**Slide 1: 제목 페이지**

* **프로젝트 제목**: CRAG(Corrective RAG)를 활용한 뉴스/공시 기반 주가 분석 AI 에이전트 개발
* **과목명**: 초거대언어모델의 이해와 응용
* **발표자**: 한영재
* **담당교수**: 김재형 교수님
* **제출일**: 2025년 6월 13일

**Slide 2: 목차 (Table of Contents)**

1. 서론: 문제 정의 및 프로젝트 목표
2. 관련 연구: 기존 RAG의 한계점
3. **제안 방법론: CRAG 기반 분석 에이전트**
4. **실험 설계 및 평가 절차**
   * 데이터 수집 및 전처리
   * 비교 모델 및 프롬프트 설계
   * 평가 지표 및 방법론
5. 실험 결과 및 분석
   * 정량적 평가: CRAG vs Standard RAG
   * 모델별 성능 비교 분석
   * 사례 연구: '모호한 상황'에서의 성능 분석
6. 결론 및 보완점
7. Q&A

**Slide 3: 서론 (Introduction)**

* **문제 정의**
  + 금융 시장의 정보 비대칭성: 방대한 양의 정형(주가) 및 비정형(뉴스/공시) 데이터가 실시간으로 발생하여 일반 투자자가 종합적으로 분석하기 어려움.
  + 기존 LLM의 한계: 지식 단절(Knowledge Cutoff) 문제로 인해 최신 금융 동향을 실시간으로 반영한 분석에 명백한 한계를 가짐.
* **프로젝트 목표**
  + 실시간 주가 데이터와 관련 뉴스/공시를 종합적으로 분석하는 RAG 시스템 구축.
  + 단순 정보 나열을 넘어, **부정확하거나 불충분한 정보를 스스로 보정(Corrective)하고 깊이 있는 인과관계를 추론**하는 고도화된 AI 분석 에이전트 개발을 목표로 함.

**Slide 4: 관련 연구 (Background)**

* **검색 증강 생성 (RAG)**: LLM의 환각 현상을 줄이고 최신성을 확보하기 위한 핵심 기술.
* **표준 RAG의 한계**:
  + **Noisy Retrieval 문제**: 검색된 정보의 품질을 신뢰해야만 함. 관련 없거나 오해의 소지가 있는 뉴스가 검색될 경우 분석의 질이 저하될 수 있음.
  + **Lost in the Middle**: 여러 정보가 주어졌을 때, LLM이 중간에 위치한 핵심 정보를 놓치는 경향이 있음.
* **LLM-as-a-Judge**: 인간 평가자를 대체하여 LLM의 생성물을 객관적으로 평가하기 위한 방법론으로, 본 프로젝트의 핵심 평가 도구로 사용함.

**Slide 5: 제안 방법론: CRAG 기반 분석 에이전트**

* **전체 아키텍처 다이어그램 제시**:
  + (1) **초기 검색 (Initial Retrieval)**: 사용자의 요청(종목, 날짜)에 따라 주가 데이터와 관련 뉴스, 공시를 1차적으로 수집합니다.
  + (2) **품질 평가 (Assess Retrieval Quality)**: 1차 수집된 정보가 특정 주가 변동(이벤트)을 설명하기에 충분한지, 관련성이 높은지 LLM이 자체적으로 평가합니다.
  + (3) **조건부 보정 (Conditional Correction)**: 품질 평가 결과 '불충분' 또는 '관련성 낮음'으로 판단될 경우, **더 구체적인 키워드로 웹 검색을 추가 수행**하여 정보를 보강합니다. 상반된 뉴스가 있을 경우, 이를 종합하여 재구성합니다.
  + (4) **최종 생성 (Final Generation)**: 정제되고 보강된 정보를 바탕으로 **시간적 인과관계에 초점을 맞춘 프롬프트**를 구성하여 최종 분석 리포트를 생성합니다.

**Slide 6: 실험 설계 및 평가 절차**

이 슬라이드에 보내주신 평가 절차를 상세히 기술하여 실험의 신뢰성을 강조합니다.

* **1. 데이터 수집 및 전처리**
  + **공정성 확보**: 모든 비교 모델이 동일한 데이터를 기반으로 분석하도록 데이터를 사전 수집 및 정제함.
  + **주가 데이터**: Intraday 1분봉 데이터 사용.
  + **이벤트 탐지**: 주가 변동률 0.6% 이상 지점을 분석할 핵심 이벤트로 자동 탐지.
  + **뉴스/공시 수집**: Naver 뉴스 API와 DART Open API를 활용하여 이벤트 발생 시점 기준 72시간 내의 모든 관련 정보를 수집함.
* **2. 비교 모델 및 프롬프트 설계**
  + **비교군 (Baselines)**: No RAG, Standard RAG
  + **실험군 (Proposed)**: CRAG (본 프로젝트 모델)
  + **사용 LLM**: Gemini 1.5 Flash, Llama 3.1 70B (Groq API 경유)
  + **프롬프트 차별화**:
    - **Standard RAG**: "오늘의 뉴스/공시와 주가 등락 정보를 바탕으로 종합 분석해줘." (단순 결합)
    - **CRAG**: "탐지된 O시 O분 주가 변동 이벤트는 해당 시점 이전에 발생한 다음 뉴스/공시들과 어떤 시간적 인과관계가 있는지 분석해줘." (인과관계 추론 강화)
* **3. 평가 지표 및 방법론**
  + **A. LLM-as-a-Judge**: 생성된 리포트의 논리성, 통찰력 등 정성적인 품질을 GPT-4를 통해 평가하여 \*\*승률(Win Rate)\*\*을 측정.
  + **B. 규칙 기반 자동 평가 (Heuristic Scoring)**: 아래 4가지 항목을 기준으로 객관적인 점수를 산출함.
    - 분석 길이 적정성 (500~2000자, +20점)
    - 구조화 수준 (볼드, 불릿 등 사용, +10~30점)
    - 핵심 키워드 포함 ("인과관계", "영향" 등, +5~15점)
    - 이벤트/뉴스 매칭 수 (+10~20점)

**Slide 7: 실험 결과 (1) - 정량적 종합 성능**

* **핵심 결과**: **CRAG는 모든 테스트 모델과 시나리오에서 Standard RAG 대비 압도적인 성능 우위를 보였습니다.**
* **시각화 자료 1 (종합 점수 비교)**: Gemini와 Llama 3 모델 각각에 대해, Standard RAG와 CRAG의 '규칙 기반 자동 평가 점수'를 비교하는 막대그래프.
* **시각화 자료 2 (승률 비교)**: LLM-as-a-Judge 평가 결과를 바탕으로, '명확한 케이스'와 '모호한 케이스'에서의 CRAG 승률을 비교하는 그래프. (모호한 케이스에서 성능 격차가 더 크게 벌어지는 점을 시각적으로 강조)

**Slide 8: 실험 결과 (2) - 모델별 심층 분석**

* **모델별 강점 분석**:
  + **Gemini 1.5 Flash**: 복잡한 인과관계를 논리적으로 추론하고, 여러 정보를 종합하여 깊이 있는 분석을 생성하는 데 강점을 보임.
  + **Llama 3.1 70B**: 빠른 응답 속도와 함께 안정적인 성능을 보였으나, 때때로 특정 뉴스의 미묘한 시장 뉘앙스를 포착하는 데는 다소 아쉬움을 보임.
* **시사점**: 제안하는 **CRAG 방법론은 특정 모델 아키텍처에 국한되지 않으며, 다양한 LLM의 추론 능력을 전반적으로 향상시키는 범용적인 프레임워크**로서의 가능성을 확인 함.

**Slide 9: 결론 및 보완점 (Conclusion & Future Work)**

* **결론**:
  + 본 프로젝트는 검색 결과의 품질을 스스로 평가하고 보정하는 CRAG 기반 주식 분석 시스템을 성공적으로 구현함.
  + 정량적 평가(규칙 기반 점수, LLM-as-a-Judge 승률)를 통해 제안 시스템이 기존 RAG 대비, 특히 정보가 불확실한 '모호한 상황'에서 분석의 정확성과 신뢰도를 크게 향상시킴을 입증함.
  + 다양한 LLM 모델에 적용하여 방법론의 범용성을 확인함.
* **보완점 및 향후 연구**:
  + **고성능/도메인 특화 LLM 활용**: 이번 테스트는 접근 가능한 무료 버전 API를 기준으로 모델을 선정했지만, 금융 분석에 특화된 BloombergGPT나 더 강력한 상용 모델(GPT-4o, Claude 3 Opus 등)을 활용했다면, 분석의 깊이와 정확도가 한층 더 향상될 가능성이 있음. 이는 향후 연구에서 탐구해볼 만한 흥미로운 지점임.
  + **장기 시계열 분석**: 현재는 '일일' 분석에 초점을 맞추었으나, 주간/월간 추세를 분석하고 장기적인 예측을 수행하는 기능으로 확장할 수 있음.
  + **비용 및 속도 최적화**: CRAG의 교정 단계는 분석 품질을 높이는 대신 추가적인 API 호출 비용과 시간이 발생함. 품질과 비용/속도 간의 최적의 균형점을 찾는 연구가 필요함.

**Slide 10: Q&A**

* **감사합니다**
* **질의응답**