 학습 및 가중치 저장을 담당하고, `inference.py`에서는 저장된 가중치를 로드하여 예측을 수행합니다.

주요 함수로는 `fine_tune()` 함수가 데이터 로드 및 전처리, 학습 로직을 포함하며, `save_weights()` 함수는 모델 가중치를 저장하고, `predict()` 함수는 가중치 로드 및 예측 로직을 수행합니다. 이러한 코드 분리를 통해 체계적인 실행 계획을 수립하고, Private Score 복원에 대한 높은 신뢰도를 제공할 수 있었습니다.

슬라이드 9: 모델 성능 및 예측 결과 분석

발표자: 저희 모델의 성능 및 예측 결과 분석입니다. 저희는 시드 고정(`torch.manual_seed(42)`)을 통해 결과의 재현성을 보장했습니다. 또한, `FileNotFoundError`와 같은 예외 처리를 통해 코드의 안정성을 확보했습니다.

성능 향상에는 테스트 시점 증강(TTA) 효과와 앙상블 기법이 크게 기여했습니다. 혼합 정밀도 학습(Automatic Mixed Precision)을 통해 GPU 메모리를 최적화하여 효율적인 학습을 가능하게 했습니다. 제로샷 학습과 온도 스케일링 또한 핵심적인 신뢰성 요소로 작용했습니다. 이러한 기술적 접근과 도메인 특화 전략을 통해 높은 성능을 달성할 수 있었습니다.

슬라이드 10: 결론 및 핵심 강점 요약

발표자: 마지막으로 결론 및 핵심 강점 요약입니다. 본 프로젝트는 CLIP 모델을 VQA 태스크에 성공적으로 적용하기 위한 심도 있는 분석과 체계적인 실험의 결과물입니다. 저희 솔루션의 핵심적인 차별점은 대회 데이터의 특성을 반영한 정교한 프롬프트 설계와 TTA 기반의 앙상블 추론 전략입니다.

저희는 높은 재현성 및 안정성, 그리고 효율적인 학습을 통해 신뢰할 수 있는 결과를 제공했습니다. 이 모든 강점들이 결합되어 VQA 태스크에서 뛰어난 성능을 보여줄 수 있었습니다.

이상으로 발표를 마치겠습니다. 감사합니다. 질문 있으신가요?