6th Assignment:

Real Usage and Final Report

Aviation News Caption Generator

이찬 (202400938, 언어인지과학과)

영어분석을위한기계학습

December 12, 2025

1. 프로젝트 개요

1.1. 프로젝트 주제 및 선정 배경

본 프로젝트 '항공 뉴스 캡션 자동 생성기'는 항공 운항 정보 사이트인 'Aeroroutes'의 기사를 분석하여, 소셜 미디어(인스타그램)에 즉시 업로드 가능한 형태의 텍스트로 자동 변환하는 자연어 처리(NLP) 기반 서비스를 만들고자 하였다.

항공 산업의 데이터는 정보의 효율성을 위해 고도로 압축된 약어와 IATA/ICAO가 정한 고유 코드로 작성되는 특징이 있다. 예를 들어, "BX6215 PUS2200 – 0305+1SPN 321 6"와 같은 텍스트는 일반 대중이 직관적으로 이해하기 어렵다. 이러한 '데이터의 비친화성'은 정보 전달의 장벽이 되며, 이를 대중적인 언어로 번역하고 가공하는 과정에는 상당한 시간이 소요된다.

본 프로젝트는 이러한 '전문 데이터와 대중적 정보 소비 사이의 간극'을 AI 기술로 해결하고자 기획되었다.

1.2. 문제 정의

프로젝트 착수 전, 기존 수작업 프로세스에서 다음과 같은 구체적인 문제점들을 식별하였다.

1) 해독의 어려움: 77W(보잉 B777-300ER), ICN(인천국제공항), x246(화/목/토 제외 운항) 등 도메인 지식이 없으면 해석이 불가능한 용어들이 만연했다.
2) 높은 리소스 소모: 기사 한 건을 선정하여 코드를 해석하고, 한국어로 번역한 뒤, 이모지 등을 활용해 가독성 있는 캡션으로 작성하는 데 평균 30분 가까이 소요되곤 했다.
3) 휴먼 에러 가능성: 수많은 공항 코드를 수동으로 검색하여 변환하는 과정에서, 오타나 오역이 발생할 가능성이 존재한다.

1.3. 프로젝트 목표

위 문제들을 해결하기 위해 다음과 같은 세부 목표를 수립하고 달성하고자 하였다.

1) 기술적 목표: 다국어 처리에 특화된 BERT 모델을 항공 도메인 데이터로 파인 튜닝(Fine-tuning)하여, 텍스트 내에서 항공사, 기종, 날짜, 노선 등의 핵심 개체명(Entity)을 정확히 추출하는 NER(개체명 인식) 모델을 구축한다.
2) 시스템적 목표: 딥러닝 모델의 확률적 불확실성을 보완하기 위해, 국토교통부 공공데이터(DB)와 정규표현식을 결합한 하이브리드 처리 파이프라인을 설계하여 정보의 정확도를 높인다.

2. 진행 과정

2.1. 데이터 수집

양질의 학습 데이터를 확보하기 위해 Python의 Requests와 BeautifulSoup 라이브러리를

활용하여 자체 크롤러를 제작하였다. 타겟 사이트인 'Aeroroutes'에서 최근 1년간 발행된 전 세계 항공 운항 뉴스 약 1,000건을 수집하여 원문 데이터셋을 구축하였다.

2.2. 데이터 전처리 및 오토 라벨링

지도 학습을 수행하기 위해서는 정답 데이터(Label)가 필수적이다. 그러나 1,000건의 데이터를 일일이 수작업으로 태깅하는 것은 비효율적이므로, '정규표현식을 활용한 오토 라벨링 기법'을 도입하고, 일부 샘플에 대해서는 수작업 검수를 통해 레이블을 확인하였다.

구체적으로는 항공기 기종 패턴(예: A3\d{2}), 항공사 코드 패턴(예: [A-Z]{2}) 등을 사전에 정의하여, 텍스트 내에서 해당 패턴을 자동으로 찾아 태깅 처리를 수행하였다.

2.3. 머신러닝 모델 도입의 당위성

단순 규칙 기반 처리가 가능함에도 불구하고, BERT 모델을 학습시킨 이유는 다음과 같다.

1) 규칙의 경직성 극복: 정규표현식은 사전에 정의된 패턴만 찾을 수 있다. 만약 기사에서 오타가 발생하거나, 공식 명칭 대신 별칭이 사용되거나, 예외적인 형식이 등장할 경우 규칙 기반 시스템은 이를 인식하지 못한다. 반면, 머신러닝 모델은 학습을 통해 유사한 패턴을 스스로 추론하는 일반화 능력을 갖는다.
2) 문맥 기반의 중의성 해소: 예를 들어 숫자 '333'은 상황에 따라 기종(A330-300)일 수도, 편명(OZ333)일 수도, 단순히 좌석 수일 수도 있다. 규칙 기반 시스템은 이를 구분하기 어렵지만, BERT 모델은 주변 문맥을 읽어 해당 숫자가 비행기를 의미하는지, 편명을 의미하는지 정확히 파악할 수 있다.
3) 확장성 확보: 모든 예외 케이스를 커버하기 위해 수백 개의 조건문을 만드는 것보다, 모델에게 데이터를 학습시켜 스스로 특징을 찾게 하는 것이 장기적인 유지보수와 확장성 측면에서 유리하다.

따라서 본 프로젝트는 '규칙으로 데이터를 효율적으로 구축하고, 학습된 모델이 규칙이 커버하지 못하는 영역까지 추론'하도록 설계하였다.

2.4. 모델 학습 및 평가

1) 모델 선정: 다국어 및 항공 관련 고유명사 처리에 유리한 bert-base-multilingual-cased를 선정하였다.
2) 학습 진행: 수집된 데이터를 학습 및 검증 세트로 나누어 개체명 인식(NER) 파인 튜닝을 진행하였다.
3) 평가 결과: 학습 결과, 기종(0.99)과 날짜(0.97) 인식에서는 매우 높은 점수(F1-Score)를 달성하였다. 반면, 노선(Route) 정보의 경우 문장 패턴이 매우 비정형적이어서 상대적으로 낮은 인식률(0.23)을 보였다. 이러한 한계점은 후술할 하이브리드 시스템을 통해 기술적으로

보완하였다.

3. 서비스 구조

앞서 수행한 모델 평가 과정에서 확인된 한계점(낮은 노선 인식률, 코드 자체의 난해함 등)을 극복하기 위해, 인공지능 모델과 공공데이터 DB, 규칙 기반 알고리즘을 유기적으로 결합한 하이브리드 시스템을 구축하였다.

3.1. 시스템 처리 흐름

전체 서비스는 사용자의 입력부터 최종 결과물 생성까지 다음과 같은 5단계의 파이프라인을 거쳐 수행된다.

1) **입력**: 사용자가 웹 (Streamlit) 상에서 변환하고자 하는 'Aeroroutes' 기사의 URL 주소를 입력한다.
2) **크롤링**: 시스템이 실시간으로 해당 URL에 접속하여 기사의 제목과 본문 텍스트 데이터를 추출한다.
3) **추론**: 학습된 BERT 모델이 입력된 텍스트를 분석하여 항공사 코드(예: KE), 기종 코드(예: 77W), 공항 코드(예: ICN) 등 핵심 개체명을 1차적으로 추출한다.
4) **데이터 보강**: 딥러닝 모델이 추출한 정보는 여전히 일반인이 이해하기 힘든 코드 형태이다. 이를 해결하기 위해 다음과 같은 이중 보완책을 적용하였다.
   A. DB 매핑: 국토교통부의 항공사·공항·기종 공공데이터(CSV)를 시스템에 연동하였다. 이를 통해 추출된 영문 코드 를 '대한항공', '보잉 777-300ER', '인천국제공항'과 같은 정식 한글 명칭으로 즉시 변환한다.
   B. 규칙 적용: AI 모델이 놓치기 쉬운 복잡한 요일 패턴(예: x135)이나 주간 운항 횟수 정보는 정규표현식 알고리즘이 2차적으로 스캔하여 누락된 정보를 보완한다.
5) **생성**: 위 과정을 거쳐 완벽하게 정제된 정보들을 사전에 정의된 인스타그램 포맷 템플릿에 조립하여, 이모지가 포함된 최종 캡션 텍스트를 출력한다.

3.2. 구현 환경

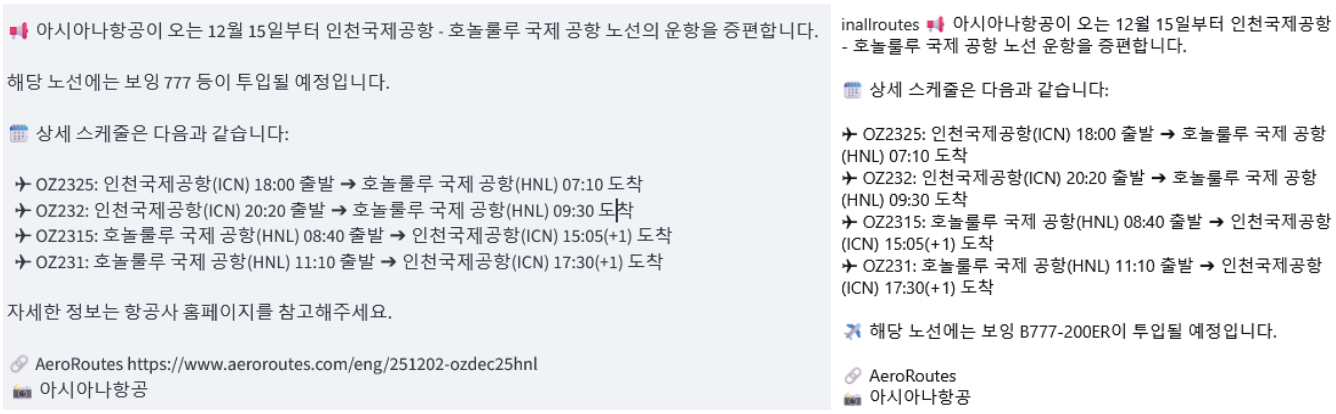
1) 프론트엔드: Python 기반의 웹 프레임워크인 Streamlit을 사용하여, 복잡한 설치 과정 없이 웹상에서 즉시 실행 가능한 사용자 인터페이스(UI)를 구현하였다.
2) 백엔드: Hugging Face Transformers 라이브러리를 활용하여 경량화된 모델 추론 엔진을 구축하였다.
3) 데이터: Pandas 라이브러리를 활용하여 CSV 형태의 대용량 공공데이터를 고속으로 조회하고 매핑하는 로직을 최적화하였다.

4. 실제 사용 결과

개발된 도구의 효용성을 검증하기 위해, 실제 'Aeroroutes'에 게재된 최신 항공 뉴스 5건을 무작위로 선정하여 테스트를 수행하였다. 앞서 언급한 웹사이트(inallroutes.streamlit.app)을 통해 생성하였으며, 각 사례별 입력 데이터와 시스템이 생성한 결과는 다음과 같다.

4.1. [Case 1] 아시아나항공 호놀룰루 노선 증편

- 입력 기사: Asiana Airlines Extra Honolulu Flight in mid-Dec 2025

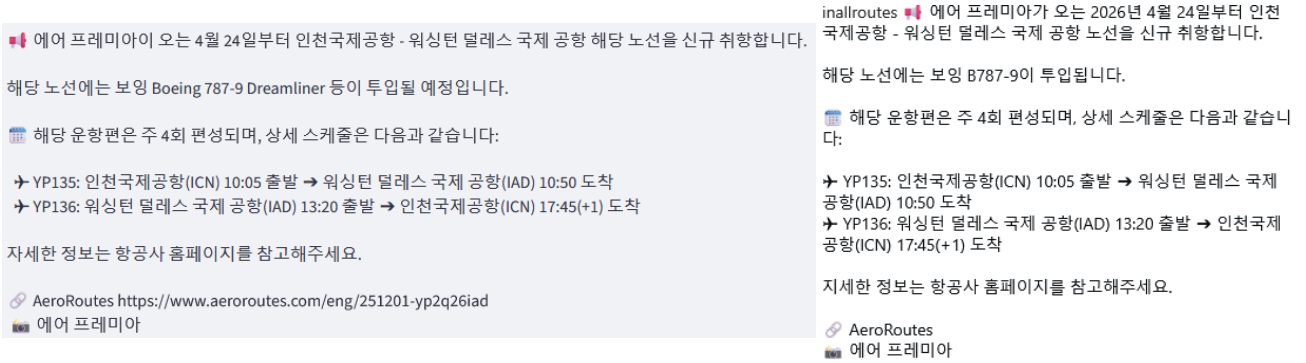


생성 결과(왼) / 실제 업로드 캡션(오)

1) ML 기반 엔티티 추출: BERT 모델이 비정형 영어 본문을 분석하여, 'Asiana Airlines'를 항공사로, 'Boeing 777-200ER'을 기종으로 정확히 태깅하였다. 이는 규칙 기반으로는 식별하기 어려운 문장 속 핵심 정보를 1차적으로 선별해 냈음을 의미한다.
2) 데이터 보강: ML이 찾아낸 텍스트(Asiana Airlines)를 키(Key)값으로 하여 공공데이터 DB를 조회, '아시아나항공'이라는 정식 한글 명칭으로 변환하였다.
3) 정확한 문맥 파악: 제목의 'Extra' 키워드를 통해 '증편' 이슈임을 파악하였다.

4.2. [Case 2] 에어프레미아 워싱턴 D.C. 노선 신규 취항

- 입력 기사: Air Premia Plans Seoul – Washington 2Q26 Launch



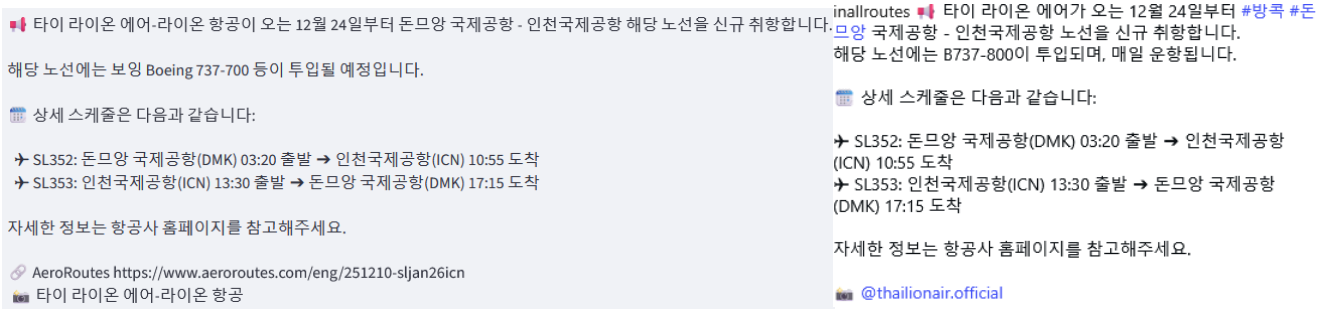
1) 엔티티 추출: BERT 모델이 비정형 텍스트 속에서 'Air Premia'를 항공사로, '789'를 단순

숫자가 아닌 기종 코드로 식별하였다. 또한 24APR26이라는 날짜 포맷을 인식하여 '4월 24일'로 정확히 변환하였다.

2) 데이터 보강: AI가 추출한 IAD 코드를 키(Key)값으로 DB를 조회하여, 단순 워싱턴이 아닌 '워싱턴 덜레스 국제공항'이라는 정식 명칭으로 변환하였다. 이는 서부의 워싱턴 주(State)와 수도인 워싱턴 D.C.를 명확히 구분하여 정확한 공항명을 출력한 사례이다.
3) 규칙 기반 스케줄 분석: 본문의 4 weekly 키워드와 스케줄표의 x246(화/목/토 제외) 패턴을 정규표현식 알고리즘이 분석하여, '주 4회 편성'이라는 정확한 운항 빈도 정보를 도출해 냈다. ML이 놓칠 수 있는 수치적 패턴을 규칙이 완벽하게 보완하였다.
4) 정확한 문맥 파악: 제목의 'Launch' 키워드를 통해 단순 운항 재개가 아닌 '신규 취항' 이슈임을 파악하였다.

4.3. [Case 3] 타이 라이온 에어 방콕(돈므앙) 신규 취항

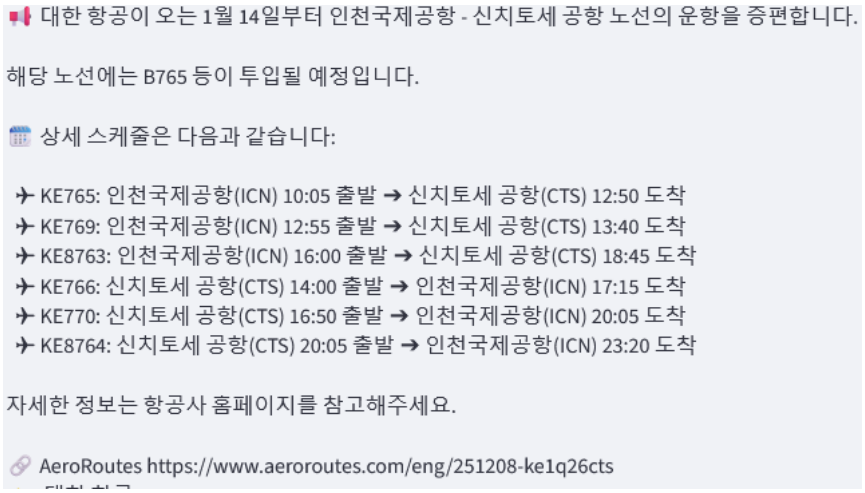
- 입력 기사: Thai Lion Air Moves Bangkok – Seoul Launch to Jan 2026

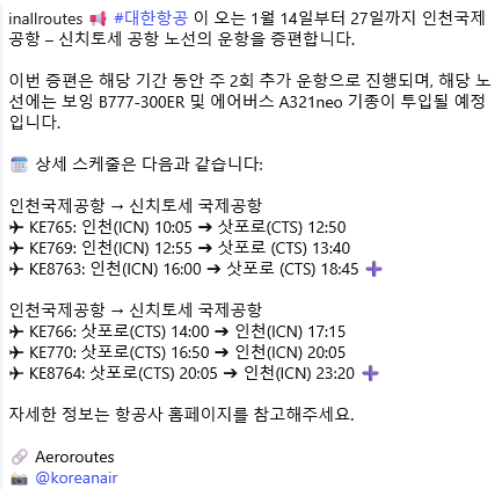


1) 정밀한 공항 코드 매핑: 기사 내 DMK 코드를 식별하고 국토교통부 DB와 대조하여, 단순 '방콕'이 아닌 '돈므앙 국제공항'이라는 구체적인 공항명으로 정확히 변환하였다. 이는 방콕의 수완나품(BKK) 공항과 혼동하지 않고 정확한 정보를 제공한 사례이다.
2) 스케줄 파싱 자동화: SL352 DMK0320 – 1055ICN와 같이 붙여 쓰인 텍스트를 정규표현식 알고리즘이 분해하여 편명, 출발지/시간, 도착지/시간으로 완벽하게 구조화 하였다.
3) 문맥 파악 및 한계 보완: AI 모델은 738 코드를 인식하여 'Boeing 737' 계열임을 파악했으나, 세부 모델명(700/800) 매핑 과정에서 일부 부정확함(700으로 출력)이 있어 실제 업로드 시 'B737-800'으로 보정하였다.

4.4. [Case 4: 수정 사례] 대한항공 삿포로(신치토세) 증편

- 입력 기사: Korean Air 1Q26 Extra Sapporo Flights

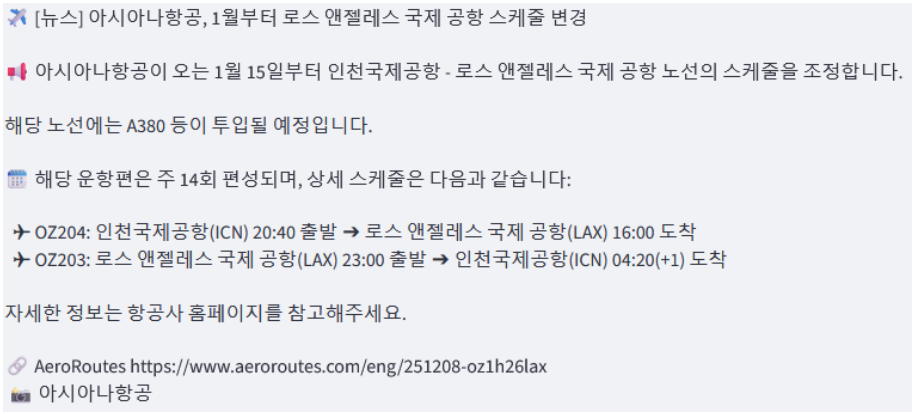


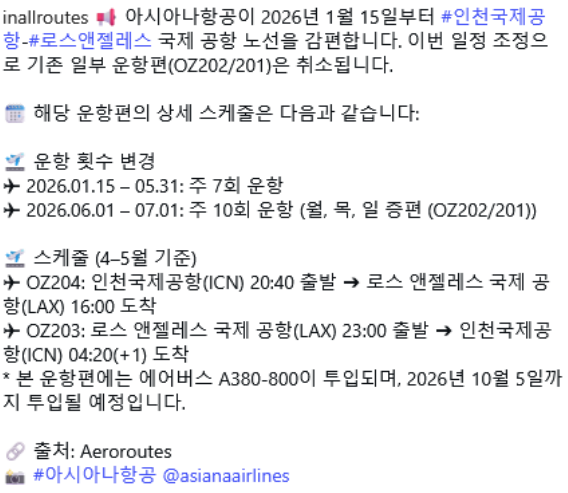


1) 정확한 노선 및 의도 파악: Sapporo New Chitose를 '신치토세 공항'으로 정확히 매핑하였고, 제목의 'Extra' 키워드를 통해 '증편' 의도를 올바르게 파악하였다.
2) 편명과 기종의 혼동: 생성된 캡션에 'B765'라는 존재하지 않는 기종이 출력되었다. 이는 편명인 KE765의 숫자 패턴을 기종 코드로 오인한 것으로 판단된다. 실제 기사 내의 77W, 32Q 코드는 스케줄표에만 존재하고 본문 요약에는 반영되지 않아, 사용자가 업로드 시 '보잉 777-300ER 및 에어버스 A321neo'로 수동 보정하였다.
3) 인간 참여 프로세스: AI가 초안을 작성하고 사용자가 오류(기종, 세부 날짜)를 수정하여 완성하는 협업의 필요성을 보여주는 사례이다.

4.5. [Case 5: 실패 사례] 아시아나항공 LA 노선 감편

- 입력 기사: Asiana Airlines 1H26 Los Angeles Service Reductions





- 원인 분석:

1) 인과관계 파악 실패: BERT 모델은 텍스트 내에서 '14 weekly'라는 수치 정보를 정확히 탐지 해냈다. 그러나 이것이 '현재 상태(From)'인지 '변경 후 상태(To)'인지를 구별하는

관계 추출 능력은 학습되지 않았기에, 가장 눈에 띄는 숫자를 최종 결과로 출력하는 오류를 범했다.

2) 조건부 정보 처리 미숙**:** 기간별로 운항 횟수가 변동(1~5월: 7회, 6월~7월: 10회)되는 복잡한 시계열 정보를 처리하지 못하고, 기사 하단에 나열된 고정 스케줄(OZ204/203)만 단순 추출하여 정보를 과도하게 단순화하였다.
3) 부정적 문맥 인식 부재: 제목의 'Reductions'나 본문의 'Cancelled' 같은 부정적 키워드를 반영하지 못하고, 일반적인 스케줄 변경 템플릿을 적용하여 정보의 중요도(감편)를 전달하지 못했다.

※ 개선 시사점: 본 사례는 BERT 기반 NER 모델이 '무엇'을 찾는 데는 탁월하지만, 단어 간의 관계나 문장의 의도를 파악하는 데는 구조적 한계가 있음을 시사한다. 이를 극복하기 위해 향후에는 엔티티 간의 연결 고리를 학습시키는 관계 추출(RE) 모델을 추가하거나, 문장 전체의 의미를 해석하는 LLM(거대언어모델)을 파이프라인에 통합해야 함을 확인하였다.

5. **결론 및 고찰** (Conclusion & Discussion)

5.1. 프로젝트 성과 및 의의

본 프로젝트는 항공 데이터 처리의 비효율성을 개선하기 위해 시작되었으며, 데이터 수집부터 서비스 배포까지 전체 과정을 수행하며 다음과 같은 성과를 얻었다.

1) 실질적인 효율 개선: 기존에 수작업으로 번역하고 포맷을 맞추느라 건당 30분 가량 걸리던 작업을 10초 이내로 단축시켰다. 단순 프로젝트를 넘어, 실제 내 인스타그램 운영에 쓸 수 있는 도구를 만들었다는 점에서 의미가 있다.

2) 현실적인 문제 및 해결: 처음에는 AI 모델 하나면 다 될 줄 알았으나, 모델이 생각보다 완벽하지 않다는 것(할루시네이션, 패턴 누락 등)을 깨달았다. 이를 해결하기 위해 AI가 못하는 건 코딩(정규표현식)과 데이터베이스로 해결해 보자는 전략을 세웠고, 결과적으로 부족한 모델 성능을 시스템 구조로 보완해 냈다.

3) 서비스 구현 경험: 단순히 Colab에서 모델 성능(F1-Score)만 찍어보고 끝낸 게 아니라 Streamlit을 통해 편하게 접근할 수 있는 웹 서비스 형태로 만들어보면서 실제로 서비스를 구현해 볼 수 있는 경험을 해보았다.

5.2. 기술적 한계

실제 사용 과정에서 식별된 시스템의 구조적 한계는 다음과 같다.

1) 문맥 및 관계 추출의 부재: 엄밀하게 말해, 현재 모델은 문장을 이해한다기 보다는, 학습된 대로 주요 개체에 형광펜을 칠하는 수준에 불과했다. 앞선 [Case 5]의 실패 사례에서 확인했듯, 개체명(Entity) 탐지에는 탁월하나, 수치 데이터 전후의 전치사(From/To)나 문장 전체의 부정적 뉘앙스(Reduction/Cancel)를 파악하는 관계 추출 능력은 부족하였다.

2) 조금만 바뀌어도 고장 나는 규칙의 취약성: 모델이 놓친 정보를 정규표현식(Regex)으로 보완

했는데, 이 코드가 너무 복잡해져 버렸다. 기사 형식이 조금만 달라지거나 규칙이 바로 깨져버려 유연성이 매우 떨어졌다.

3) **1기사 1노선의 단순함:** 실제 기사 중에는 "캐세이퍼시픽항공 9월 스케줄 변경 안내"처럼 한 페이지에 수십 개의 노선 정보가 있는 경우도 빈번하게 나타났다. 현재 내 시스템은 가장 위에 있는 노선 하나만 잡고 나머지는 다 버리는 치명적인 구조적인 문제가 있었다.

5.3. 향후 발전 방향

위 한계점을 극복하고 시스템을 고도화하기 위해 다음과 같은 로드맵을 구상하여 추후 개선해보고자 한다.

1) 기사 분류 및 요약 모델 도입: NER 수행 전, 기사의 유형(신규/증편/감편/단항)을 먼저 분류하는 모델을 추가하고, 복잡한 문맥 처리를 위해 LLM(거대언어모델)을 연동하여 문장 해석 능력을 강화한다.

2) 동적 데이터 파이프라인 구축: 정적인 CSV 파일 대신, FlightRadar24 등의 항공 API와 연동하여 DB가 실시간으로 최신화되는 동적 시스템으로 전환한다.

3) 멀티모달(Multimodal) 확장: 현재의 텍스트 캡션 생성을 넘어, 기사 내용에 부합하는 썸네일 이미지를 생성형 AI로 자동 제작하는 기능까지 확장하여 진정한 의미의 '콘텐츠 자동 생성기'로 발전시킨다.