|  |
| --- |
| **1. 주제**  다중 AI 모델 협업지능(Collective Intelligence)을 위한 자동 라우팅 및 응답 합성 시스템 설계  **나분반, 8팀, 20251781, 진영헌** |

|  |  |
| --- | --- |
| **2. 요약**  본 프로젝트의 목표는 단일 AI모델의 한계를 극복하기 위해 여러 AI모델을 통합하여 자동으로 질의의 난이도와 주제에 따라 최적의 모델을 자동 선택하거나 복수 모델의 응답을 비교/결합하는 BlendBot을 개발하는 것이다.  핵심내용은 난이도 분석기(Analyzer)을 통해 문장의 복잡도와 도메인을 판별하고, 라우팅 엔진(Router)이 가장 적합한 모델 또는 조합을 자동 지정하며, 조율기(OrchestrAI)가 여러 모델의 응답을 평가/합성해 최종 결과를 산출하는 세 단계 구조로 구성된다.  본 프로젝트의 중요성은 단일 AI 모델의 편향과 한계를 보완하고, 이를 통해 사용자는 각 모델의 장단점을 비교하거나 직접 조정할 필요 없이, 하나의 플랫폼에서 가장 최적화된 응답을 얻을 수 있다. BlendBot은 향후 AI 간 협업지능의 기반이 되는 오픈소스 플랫폼으로 발전할 가능성을 지니며, 인공지능 기술이 “경쟁에서 조율과 협력”으로 전환되는 시대적 흐름을 선도할 것으로 기대된다. | **3. 대표 그림**    그림 1. 개발 배경(단일 모델의 한계와 기능에 따른 AI모델 교체 사용의 번거로움)    그림 2. 예상결과 |

|  |
| --- |
| **4. 서론**  최근 인공지능(AI) 기술은 산업·학문·사회 전반에서 인간의 지식 활동을 보조하거나 대체하는 수준으로 발전하였다. 그러나 이러한 AI 모델의 급속한 발전은 동시에 새로운 문제를 낳고 있다. 대표적인 한계는 모델 간 편향성과 기능 분화다. 예를 들어 OpenAI의 GPT-4는 논리적 추론과 코드 생성에서 뛰어나지만 이미지나 동영상 생성에서 약점을 보인다. 따라서 하나의 모델만 사용한다면 편향·비일관성·도메인 한계가 발생한다. 나 역시 이러한 한계를 직접 경험한 바 있다. 평소에는 GPT 시리즈만 주로 사용했지만, 이미지 생성이 필요해 GPT에게 시도했을 때 생성된 결과물의 완성도가 낮아 실망한 적이 있었다. 결국 이미지를 생성하기 위해 Stable Diffusion이나 Runway Gen-4와 같은 이미지 특화 AI 모델을 별도로 찾아 사용해야 했다. 이 과정에서 단일 모델에 의존하는 불편함과 매번 다른 AI모델을 따로 사용하는 것에 대한 비효율성을 체감했고, “질문이나 목적에 따라 가장 적합한 모델이 자동으로 선택된다면 어떨까?”라는 문제의식이 본 프로젝트의 출발점이 되었다. 최근 연구 SCE: Scalable Consistency Ensembles Make Blackbox Large Language Model Generation More Reliable(2025)에서는 복수의 LLM을 조합하면 사실성과 일관성이 향상된다고 보고하였다. 이는 다중 모델 협업형 AI 시스템의 필요성을 뒷받침한다. 이에 본 프로젝트 BlendBot은 “다중 AI 모델의 자동 라우팅 및 합성 응답을 통해 사용자의 질의에 최적화된 결과를 생성하는 플랫폼”을 목표로 한다. 사용자의 질문을 분석하여 여러 AI 모델 중 최적의 조합을 자동 선택하고, 복수 모델의 응답을 비교·평가·결합해 가장 신뢰도 높은 결과를 생성하는 플랫폼을 개발할 것이다. |

|  |
| --- |
| **5. 본론**  1. 시스템 개요  BlendBot의 전체 시스템 구조는 사용자의 입력 질의를 분석하고, 이를 바탕으로 **최적의 모델을 선택하거나 복수의 모델 응답을 합성**하여 최종 결과를 제공하는 방식으로 구성된다. 시스템의 동작은 다음과 같은 다섯 단계로 구분된다.    이러한 흐름을 통해 BlendBot은 질문 난이도별 동적 라우팅(Dynamic Routing)과 다중 응답 합성(Ensemble Response)이라는 두 핵심 기능을 수행한다. 단일 AI 모델이 특정 영역에서 편향된 출력을 내는 문제를 방지하고, 모델 간의 장단점을 상호보완적으로 결합함으로써 신뢰도 높은 결과를 생성할 수 있다.  2. 핵심 기술 요소 및 구현방법  1) 질문 난이도 분석기(Analyzer.py)  질문 난이도 분석기는 사용자의 입력 문장을 형태소 단위로 분해하고, **spaCy의 언어 복잡도 지표**와 **scikit-learn의 TF-IDF 기반 도메인 분류기**(Pedregosa et al., 2011)를 이용해 질문을 ‘일반 질의’, ‘기술 질의’, ‘창의적 요청’ 등으로 분류한다. 예를 들어 “전기회로에서 저항 병렬 연결의 전류 분배를 설명하라”는 기술적·분석형 질문으로 분류되어 GPT-4와 Claude가 병렬 호출된다. 반면 “봄을 주제로 한 시를 써줘”는 창의형 질의로 분류되어 Claude 또는 Gemini로 라우팅된다.  # 구현방법 의사코드    2) 모델 라우팅 엔진(Router.py)  라우팅 엔진은 각 모델의 성능 프로파일(performance profile)을 데이터베이스 형태로 저장하고, 입력 질문의 메타데이터와 비교해 가장 적합한 모델을 점수화한다. 이때 모델 점수는 다음 요소로 계산된다.  (C: 복잡도 적합도, D: 도메인 일치도, L: 모델 신뢰도)  이 가중합 기반 라우팅 구조는 **모델의 선택적 활성화(sparse expert routing)** 개념을 응용한 것이다. α, β, γ는 사용자가 설정 가능한 가중치이며, *config.ini*에서 조정 가능하다. 라우팅은 단일 선택(single-mode) 또는 다중 조합(hybrid-mode)으로 작동하며, hybrid-mode에서는 두 개 이상의 모델을 병렬 호출한다.  #구현 방법 의사코드(테스트용으로 3개(gpt-4o, claude-3.5, gemini-1.5-pro)만 사용한 버전)    3) 응답 합성기(JudgeAI)  OrchestrAI는 BlendBot의 핵심 알고리즘으로, 여러 모델의 응답을 비교·평가하여 최적의 최종 답변을 도출한다. 이 모듈은 “응답 평가기(response evaluator)”와 “가중 합성기(weighted synthesizer)”의 두 부분으로 구성된다.   * **응답 평가기:** 각 모델 응답을 문법 정확성(Grammar), 사실성(Factuality), 논리적 일관성(Coherence) 항목으로 점수화한다. 이러한 평가 방식은 *truthful and consistent decoding* 구조를 기반으로 한다. * **가중 합성기:** 평가 점수 기반으로 응답별 가중치를 부여하고, 텍스트 유사도 알고리즘(cosine similarity)을 사용해 공통된 내용이 많은 부분을 우선 결합한다.   이 과정을 통해 BlendBot은 다수의 응답을 단순 나열하는 것이 아니라, 지식적 합의  (consensus)”를 도출하는 구조를 구현한다. #응답 평가기 구현방법 의사코드 #가중 합성기 구현방법 의사코드    # OrchestrAI 구현방법 의사코드    3. 개발 방향  BlendBot은 여러 AI 모델을 **조율(Orchestrate)** 해서 가장 좋은 답을 만드는 방향으로 개발된다. 초기에는 6개의 AI모델을 활용할 예정이다. (GPT-4o, Claude 3.5, Gemini 1.5 Pro, Stable Diffusion XL, Runway Gen-4, Whisper)  시스템 구조는 **모듈화(Modular)** 되어 있어 모델을 쉽게 추가·교체할 수 있으며, , .env를 이용해API 키를 안전하게 관리하는 방향으로 개발될 것이며, 사용자가 BlendBot이 사용하는 AI모델의 API모두를 가지고 있지 않다해도 사용자가 지닌 AI모델만을 사용해 작동될 수 있도록 개발될 것이다.  기존의 Microsoft AutoGen 같이 모델끼리의 한정적 협업을 하는 모델은 이미 존재하지만, 동일 계열 모델 간 협업에 한정되어 있다. 반면 BlendBot은 자동 판단 기반 라우팅과 응답 합성을 통합한 플랫폼이라는 점에서 차별화된다는 점에서 BlendBot을 개발하려 한다. |

|  |
| --- |
| **6. 결론**  본 프로젝트 BlendBot은 여러 AI 모델의 협업을 통해 단일 모델의 한계를 보완하고, 자동 라우팅과 응답 합성을 통해 가장 신뢰도 높은 결과를 생성하는 시스템이다. 프로젝트의 핵심 구조는 질문 분석기(Analyzer), 모델 라우팅 엔진(Router), 응답 합성기(OrchestrAI) 의 세 모듈로 구성되며, 입력된 질의의 난이도와 도메인을 자동 판별하여 최적의 모델 조합을 선택하고, 그 결과를 가중 평가·합성하는 과정으로 구현된다.  향후 기능 고도화와 안정성 확보를 위해 라우팅 알고리즘 개선, JudgeAI(OrchestrAI) 고도화, 모듈 간 비동기 통신 및 캐싱 최적화, Streamlit UI 통합, API 구조 정립 및 문서화에 초점을 둘 것이다. |

**7. 출처**

[1] 허균, 임꺽정, “홍길동의 얼굴 분석,” 한국OOO논문지, 제5권, 제6호, pp. 1-10, 2006.

Fedus, W., Zoph, B., & Shazeer, N. (2022). *Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity.* *Journal of Machine Learning Research*, 23(120), 1–40.

**Zhang, J., Li, Z., Cui, W., Das, K., Malin, B., & Kumar, S. (2025). SCE: Scalable consistency ensembles make blackbox large language model generation more reliable pp. 1–12.**

Honnibal, M., & Montani, I. (2020). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

**Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011).** *Scikit-learn: Machine learning in Python.* *Journal of Machine Learning Research, 12*, 2825–2830, pp. 1-6.

**Fedus, W., Zoph, B., & Shazeer, N. (2022).** *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity.* *Journal of Machine Learning Research, 23*(120), lbid., pp. 1-40.

Welleck, S., Kulikov, I., Kim, J., & Cho, K. (2020). Consistency of a Recurrent Language Model with Respect to Incomplete Decoding. *EMNLP 2020*, pp. 1-16.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT networks.* *EMNLP 2019*, pp. 1-11.