대화 요약 경진대회 전략 로드맵 및 베이스라인 코드 생성 프롬프트 제안

1. 개요

본 보고서는 한국어 대화 요약 경진대회에서 경쟁 우위를 확보하기 위한 포괄적인 전략 로드맵을 제시하고, 효율적인 베이스라인 코드 개발을 위한 Gemini-CLI 프롬프트를 제안한다. 경진대회는 한국어로 번역된 DialogSum 데이터셋을 활용하며, 이는 대화의 상호작용성, 비격식적 표현, 그리고 복잡한 정보 흐름으로 인해 일반적인 텍스트 요약보다 높은 난이도를 가진다.[1]

성공적인 경진대회 참여를 위해서는 대화의 역동성에 대한 깊은 이해, 세심한 데이터 전처리, 인코더-디코더 또는 디코더-온리 아키텍처 중 적절한 모델 선택, 그리고 부정확한 지시 정보(coreference), 핵심 정보 누락, 사실 불일치 등 일반적인 요약 오류를 사전에 방지하는 데 중점을 둔 사전 대응이 필수적이다. 본 보고서는 이러한 핵심 전략적 기둥을 중심으로 다단계 로드맵을 제시하며, 궁극적으로 강력한 베이스라인을 신속하게 구축할 수 있도록 상세하고 아키텍처를 고려한 Gemini-CLI 프롬프트를 제공하여 즉각적인 실용적 적용과 견고한 시작점을 보장한다.

2. 경진대회 개요 및 데이터 분석

2.1. 경진대회 목표 및 태스크 정의

본 경진대회는 텍스트 요약, 특히 대화 요약에 초점을 맞추고 있으며, 원본 DialogSum 데이터셋을 한국어로 번역한 자료를 활용한다.[1] 대화 요약은 본질적으로 다중 화자의 상호작용적 특성, 비공식적인 언어 사용, 그리고 복잡한 정보 흐름으로 인해 일반적인

텍스트 요약보다 복잡하다. 한국어로 번역된 데이터셋을 사용하는 것은 추가적인 언어적, 문화적 고려 사항을 수반한다.

데이터셋이 "한국어로 번역된" 버전이라는 점은 중요한 의미를 지닌다.[1] 번역 과정에서 구어체 표현, 관용구, 담화 표지어 등의 미묘한 뉘앙스가 완벽하게 전달되지 않아 원래의 의도나 정보 흐름이 변경될 수 있다. 특히 기계 번역이 사용되었다면, 사람에 의해 생성된 한국어 대화에 비해 오류, 문법적 어색함, 또는 부자연스러운 흐름이 발생할 수 있다. 이러한 번역 과정의 특성은 사전 학습된 한국어 자연어 처리 모델이 이 번역된 데이터셋에 특화된 불일치나 노이즈에 직면할 수 있음을 시사한다. 따라서 탐색적 데이터 분석(EDA) 과정에서 번역된 예시들을 철저히 질적으로 검토하고, 번역으로 인해 발생할 수 있는 언어적 "노이즈"를 처리하기 위한 특정 데이터 정제 단계가 필요할 수 있다. 이는 모델의 미세 조정 전략이 번역 아티팩트로부터 발생할 수 있는 언어적 노이즈를 처리하기 위해 더욱 견고해야 함을 의미한다.

2.2. DialogSum 데이터셋 특성

DialogSum 데이터셋의 구조와 규칙을 깊이 이해하는 것은 정확한 데이터 전처리, 효과적인 모델 입력 표현, 그리고 생성된 요약이 경진대회 지침을 준수하도록 보장하는 데 필수적이다. 각 대화에 대해 세 가지 요약문이 제공되는 것은 모델 학습 및 평가에 더 풍부한 목표를 제공한다.

- 학습 데이터**:** 총 12,457개의 대화-요약 쌍으로 구성되어 있으며, 한국어로 번역되었다.[1]
- 테스트 데이터**:** 총 499개의 대화로 구성되며, 이 중 250개는 공개 평가에, 249개는 최종 비공개 평가에 사용된다. 두 분할 간의 대화 주제 분포는 유사하다.[1]
- 요약문 개수**:** 각 대화에는 3개의 고유한 정답 요약문이 존재하며, 이는 주제, 내용, 개체명 등 다양한 관점을 반영한다.[1] 각 대화에 대해 세 가지 고유한 정답 요약문이 제공되는 것은 의도적인 설계 선택이다. 이는 인간의 요약이 주관적이며, 주요 주제, 특정 내용, 핵심 개체 등 다양한 측면을 우선시할 수 있음을 인정하는 것이다. 만약 하나의 참조 요약만 존재했다면, 모델은 의미적으로는 정확하지만 스타일이 다른 요약을 생성했을 때 불이익을 받을 수 있다. 여러 참조 요약이 존재함으로써 평가는 더욱 견고해지며, 더 넓은 범위의 허용 가능한 출력을 허용한다. 모델 학습의 관점에서는, 이러한 다양성을 모델이 학습하도록 노출시키는 것이 중요하다. 여러 참조 요약으로 학습하는 것은 모델이 더 넓은 범위의 핵심 정보를 포착하고, 적어도 하나의 인간적 관점과 일치할 가능성이 높은 더 다재다능한 요약을 생성하도록 장려할 수 있다. 평가 측면에서는, ROUGE 점수가 이러한 다양한 참조 요약에 대해 평균화되므로, 요약 품질을 더욱 포괄적으로 나타내는 지표가 된다.

- 대화 규칙**:**
  - 화자는 #Person [Number]# 형식으로 식별된다.
  - 각 발화문은 콜론과 공백(:) 뒤에 등장한다.
  - 화자 간 구분은 개행 문자(\n)로 이루어진다.[1]
  - 예시: #Person1#: 안녕하세요, 스미스씨, 저는 호킨스 의사입니다. 오늘 왜 모셨나요?\n#Person2#: 건강검진을 받는 것이 좋을 것 같아서요..[1]
- 요약 규칙**:**
  - 한 문장 또는 여러 문장으로 구성될 수 있다.
  - 화자 또는 주요 개체명을 포함해야 한다.
  - 대화 길이의 20% 이내로 간결하게 요약되어야 한다.
  - 관찰자 관점에서 작성되며, 화자의 의도를 이해하고 작성해야 한다.
  - 은어나 약어 없이 공식적으로 사용되는 언어로 작성되어야 한다.[1]
- 개인 정보 마스킹**:** 총 8가지 특정 개인 식별 정보(PII) 범주가 고유한 특수 토큰(예: #PhoneNumber#, #Address#)으로 마스킹되어 있다. 이러한 토큰은 모델에 의해 분리할 수 없는 단위로 처리되어야 한다.[1] 개인 정보 마스킹은 단순히 개인 정보 보호 조치가 아니다. 요약 규칙 3번은 "대화 내에 중요한 명명된 개체를 보존"해야 한다고 명시하고 있으며, 이는 마스킹된 PII 토큰이 요약문의 핵심 콘텐츠 요구 사항과 직접적으로 연결됨을 의미한다. 이는 이러한 토큰이 일반적인 알 수 없는 토큰으로 취급되어서는 안 된다는 것을 시사한다. 대신, 이들은 토크나이저의 어휘에 고유하고 분리할 수 없는 단위로 추가되어야 한다. 더욱이, 이들의 존재는 모델에게 이들이 요약문에 유지되어야 할 매우 중요한 개체임을 알려주는 신호로 작용해야 한다. 이는 예를 들어, 어텐션 메커니즘에서 더 높은 중요도를 부여하거나, 소스 대화에서 요약문으로 이러한 특정 토큰을 복사하는 것을 우선시하는 복사 메커니즘을 통합함으로써 달성될 수 있으며, 이는 사실적 일관성과 요약 규칙 준수를 향상시킨다.

다음 표는 DialogSum 데이터셋의 주요 특성을 요약한 것이다.

특성	세부 내용
데이터셋 크기	학습: 12,457개 대화-요약 쌍

2.3. 평가 지표

주요 평가 지표는 ROUGE-1-F1, ROUGE-2-F1, ROUGE-L-F1이며, 이들은 토큰화 후에 계산된다.[1] ROUGE는 생성된 요약과 참조 요약 간의 n-그램(1-그램, 2-그램) 및 최장 공통

부분 수열(L)의 중복도를 측정한다. F1 점수는 정확도(생성된 내용이 정확한가)와 재현율(생성되어야 할 내용이 생성되었는가)의 균형을 맞춘다. "토큰화 후"라는 조건은 매우 중요하다.

ROUGE 점수는 토큰화 방식에 매우 민감하다.[1] 만약 경진대회 평가 시스템이 특정 한국어 토크나이저(예: Mecab과 같은 특정 형태소 분석기, 또는 특정 코퍼스에서 학습된 SentencePiece와 같은 서브워드 토크나이저)를 사용하는데, 모델의 전처리 과정에서 다른 토크나이저를 사용한다면, 토큰 집합이 완벽하게 일치하지 않을 수 있다. 예를 들어, "건강검진을"이라는 단어가 한 토크나이저에서는 "건강", "검진", "을"로 토큰화될 수 있고, 다른 토크나이저에서는 "건강검진", "을"로 토큰화될 수 있다. 이러한 불일치는 의미적으로는 사소하더라도 n-그램 중복도를 감소시켜 낮은 ROUGE 점수로 이어질 수 있다. 따라서 경진대회 평가 시스템이 사용하는 정확한 토큰화 방법을 식별하거나 추론하는 것이 무엇보다 중요하다. 만약 이 정보가 명시적으로 제공되지 않는다면, 팀은 일반적인 한국어 토크나이저들을 실험하고 데이터셋에 대한 출력을 분석하여 평가 방식과 가장 잘 일치할 가능성이 있는 것을 선택해야 한다. 이는 전처리 파이프라인과 Gemini-CLI 프롬프트의 "입력 토큰화" 지침에 직접적인 영향을 미치며, 학습, 생성, 평가 전반에 걸쳐 일관된 토크나이저를 사용하는 것이 보고되는 성능을 극대화하는 데 필수적이다.

2.4. 탐색적 데이터 분석**(EDA)** 결과

EDA는 데이터셋의 본질적인 특성에 대한 기본적인 통찰력을 제공하며, 이는 효과적인 전처리 전략, 모델 아키텍처 선택, 그리고 학습 방법론을 안내하는 데 매우 중요하다.

- 텍스트 길이 분포**:** 대화와 요약문 길이 모두 '오른쪽으로 치우친(right-skewed)' 분포를 보이며, 이는 짧은 텍스트가 많지만 상당히 긴 텍스트도 존재함을 나타낸다. 요약문은 일관되게 대화 길이의 약 20% 수준이다.[1] 오른쪽으로 치우친 길이 분포는 많은 대화가 짧지만, 상당히 긴 대화도 존재하여 모델이 확장된 컨텍스트를 처리하는 능력에 도전 과제를 제시함을 의미한다. 요약문 길이의 20% 제약은 모델이 상당한 추상화와 압축을 수행해야 함을 의미한다.[1] 이는 긴 입력 시퀀스를 효율적으로 처리할 수 있는 모델(예: 잘 확장되는 어텐션 메커니즘 또는 계층적 인코딩 기술)이 필요하다는 것을 시사한다. 또한 전처리 과정에서 효과적인 자르기(truncation) 및 패딩(padding) 전략의 중요성을 강조한다. 매우 긴 대화의 경우, 모델은 "요약 후 요약" 또는 대화 분할과 같은 전략을 사용해야 할 수도 있으며, 이는 복잡성을 추가하지만 분포의 긴 꼬리 부분에서 성능을 유지하는 데 중요할 수 있다.
- 대화 카테고리**:** 데이터셋에는 일상 대화, 비즈니스 상호작용, 인터뷰 등 다양한 대화 카테고리가 포함되어 있다. 상위 30개 주제를 시각화한 결과, 일상 대화(17.5%)와

비즈니스 대화(13.9%)가 두드러진다.[1] 다양한 대화 카테고리(일상, 비즈니스, 인터뷰)의 존재는 언어 스타일, 핵심 개체, 그리고 요약에 중요하다고 간주되는 정보의
유형이 카테고리별로 상당히 다를 수 있음을 나타낸다.[1] 예를 들어, 비즈니스 대화 요약은 결정과 실행 항목을 우선시할 수 있는 반면, 인터뷰 요약은 자격과 결과에 초점을 맞출 수 있다. 문서 자체에서도 "카테고리별로 모델을 구분하거나 이에 맞는 처리를 하는 것이 필요하다"고 명시적으로 언급하며 이러한 관찰을 뒷받침한다.[1] 이는 단일의 일반적인 요약 모델이 모든 카테고리의 뉘앙스를 최적으로 포착하는 데 어려움을 겪을 수 있음을 시사한다. 따라서 도메인 적응 또는 카테고리 인식 모델링 전략을 탐색해야 한다. 잠재적인 접근 방식으로는 1) 모델을 안내하기 위한 입력 특징으로 카테고리 임베딩을 통합하는 것; 2) 모델이 대화 카테고리를 동시에 예측하는 다중 작업 학습; 또는 3) 데이터 볼륨이 허용하는 경우, 고빈도 카테고리에 특화된 모델을 훈련하거나 기본 모델을 미세 조정하는 것이 있다.

- 비격식적 언어**:** 약어(예: ETA, BWT) 및 이모티콘(예: :), ^^)을 최소화하려는 노력에도 불구하고, 채팅과 같은 데이터에는 여전히 이러한 비격식적 요소가 포함될 수 있다.[1] 문서에서는 약어와 이모티콘이 데이터셋에 여전히 존재할 수 있음을 인정한다.[1] 이러한 비격식적 요소는 표준 토크나이저와 주로 공식 텍스트로 학습된 모델에 어려움을 줄 수 있다. 더욱이, 요약 규칙은 요약문이 은어나 약어 없이 "공식적으로 사용되는 언어"로 작성되어야 한다고 명시하고 있다.[1] 이는 이러한 비격식적 요소를 정규화하거나 처리하기 위한 전용 전처리 단계가 필요하다는 것을 의미한다. 전략에는 일반적인 약어를 전체 형태로 대체하는 규칙 기반 대체(예: "BTW"를 "By the way"로) 또는 감정이 요약에 중요하지 않다면 이모티콘을 일반적인 `` 토큰으로 제거/대체하는 것이 포함될 수 있다. 이는 모델의 입력이 더 깨끗하고 출력이 공식 언어 요구 사항을 준수하여 비격식적인 텍스트를 생성하는 것을 방지한다.

3. 팀 협업을 위한 전략적 로드맵

경진대회에서 높은 성과를 달성하기 위한 팀 협업 로드맵은 다음과 같은 **4**단계로 구성된다.

3.1. 1단계**:** 데이터 이해 및 전처리 **(**기반 구축**)**

이 단계는 대화 요약 모델의 성공을 위한 가장 중요한 기반을 구축한다.

- 데이터 정제 및 정규화**:**
  - 오타 및 문법 오류 수정, 텍스트 유사성 기반 중복 데이터 제거, 그리고 데이터셋이 일관되게 한국어로 구성되도록 보장하는 표준 자연어 처리 정제 절차를 구현해야 한다.[1] "영어만"이라는 지침은 원본 DialogSum 논문에서 영어에 초점을 맞추었을 수 있으나, 이 한국어 데이터셋의 경우 번역 과정에서 유입될 수 있는 비한국어 아티팩트를 제거하고 언어적 일관성을 보장하는 것을 의미한다. 불필요한 기호를 제거하고, 공백을 정규화하며, 일반적인 오타를 수정하고(신뢰할 수 있는 한국어 오타 사전이나 모델이 있다면), 데이터 누출을 방지하고 모델 일반화를 보장하기 위해 중복된 대화-요약 쌍을 제거해야 한다.
- 화자 정보 및 지시 정보**(Coreference)** 처리**:**
  - 대화 요약에서 매우 중요한 단계이다. #PersonX# 태그는 모델의 입력에 효과적으로 통합되어야 한다.[1] 일반적인 방법으로는 1) 연결**(Concatenation):** 각 발화 앞에 #[PersonX]#: 를 추가하는 방식. 2) 특수 토큰**/**임베딩**:** #PersonX#를 토크나이저의 어휘에 추가하고 모델이 학습하는 별도의 화자 임베딩을 사용하는 방식이 있다. 문서에서 "bi-turn dialogue flow: 동일한 화자의 연속적인 발화를 하나의 발화로 묶어서 항상 A화자가 말하면 다음은 B 화자가 말하는 식으로 구성됨" [1]을 언급하고 있는데, 이는 특정 턴 구조를 의미하며 입력 형식 지정 시 고려되어야 한다. 여러 스니펫에서 화자 정보 활용의 중요성을 반복적으로 강조하고, 트랜스포머 모델의 "부정확한 지시 정보(Incorrect Coreference)" 오류율이 94%로 매우 높다는 점은 화자 식별이 단순한 표면적 태그가 아니라 대화 담화 구조의 근본적인 요소임을 나타낸다.[1] "너", "나"와 같은 대명사는 화자를 알지 못하면 매우 모호하다. 이러한 지시 정보를 올바르게 해결하지 못하는 모델은 사실적으로 부정확하거나 혼란스러운 요약을 생성할 것이다. 이는 단순한 화자 태그 연결만으로는 충분하지 않을 수 있음을 시사한다. 모델은 이 정보를 적극적으로 활용하도록 "학습"되어야 한다. 이는 더 정교한 입력 표현(예: 전용 화자 임베딩 레이어 사용)을 탐색하거나, 모델이 화자 턴을 명시적으로 추적하고 발화 간의 지시 정보를 해결할 수 있도록 하는 아키텍처 수정까지 포함할 수 있다. 이러한 관찰은 입력 토큰화 및 전처리 파이프라인 설계에 직접적인 영향을 미치며, Gemini-CLI 프롬프트의 핵심적인 측면이 된다.
- 개인 정보 마스킹 통합**:**
  - 지정된 8개 PII 토큰(예: #PhoneNumber#, #Address#, #SSN#)이 토크나이저의 어휘에 모두 추가되었는지 확인해야 한다.[1] 이는 이들이 하위 단어로 분해되지 않고 단일의 원자적 단위로 처리되도록 보장하며, 요약 규칙에 따라 생성된 요약에서 이들을 정확하게 보존하는 데 필수적이다.
- 비격식적 언어**(**약어 및 이모티콘**)** 처리**:**
  - 이러한 비격식적 요소를 정규화하거나 제거하기 위한 체계적인 접근 방식을 개발해야 한다.[1] 이는 1) 규칙 기반 대체**:** 일반적인 약어를 전체 형태로 매핑하는 것(예: "ETA"를

"Estimated Time of Arrival"로). 2) 제거 또는 토큰화**:** 이모티콘의 의미론적 기여가 요약에 미미하다면 이모티콘을 제거하거나 일반적인 `` 토큰으로 대체하는 것을 포함할 수 있다. 이는 공식 언어를 요구하는 요약 규칙과 일치한다.

- 정보 분산 처리**(**방해**):**
  - "Excuse me...", "Ummmm...", "Wait..."와 같은 방해 표현의 빈도와 맥락을 분석해야 한다.[1] 만약 이들이 요약에 필수적인 정보를 전달하지 않는 단순한 비유창성이라면, 전처리 과정에서 이들을 제거하거나 노이즈로 처리하는 것을 고려해야 한다. 만약 이들이 주제 변경이나 중요한 일시 중지를 나타낸다면, 다른 표현 방식이 필요할 수 있다. 이 단계는 노이즈를 줄이고 요약 모델의 신호 대 잡음비를 개선하는 것을 목표로 한다.

각 전처리 단계는 단순히 데이터를 정제하는 것을 넘어선다. 이는 원시 대화를 요약 태스크에 중요한 특징들을 명시적으로 강조하는 구조화된 형식으로 변환하는 의도적인 행위이다. 예를 들어, PII 토큰이 분리할 수 없도록 보장하는 것은 핵심 개체를 보존해야 한다는 요약 규칙을 직접적으로 지원한다.[1] 마찬가지로, 화자 정보를 견고하게 처리하는 것은 가장 널리 퍼진 오류 유형(부정확한 지시 정보)을 직접적으로 해결한다.[1] 이러한 관점은 전처리를 일상적인 작업에서 특징 공학의 중요한 단계로 격상시킨다. 전처리 파이프라인의 품질과 정교함은 모델이 대화를 이해하고, 핵심 정보를 식별하며, 정확하고 규칙을 준수하는 요약을 생성하는 능력에 직접적인 영향을 미친다. 이는 Gemini-CLI 프롬프트의 "입력 토큰화" 및 "전처리 코드" 구성 요소가 일반적이지 않고 고도로 맞춤화되어야 함을 강조한다.

다음 표는 필수적인 데이터 전처리 단계와 그 중요성을 설명한다.

전처리 단계	상세 설명	근거
데이터 정제 및 정규화	불필요한 기호 제거, 공백 정규화, 오타 수정, 중복 대화-요약 쌍 제거. 한국어 일관성 유지.	데이터 품질 향상, 모델 학습 효율성 증대, 데이터 누출 방지.[1]
화자 정보 통합	#PersonX# 태그를 각 발화 앞에 추가하거나, 별도 화자 임베딩 사용. \n을 턴 구분자로 활용.	대화의 상호작용성 및 지시 정보(coreference) 해결에 필수적. 가장 흔한 오류 유형 해결.[1]
개인 정보 마스킹 통합	8개 PII 토큰을 토크나이저 어휘에 추가하여 단일 토큰으로 처리.	요약 규칙에 따라 중요한 명명된 개체를 정확하게 보존.[1]

비격식적 언어 처리	약어(ETA, BWT 등)를 전체 형태로 대체, 이모티콘 제거 또는 `` 토큰으로 대체.	요약문의 공식 언어 사용 규칙 준수, 모델 입력 정제.[1]
정보 분산 처리	"Excuse me...", "Ummmm..." 등 발화 중단 표현 분석 및 불필요한 경우 제거.	노이즈 감소, 모델의 핵심 정보 파악 능력 향상.[1]

3.2. 2단계**:** 베이스라인 모델 개발 및 실험 **(**신속한 프로토타이핑**)**

- 초기 모델 선택**:** 추상적 요약에서 강력한 성능을 보여온 잘 확립된 시퀀스-투-시퀀스(Seq2Seq) 모델 아키텍처로 시작해야 한다. 한국어의 경우, Hugging Face transformers 라이브러리를 통해 쉽게 사용할 수 있는 KoBART (monologg/kobart-base-v2) 또는 mBART, mT5와 같은 다국어 모델이 훌륭한 후보이다.
- 신속한 프로토타이핑**:** 1단계에서 정의된 핵심 데이터 로딩 및 전처리 파이프라인을 구현해야 한다. 기본 하이퍼파라미터를 사용하여 공개 테스트 세트에서 초기 성능 베이스라인을 설정하기 위해 기본 모델을 신속하게 학습시킨다. 이는 전체 파이프라인의 초기 검증과 주요 통합 문제 식별을 가능하게 한다. 베이스라인 구축은 단순히 초기 점수를 달성하는 것을 넘어선다. 이는 중요한 진단 단계이다. 현저히 낮은 베이스라인 점수 또는 예상치 못한 동작을 보이는 경우, 데이터 전처리(예: 잘못된 토큰화, 핵심 정보 누락), 모델 구성, 심지어 선택한 모델과 태스크의 본질적인 복잡성 간의 불일치와 같은 근본적인 문제를 즉시 알릴 수 있다. 반대로, 합리적인 베이스라인은 이후의 더 고급 최적화의 영향을 측정할 수 있는 안정적인 시작점을 제공한다. 이 단계는 신속한 반복의 중요성을 강조한다. 목표는 가능한 한 빨리 작동하는 시스템을 엔드-투-엔드로 구축하는 것이다. 이는 데이터 수집부터 평가까지 전체 파이프라인을 검증한다. 이는 모든 향후 개선 사항(고급 모델링부터 하이퍼파라미터 튜닝까지)을 객관적으로 비교할 수 있는 정량적 벤치마크를 설정하여, 팀의 노력이 가장 영향력 있는 개선 사항으로 향하도록 안내한다.
- 입력 표현**:** 베이스라인의 경우, 모든 턴을 해당 화자 태그와 턴 구분자와 함께 연결하는 간단한 대화 입력 형식을 채택한다. 예: #Person1#: 발화1\n#Person2#: 발화2....
- 초기 특징 공학**:** 기본 텍스트 외에, 선택한 모델 아키텍처가 이러한 다중 모달 입력을 지원한다면, 전체 대화 길이 또는 턴 수와 같이 쉽게 추출할 수 있는 간단한 특징을 보조 입력으로 통합하는 것을 고려한다.

3.3. 3단계**:** 고급 모델링 및 개선 **(**성능 최적화**)**

이 단계에서는 모델의 성능을 한 단계 끌어올리기 위한 고급 전략들을 논의한다.

- 계층적 학습**:**
  - 대화는 본질적으로 계층적 구조(개별 발화가 턴을 형성하고, 턴이 완전한 대화를 형성)를 가지므로, 이러한 계층을 명시적으로 인코딩하는 모델은 장거리 의존성과 복잡한 담화 관계를 더 잘 포착할 수 있다.[1] 이 접근 방식은 특히 긴 대화에서 유용하며 [1], 평면 시퀀스 모델이 많은 턴에 걸쳐 컨텍스트를 유지하는 데 어려움을 겪을 수 있는 경우에 효과적이다. 구현에는 2단계 인코더 또는 특수 어텐션 메커니즘이 포함될 수 있다.
- 데이터 증강**:**
  - 모델의 일반화 및 견고성을 향상시키기 위해 새로운 학습 예시를 생성해야 한다.[1] 기술에는 1) 의역**:** 의미를 보존하면서 발화를 다시 작성하는 것. 2) 개체 대체**:** 동일한 유형의 명명된 개체(예: 이름, 위치)를 다른 것으로 대체하면서 지시 정보(coreference)를 유지하는 것. 3) 턴 재정렬**:** 대화의 논리적 흐름이 허용하는 경우 턴을 신중하게 재정렬하는 것이 포함된다. 이는 효과적인 학습 데이터 크기를 확장하고 모델을 더 다양한 대화 패턴에 노출시킨다.
- **QA** 데이터셋 전이 학습**:**
  - 질의응답(QA) 태스크로 사전 학습된 모델은 정보 추출, 핵심 범위 식별, 질문과 답변의 상호작용적 특성 이해에 본질적으로 능숙하다.[1] 이러한 귀납적 편향은 대화 요약에 매우 유용할 수 있으며, 이는 또한 상호작용적 교환 내에서 핵심 정보를 식별하는 것을 요구한다. DialogSum 데이터셋에 이러한 모델을 미세 조정하면 상당한 성능 향상을 가져올 수 있다.
- 일반적인 오류 유형 해결 **(DialogSum** 논문의 오류 분석 기반**):**
  - DialogSum 논문에서 제공된 상세한 오류 분석은 단순한 설명적 목록이 아니다. 이는 정확한 진단 도구 역할을 한다.[1] "부정확한 지시 정보(Incorrect Coreference)"의 압도적인 유병률(트랜스포머의 경우 94%)은 기존 모델이 대화 특유의 현상을 처리하는 방식에 근본적인 한계가 있음을 즉시 지적한다. 마찬가지로 "핵심 정보 누락(Missing Salient Information)"과 "사실 불일치(Unfactual Information)"는 내용 선택 및 사실적 기반의 중요한 실패를 강조한다. 이러한 관찰은 일반적인 요약 접근 방식으로는 최적의 결과를 얻기 어려울 것임을 의미한다. 대신, 팀은 이러한 식별된 오류 유형을 해결하기 위해 특별히 설계된 솔루션을 우선적으로 구축하고 엔지니어링해야 한다. 예를 들어, 정교한 지시 정보 해결 기술(예: 화자 임베딩, 명시적 지시 정보 모듈)에 투자하는 것이 높은 우선순위가 되어야 한다. 환각 현상을 방지하기 위해 복사 메커니즘 또는 사실 일관성 검사를 통합하는 것을 적극적으로 탐색해야 한다. 이는 모델링

단계를 단순히 모델을 학습시키는 것에서 알려진 실패 모드를 적극적으로 해결하기 위한 솔루션을 엔지니어링하는 것으로 전환하여, 더 경쟁력 있는 성능으로 이어진다.

- 부정확한 지시 정보**(Incorrect Coreference):** 전처리 단계의 기본적인 화자 태그 지정 외에, 더 고급 지시 정보 해결 기술을 탐색해야 한다. 여기에는 명시적인 지시 정보 해결 모듈 통합, 지시 정보가 주석 처리된 데이터셋(사용 가능한 경우)에 대한 미세 조정, 또는 강력한 지시 정보 능력을 보여준 모델(예: QA 태스크로 사전 학습된 모델) 활용이 포함될 수 있다.[1]
- 핵심 정보 누락**(Missing Salient Information):** 이는 모델이 대화에서 가장 중요한 정보를 일관되게 식별하고 포함하지 못함을 나타낸다.[1] 이를 완화하기 위한 전략은 다음과 같다: 1) 어텐션 메커니즘 분석**:** 모델이 어디에 초점을 맞추고 있고 핵심 부분을 놓치고 있는지 이해하기 위해 어텐션 가중치를 시각화한다. 2) 내용 선택 모듈**:** 요약 생성 단계 전에 핵심 부분을 식별하기 위한 하위 모듈을 명시적으로 학습시킨다. 3) 손실 함수 튜닝**:** 중요한 정보의 누락에 대해 강력하게 페널티를 부과하는 손실 함수를 실험한다.
- 중복 정보**(Redundant Information):** 생성된 요약 내에서 반복되는 문장이나 구문을 감지하고 제거하는 후처리 단계를 구현해야 한다.[1] 이는 문장 유사성(예: 문장 임베딩의 코사인 유사성 사용)을 계산하고, 이전 문장과의 유사성 임계값을 초과하는 문장을 제거함으로써 달성할 수 있다.
- 사실 불일치**(Unfactual Information,** 환각**):** 이는 모델이 원본 대화에 없는 정보를 생성하는 심각하고 어려운 문제이다.[1] 완화 전략은 다음과 같다: 1) 복사 메커니즘**:** 모델이 사실적 개체와 구문을 처음부터 생성하는 대신 원본 대화에서 직접 복사하도록 장려한다. 2) 사실 확인 모듈**:** 복잡하지만, 생성된 사실을 원본 대화 또는 외부 지식에 대해 검증하는 모듈을 통합한다. 3) 강화 학습**:** 사실적 불일치에 페널티를 부과하는 보상 함수를 설계한다.
- 구문 오류**(Syntactic Error):** 생성된 요약에서 문법 오류나 어색한 구문을 나타낸다.[1] 이는 일반적으로 더 크고 견고한 사전 학습 모델, 더 다양한 학습 데이터, 그리고 확장된 미세 조정을 통해 개선된다. 최후의 수단으로, 고품질 한국어 문법 검사기가 있다면 후처리 단계에서 고려될 수 있다.

다음 표는 고급 모델 학습 전략과 그 적용을 요약한 것이다.

전략	상세 설명	주요 이점	구현 고려사항/과제
계층적 학습	발화, 턴, 대화의 계층 구조를 모델이 명시적으로	긴 대화의 컨텍스트 이해도 향상, 복잡한 담화 관계 포착.[1]	2단계 인코더 또는 특수 어텐션 메커니즘 필요, 구조적 토큰화.

	학습하도록 설계.		
데이터 증강	의역, 개체 대체, 턴 재정렬 등을 통해 유사한 대화문을 생성하여 학습 데이터 확장.	모델 일반화 능력 및 견고성 향상, 과적합 방지.[1]	자연스러운 증강 방법 개발, 증강된 데이터의 품질 관리.
QA 데이터셋 전이 학습	질의응답 태스크로 사전 학습된 모델을 대화 요약에 미세 조정.	정보 추출 및 상호작용 이해 능력 활용, 초기 성능 부스트.[1]	QA 모델의 특정 편향 고려, 태스크 불일치 완화.

다음 표는 대화 요약에서 흔히 발생하는 오류 유형과 그 완화 전략을 제시한다.

오류 유형	식별된 원인	제안된 완화 전략
부정확한 지시 정보	상호작용적 정보 흐름으로 인한 혼란, 대명사 지칭 대상 오인.[1]	화자 정보 명시적 활용, 고급 지시 정보 해결 모듈 통합, 관련 데이터셋 미세 조정.
핵심 정보 누락	대화 내 중요 정보 식별 실패.[1]	어텐션 메커니즘 분석, 내용 선택 모듈 도입, 핵심 정보 누락에 대한 손실 함수 페널티.
중복 정보	생성된 요약 내 반복적인 문장 또는 구문 발생.[1]	후처리 단계에서 문장 유사성 검사 및 중복 문장 제거.
사실 불일치 **(**환각**)**	원본 대화에 없는 정보 생성.[1]	복사 메커니즘 활용, 사실 확인 모듈 탐색, 강화 학습을 통한 사실 일관성 보상.
구문 오류	생성된 요약문의 문법적 오류 또는 어색한 표현.[1]	더 크고 견고한 사전 학습 모델 사용, 다양한 학습 데이터, 한국어 문법 검사기 활용.

3.4. 4단계**:** 평가**,** 반복 및 최적화 **(**지속적인 개선**)**

- 지속적인 평가**:** 공개 테스트 세트에서 모델 성능을 평가하는 엄격한 루틴을 확립해야 한다. 진행 상황을 모니터링하고 추세를 식별하기 위해 핵심 지표(ROUGE-1/2/L

F1)를 시간에 따라 추적해야 한다.

- 상세한 질적 오류 분석**:** 정량적인 ROUGE 점수 외에, 생성된 요약에 대한 심층적인 질적 분석을 수행해야 한다. 요약 샘플을 수동으로 검토하고, DialogSum 논문에서 식별된 유형(예: 지시 정보 문제, 사실 오류, 핵심 정보 누락)을 기반으로 오류를 분류하며, 새로운 패턴을 식별해야 한다.[1] ROUGE 점수는 성공의 정량적 척도를 제공하지만, 모델이 왜 잘 작동하거나 제대로 작동하지 않는지를 설명하지는 않는다. 문서 자체의 오류 분석은 특정 실패 모드를 분류하는 것의 중요성을 강조한다.[1] 인간 전문가가 생성된 요약을 검토하는 질적 분석은 이러한 중요한 "이유"를 제공한다. 이는 ROUGE가 놓칠 수 있는 미묘한 오류, 예를 들어 미묘한 사실 불일치나 어색한 구문을 드러낸다. 이 단계는 선형적이기보다는 순환적인 개선 과정을 강조한다. 질적 오류 분석에서 얻은 통찰력은 데이터 전처리 파이프라인, 모델 아키텍처 선택, 학습 전략, 심지어 이전 단계의 맞춤형 손실 함수 설계에 직접적으로 정보를 제공하고 안내해야 한다. 이는 지속적인 최적화를 위한 견고한 피드백 루프를 생성하여, 단순히 점수를 쫓는 것을 넘어 모델의 약점을 진정으로 이해하고 완화하는 데 기여한다.
- 하이퍼파라미터 튜닝**:** 그리드 검색, 랜덤 검색 또는 더 고급 베이시안 최적화와 같은 기술을 사용하여 모델 하이퍼파라미터(예: 학습률, 배치 크기, 옵티마이저 선택, 드롭아웃 비율)를 체계적으로 최적화하여 최대 성능을 위한 최적 구성을 찾아야 한다.
- 앙상블 방법**:** 여러 모델(예: 다른 아키텍처, 다른 랜덤 시드로 학습된 모델, 또는 데이터의 다른 하위 집합에 미세 조정된 모델)의 예측을 결합하여 집단적 강점을 활용하고 전반적인 견고성 및 성능을 향상시키는 것을 탐색해야 한다.
- 후처리 개선**:** 앞서 논의된 중복 문장 제거 외에, 요약문의 유창성, 간결성 또는 특정 요약 규칙 준수(예: 모든 PII 토큰이 올바르게 형식화되었는지 확인)를 더욱 향상시키기 위한 다른 규칙 기반 또는 모델 기반 후처리 기술을 조사해야 한다.

4. 모델 아키텍처 및 전처리 고려사항

모델 아키텍처의 선택은 입력 데이터의 구조와 모델이 요약 태스크를 인식하는 방식에 근본적인 영향을 미친다.

4.1. 비교 분석**:** 인코더**-**디코더 **vs.** 디코더**-**온리 아키텍처

- 인코더**-**디코더 **(Seq2Seq)** 모델 **(**예**: T5, BART, KoBART, mBART):**

    - 메커니즘**:** 이 모델들은 소스 대화를 처리하여 풍부하고 맥락적인 표현을 생성하는 인코더와 이 인코딩된 표현을 기반으로 토큰별로 요약을 생성하는 디코더의 두 가지 구성 요소로 이루어져 있다.
    - 장점**:** 입력 및 출력 시퀀스의 길이와 구조가 다른 요약과 같은 시퀀스-투-시퀀스 태스크에 본질적으로 적합하다. 입력 정보를 새로운 구문으로 변환하는 추상적 요약에서 뛰어나다. 아키텍처는 긴 입력을 짧은 출력으로 압축하는 것을 자연스럽게 처리한다.
    - 단점**:** 특히 매우 긴 입력 시퀀스의 경우 계산 집약적일 수 있지만, 희소 어텐션(sparse attention)과 같은 발전으로 완화된다. 극도로 긴 대화의 경우 컨텍스트 창 관리에 세심한 주의가 필요할 수 있다.
    - **DialogSum** 관련성**:** 대화 요약이 잠재적으로 긴 대화를 간결한 요약으로 매핑하는 추상적 태스크임을 고려할 때, 인코더-디코더 모델은 태스크 요구 사항과 잘 부합하는 매우 자연스럽고 효과적인 선택이다.
- 디코더**-**온리 모델 **(**예**: GPT-3, Polyglot-Ko, HyperCLOVA):**
    - 메커니즘**:** 이 모델들은 이전의 모든 토큰(입력 프롬프트와 이미 생성된 출력 모두 포함)을 기반으로 각 후속 토큰을 예측하며 자동 회귀적으로 텍스트를 생성한다. 주로 언어 모델링 및 텍스트 완성에 설계되었다.
    - 장점**:** 아키텍처가 더 간단하고, 종종 개방형 텍스트 생성에 매우 강력하며, 태스크를 프롬프트 완성(예: "대화: [대화] 요약:")으로 구성하여 요약에 미세 조정할 수 있다.
    - 단점**:** 표준 자기 어텐션 메커니즘의 2차 스케일링으로 인해 매우 긴 입력 컨텍스트에 어려움을 겪을 수 있으며, 더 공격적인 자르기(truncation) 또는 특수 기술이 필요할 수 있다. 주요 목표가 사실적 준수보다는 유창성이므로, 신중하게 제약하지 않으면 환각(사실이 아닌 정보 생성)에 더 취약할 수 있다. 전용 인코더에 비해 입력이 인코딩되는 방식에 대한 직접적인 제어가 적다.
    - **DialogSum** 관련성**:** 강력한 생성기이지만, 대화 요약에 적용하려면 신중한 프롬프트 엔지니어링과 사실적 일관성 및 요약 길이 제약 준수를 보장하기 위한 추가 메커니즘(예: 복사 메커니즘)이 필요하다. 매우 긴 대화 입력을 관리하는 것은 인코더-디코더 모델보다 더 어려울 수 있다.

대화 요약은 본질적으로 복잡하고 다중 턴의 상호작용적 입력 시퀀스에서 간결한 출력 시퀀스로의 변환이다. 인코더-디코더 모델은 이러한 유형의 시퀀스-투-시퀀스 매핑에 특화되어 있으며, 소스에서 풍부한 맥락 정보를 인코딩하고 추상적 요약을 생성하는 데 뛰어나다. 디코더-온리 모델은 능력이 있지만, 일반적으로 입력을 연속을 위한 프롬프트의 일부로 처리하며, 이는 특히 문서에 언급된 "정보 흐름" 및 "정보 분산" 문제를 고려할 때, 여러 턴에 걸쳐 정보를 깊이 이해하고 추상화하는 데 덜 최적화될 수 있다.[1] 따라서 두 아키텍처 모두 탐색할 수 있지만, 인코더-디코더 아키텍처(또는 하이브리드 접근 방식)가 대화 요약의 고유한 복잡성(긴 상호작용적 입력 처리, 지시 정보 해결 보장, 간결하고 추상적인 출력 생성, 사실 오류와 같은 문제 완화)을 처리하는 데 더

직접적이고 견고하며 제어 가능한 접근 방식을 제공할 수 있다. 아키텍처 선택은 입력 전처리 파이프라인 및 토큰화 전략 설계에 깊은 영향을 미칠 것이다.

4.2. 입력 토큰화 및 전처리 파이프라인에 미치는 영향

모델 아키텍처의 선택은 단순히 선호의 문제가 아니다. 이는 입력 데이터의 구조와 모델이 요약 태스크를 인식하는 방식을 근본적으로 결정한다. 인코더-디코더 모델은 소스 시퀀스를 기대하고 대상 시퀀스를 생성하는 반면, 디코더-온리 모델은 프롬프트를 기대하고 생성을 계속한다. 이는 전처리 파이프라인이 일반적이고 모든 경우에 적용되는 솔루션이 아니라는 것을 의미한다. 컨텍스트 처리, 토큰화, 프롬프트 엔지니어링과 관련된 특정 아키텍처 요구 사항에 맞춰 세심하게 조정되어야 한다. 이러한 관찰은 Gemini-CLI 프롬프트에 매우 중요하다. 프롬프트는 선택한 아키텍처가 입력 형식 지정, 모든 특수 토큰(PII, 화자, 구조적 구분자) 처리, 그리고 시퀀스 길이 관리에 미치는 영향을 명시적으로 전달해야 한다. 이는 생성된 베이스라인 코드가 즉시 호환되고, 선택한 모델 유형에 최적화되며, 데이터 분석에서 도출된 미묘한 요구 사항을 준수하도록 보장한다.

- 일반적인 토큰화 고려사항 **(**두 아키텍처 모두 적용**):**
  - 한국어 토크나이저 선택**:** 강력한 한국어 형태소 분석기(예: Mecab, Okt, Komoran) 또는 대규모 한국어 코퍼스에서 사전 학습된 서브워드 토크나이저(예: SentencePiece, BPE)를 사용하는 것이 중요하다. 토크나이저 선택은 n-그램 중복도에 영향을 미치므로 ROUGE 점수에 직접적인 영향을 미친다.[1]
  - 특수 토큰 처리**:** 모든 8개 PII 마스킹 토큰(예: #PhoneNumber#, #Address#) 및 화자 토큰(예: #Person1#, #Person2#)은 토크나이저의 어휘에 명시적으로 추가되어야 한다.[1] 이는 이들이 단일의 분리할 수 없는 토큰으로 처리되도록 보장하여, 분할을 방지하고 요약에서 정확하게 보존되도록 한다.
- 인코더**-**디코더 특정 전처리**:**
  - 입력 형식**:** 대화는 인코더를 위한 단일 입력 시퀀스로 형식화되어야 한다. 이는 일반적으로 모든 대화 턴을 각 발화 앞에 화자 태그를 붙이고 일관된 턴 구분자(예: \n)와 함께 연결하는 것을 포함한다. 예: #Person1#: 발화1\n#Person2#: 발화2\n.... \n 문자는 특수 토큰으로 처리되거나 토크나이저에 의해 단순히 구분자로 처리될 수 있다.
  - 최대 시퀀스 길이**:** EDA 결과[1]를 기반으로 인코더를 위한 적절한 max_input_length(예: 1024 토큰)와 디코더를 위한 max_output_length(예: 256 토큰, 요약 길이 약 20% 반영)를 정의해야 한다. 자르기(truncation) 및 패딩(padding) 전략은 신중하게 선택되어야 한다.
  - 계층적 인코딩 지원**:** 계층적 학습을 구현하는 경우, 전처리 파이프라인은 대화를 발화와 턴으로 명시적으로 분할하고, 인코더가 활용할 수 있도록 이러한 구조적

경계를 표시하는 특정 토큰을 도입해야 한다.

- 디코더-온리 특정 전처리:
  - 입력 형식 (프롬프트): 대화는 일반적으로 디코더의 생성을 안내하는 프롬프트의 일부로 구성된다. 일반적인 형식은 다음과 같다: #Person1#: 발화1\n#Person2#: 발화2\n.... 모델은 `` 뒤에 요약을 자동 회귀적으로 생성한다.
  - 프롬프트를 위한 특수 토큰: , , ``와 같은 고유한 특수 토큰은 디코더가 처리하는 단일 시퀀스 내에서 입력 및 출력 섹션을 명확하게 구분하는 데 필수적이다. 이들도 토크나이저의 어휘에 추가되어야 한다.
  - 컨텍스트 창 제약: 디코더-온리 모델은 전체 입력 및 출력 시퀀스를 함께 처리한다. 이는 max_sequence_length 제약이 프롬프트와 생성된 요약의 결합된 길이에 적용됨을 의미한다. 이는 공격적인 자르기 없이 처리할 수 있는 최대 대화 길이에 더 엄격한 제한을 부과할 수 있다.

5. Gemini-CLI 베이스라인 코드 생성 프롬프트

목표: 한국어 대화 요약 경진대회를 위한 견고한 베이스라인 프로젝트 구조 및 코드를 생성하며, 선택한 모델 아키텍처(인코더-디코더 또는 디코더-온리)가 입력 토큰화, 데이터 전처리 및 모델 미세 조정에 미치는 영향을 세심하게 고려한다.

프롬프트:

Generate a Python project for a Korean Dialogue Summarization competition.
The task is abstractive summarization of multi-turn dialogues.
Dataset: Korean-translated DialogSum (assume CSV/JSON format with 'dialogue' and 'summary' columns).
Evaluation Metric: ROUGE-1, ROUGE-2, ROUGE-L F1 scores.

Implement a `DataLoader` class that loads data from specified file paths.

For each dialogue, prepend speaker tags (`#Person1#`, `#Person2#`, etc.) to their respective utterances. Ensure `\n` acts as a turn separator. Example: `#Person1#: 안녕하세요.\n#Person2#: 네, 안녕하세요.`

Add the following 8 special tokens to the tokenizer's vocabulary and ensure they are treated as single tokens: `#PhoneNumber#`, `#Address#`, `#DateOfBirth#`, `#PassportNumber#`, `#SSN#`, `#CardNumber#`, `#CarNumber#`, `#Email#`.

Include a placeholder function or comment for handling potential abbreviations and emoticons in the dialogue text, noting the need for normalization or removal.

Use `AutoTokenizer.from_pretrained()` with the chosen base model's identifier. Ensure the added special tokens are correctly passed to `tokenizer.add_special_tokens()`.

Set `max_input_length` to 1024 tokens and `max_output_length` to 256 tokens.
Implement truncation for inputs exceeding `max_input_length` and padding to `max_input_length`.

Load the specified pre-trained model for summarization/causal language modeling.
Implement a standard fine-tuning loop using Hugging Face `Trainer`.
Use `AdamW` optimizer.
Include ROUGE metric calculation (using `evaluate` library) during evaluation.
Save best model checkpoint based on validation ROUGE-L F1.

Project Structure:
- `main.py`: Main script for training and evaluation.
- `data_loader.py`: Handles data loading, preprocessing, and `Dataset` creation.
- `model.py`: For model initialization and specific architectural configurations.
- `utils.py`: Contains helper functions, metric computation, and potentially custom callbacks.
- `config.py`: Stores hyperparameters and model paths.
- `requirements.txt`: Lists all necessary Python packages.

Choose ONE of the following architectural options:

--- OPTION A: Encoder-Decoder Architecture ---
Base model: `monologg/kobart-base-v2` (KoBART) from Hugging Face Transformers.
Use `AutoModelForSeq2SeqLM`.
Configure `Seq2SeqTrainingArguments` and `DataCollatorForSeq2Seq`.
Ensure the decoder uses `decoder_start_token_id` (typically `tokenizer.bos_token_id` or `tokenizer.pad_token_id`).

--- OPTION B: Decoder-Only Architecture ---

Base model: `EleutherAI/polyglot-ko-1.3b` from Hugging Face Transformers.
Use `AutoModelForCausalLM`.
Configure `TrainingArguments` and `DataCollatorForLanguageModeling`.
Format input as: ` {dialogue_text} {summary_text}`. Add ``, ``, `` as special tokens to the tokenizer.
Ensure the loss calculation correctly masks the prompt part and only computes loss on the summary part.
For inference, generate text by prompting with ` {dialogue_text}`.

프롬프트 구성 요소 설명**:**

이 프롬프트는 Gemini-CLI가 경진대회의 특정 요구 사항에 맞춰진 코드를 생성하도록 안내한다. 프롬프트의 품질과 구체성은 생성된 코드의 유용성과 정확성을 직접적으로 결정한다. 일반적인 프롬프트는 상당한 수동 적응이 필요한 일반적인 코드를 생성할 것이다. 고도로 맞춤화되고 즉시 사용 가능한 베이스라인을 얻기 위해서는 프롬프트가 데이터 분석, 아키텍처 고려 사항 및 경진대회 규칙(예: 특정 토큰 유형, 입력 형식, 모델 유형, 평가 지표 및 오류 완화 전략)에서 파생된 모든 미묘한 기술 요구 사항을 캡슐화해야 한다. 이는 프롬프트에 무엇을 포함할 뿐만 아니라 어떻게 구조화하고 명확하게 표현할 것인지의 중요성을 강조한다. 특수 토큰 처리, 데이터 흐름에 대한 아키텍처적 영향 지정, 원하는 프로젝트 구조 명확화에 대한 명시적인 지침의 중요성이 강조된다. 이러한 세심한 접근 방식은 AI의 출력이 기능적일 뿐만 아니라 경진대회의 전략적 목표와 일치하도록 보장하여, 생성 후 수동 조정을 최소화한다.

- 프로젝트 컨텍스트 및 목표**:** 경진대회의 본질, 태스크 유형, 데이터셋 및 평가 지표를 명확히 정의하여 AI가 올바른 도메인과 목표를 이해하도록 한다.
- 데이터 로딩 및 전처리 요구 사항**:**
  - DataLoader 클래스 구현은 데이터 파이프라인의 모듈화를 장려한다.
  - 화자 정보 통합**:** 대화의 핵심인 화자 정보를 모델이 명확하게 인식하도록 입력 형식을 지정한다. 이는 지시 정보 해결의 정확도를 높이는 데 필수적이다.[1]
  - **PII** 마스킹 처리**:** PII 토큰을 토크나이저의 어휘에 추가하도록 지시함으로써, 이들이 중요한 개체로 인식되어 요약에 정확히 포함되도록 한다.[1]
  - 비격식적 언어 처리**:** 잠재적인 비격식적 표현에 대한 처리 필요성을 명시하여, 모델 출력이 공식 언어 규칙을 따르도록 유도한다.[1]
  - 토큰화**:** AutoTokenizer 사용 및 특수 토큰 추가 지시는 Hugging Face 생태계와의 호환성을 보장하고, 사용자 정의 토큰이 올바르게 처리되도록 한다.
  - 시퀀스 길이**:** max_input_length 및 max_output_length 설정은 모델이 처리할 수 있는 데이터의 범위를 정의하고, 메모리 사용량 및 학습 효율성에 영향을 미친다.
- 모델 설정 및 학습 루프**:** Hugging Face Trainer를 사용한 표준 미세 조정 루프, AdamW 옵티마이저, ROUGE 지표 계산 및 최고 모델 체크포인트 저장 지시는 일반적인 NLP 학습 모범 사례를 따른다.

- 프로젝트 구조**:** 명확한 파일 구조를 제시하여 코드의 가독성, 유지 보수성 및 팀 협업 용이성을 높인다.
- 아키텍처별 지시**:**
  - 인코더-디코더 **(**옵션 **A):** AutoModelForSeq2SeqLM 사용, Seq2SeqTrainingArguments, DataCollatorForSeq2Seq 및 decoder_start_token_id 설정은 추상적 요약에 최적화된 시퀀스-투-시퀀스 모델의 표준 구성이다.
  - 디코더-온리 **(**옵션 **B):** AutoModelForCausalLM 사용, TrainingArguments, DataCollatorForLanguageModeling 및 특정 프롬프트 형식 지정은 디코더-온리 모델의 자동 회귀적 생성 특성을 활용하기 위한 필수 구성이다. 손실 계산에서 프롬프트 부분을 마스킹하고 요약 부분에만 손실을 계산하도록 하는 지시는 모델이 불필요한 부분에 집중하지 않고 요약 생성에만 집중하도록 한다.

6. 주요 권고 사항 및 향후 연구

6.1. 핵심 실행 권고 사항

- 견고한 전처리 우선순위**:** 화자 정보의 지능적 통합과 PII 마스킹의 정밀한 처리를 포함한 세심한 데이터 전처리는 단순한 예비 단계가 아니라 핵심적인 경쟁 우위이다. 이러한 단계는 가장 널리 퍼진 오류 유형과 평가 기준을 직접적으로 해결한다.
- 강력한 베이스라인 신속 구축**:** 적절한 사전 학습된 한국어 Seq2Seq 모델을 사용하여 기능적인 베이스라인을 신속하게 구현하도록 권고한다. 이 베이스라인은 중요한 진단 도구이자 이후의 모든 최적화를 위한 측정 가능한 벤치마크 역할을 한다.
- 표적화된 오류 완화에 집중**:** DialogSum 논문의 오류 분석을 기반으로, 부정확한 지시 정보, 핵심 정보 누락, 사실 불일치를 해결하기 위한 전략을 적극적으로 구현해야 한다. 이들은 주요 성능 병목 현상으로 식별되었으며, 이를 해결하는 것은 상당한 개선을 가져올 것이다.
- 반복적 개선 수용**:** 모델 학습, 포괄적인 평가(정량적 ROUGE 점수 및 질적 오류 분석), 심층적인 오류 진단, 그리고 후속 모델 개선의 지속적인 순환을 옹호한다. 이러한 반복적 접근 방식은 지속적인 성능 향상의 핵심이다.

6.2. 추가 탐색 및 경쟁 우위를 위한 제안

- 고급 지시 정보 해결**:** 기본적인 화자 태그 외에, 모호한 대명사 및 개체를 더 견고하게 해결하기 위해 전용 지시 정보 해결 모듈 또는 지시 정보가 주석 처리된 데이터셋을 활용하는 미세 조정 기술을 탐색해야 한다.
- 담화 관계 모델링**:** DialogSum 논문에서 제안된 바와 같이 [1], 대화 내에서 담화 관계(예: 인과 관계, 시간 순서, 양보)를 명시적으로 인코딩하거나 학습하는 모델을 조사하여 대화의 근본적인 구조와 의도에 대한 모델의 이해를 향상시켜야 한다.
- 다중 작업 학습**:** 요약과 함께 대화 행위 인식, 주제 분류 또는 감성 분석과 같은 보조 작업에 모델을 학습시키는 것을 고려한다. 이는 추가적인 감독 신호를 제공하여 모델이 대화 의도 및 컨텍스트에 대한 더 풍부한 표현을 학습하도록 강제할 수 있다.
- 외부 지식 통합**:** 사실 불일치를 더욱 방지하고 요약의 풍부함을 향상시키기 위해, 특히 명명된 개체 및 도메인별 정보에 대해 외부 지식 베이스를 통합하는 방법을 탐색해야 한다.
- **Human-in-the-Loop** 평가**:** 중요하고 미묘한 오류 분석을 위해, 특히 사실적 정확성과 유창성을 평가하기 위해 자동화된 ROUGE 점수 이상의 더 세분화된 피드백을 제공할 수 있는 인간 주석가를 참여시키는 것을 고려한다.
- 앙상블 및 모델 증류**:** 최종 제출을 위해, 여러 강력한 모델을 결합하는 앙상블 방법을 탐색하여 더 높은 견고성을 확보해야 한다. 경진대회 후에는 더 크고 고성능 모델을 더 작고 효율적인 모델로 압축하여 배포하기 위한 모델 증류 기술을 조사해야 한다.

참고 자료

1. NLP_-_Dialogue_Summarization.pdf