

Dify로 대학입시 로컬 RAG 상담 시스템 시작하기

| 작성: 2026-09-07
| 대상: Ollama와 로컬 문서로 대학입시 상담 시스템을 만들려는 경우

대학입시 상담 시스템은 단순한 채팅 앱보다 RAG(검색 증강 생성), 상담 워크플로, 구조화된 입시 데이터, 계산 규칙을 함께 설계해야 안정적으로 운영할 수 있습니다. 첫 단계는 Dify와 Ollama를 연결해 문서 근거가 있는 상담 MVP를 만드는 것이 적합합니다.

결론

우선은 Dify + Ollama로 시작하는 것을 권장합니다.

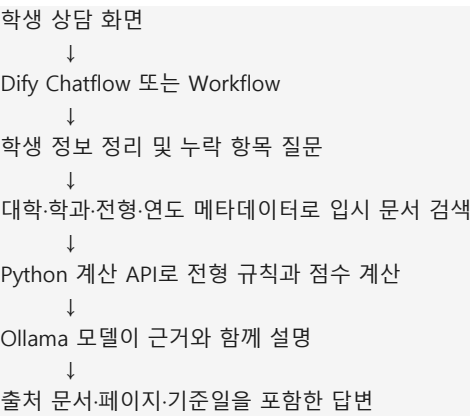
- Dify: 지식베이스, 검색, 상담 단계, 변수, API와 웹 채팅을 구성
- Ollama: 인터넷 없이 로컬에서 언어 모델 실행
- 입시 문서: 모집요강, 전형 안내, 입시 결과를 근거로 제공
- 별도 계산 API: 반영 비율, 가산점, 지원 자격과 같은 규칙 계산

Dify는 검색 결과를 워크플로의 다음 단계로 전달하고, 문서 메타데이터를 활용할 수 있습니다. 따라서 대학·학과·전형·연도별 검색과 출처 표시를 구성하기 좋습니다. 복잡한 상담 흐름이 필요해지면 Flowise나 별도의 Python API를 추가할 수 있습니다.

도구별 선택

도구	적합한 경우	특징
Dify + Ollama	실제 상담 서비스의 첫 버전	지식 검색, 워크플로, 변수, API, 웹 채팅을 시각적으로 구성
Flowise + Ollama	검색과 에이전트를 세밀하게 제어	재검색, 재랭킹, 에이전트, 외부 API와 MCP를 노드로 구성
RAGFlow + Ollama	복잡한 PDF와 표가 핵심	PDF, DOCX, 표, 이미지의 문서 구조 분석에 집중
AnythingLLM + Ollama	빠르게 문서 Q&A를 시험	Workspace와 문서 업로드가 쉽지만 복잡한 상담 흐름은 제한적

권장 전체 구조



입시 지식베이스 구성

처음부터 모든 자료를 넣기보다 최근 연도 자료로 작은 검증 세트를 만드는 것이 좋습니다.

문서 종류

- 대학별 모집요강 원문
- 전형별 지원 자격과 제출 서류
- 모집 인원과 전형 일정
- 학생부·수능·면접·논술 등 반영 방법
- 대학·학과·전형별 경쟁률
- 전년도 입시 결과와 충원 자료
- 지역, 계열, 고교 유형별 안내 자료

문서 메타데이터

각 문서나 문서 조각에 다음 값을 붙입니다.

```
university: 서울대학교
department: 컴퓨터공학부
admission_type: 학생부종합
year: 2027
region: 서울
source_date: 2026-05-01
source_url: https://...
```

메타데이터가 있으면 "2027학년도 서울 지역 학생부종합 전형"처럼 조건을 좁힌 검색이 가능하고, 오래된 모집요강이 새 자료에 섞이는 문제를 줄일 수 있습니다.

상담 워크플로

첫 번째 MVP는 다음 순서로 구성합니다.

1. 학생의 학년, 계열, 내신, 모의고사, 희망 지역과 대학을 입력받습니다.
2. 비어 있는 핵심 정보가 있으면 먼저 추가 질문을 합니다.
3. 연도와 전형 조건에 맞는 모집요강을 검색합니다.
4. 지원 자격과 전형 일정을 확인합니다.
5. 가능한 전형을 안정·적정·도전 그룹으로 나눕니다.
6. 각 판단에 문서명과 페이지 또는 출처 URL을 붙입니다.
7. 자료 기준일과 추가로 확인해야 할 사항을 표시합니다.

상담 프롬프트에는 다음 규칙을 넣는 것이 좋습니다.

- 검색된 문서에 없는 사실은 단정하지 않는다.
- 대학, 전형, 연도를 답변에 반드시 표시한다.
- 숫자 계산은 계산 API 결과를 우선한다.
- 근거 문서와 페이지 또는 URL을 함께 표시한다.
- 자료 기준일 이후 변경 가능성이 있으면 확인이 필요하다고 표시한다.
- 합격을 보장하는 표현을 사용하지 않는다.

계산은 LLM과 분리

반영 비율, 가산점, 지원 자격, 모집 인원, 환산 점수는 LLM에게 계산만 시키지 않는 것이 좋습니다. 대학별 규칙을 데이터베이스나 Python 함수로 관리하고, Dify가 API를 호출한 뒤 그 결과를 모델이 설명하도록 구성합니다.

예를 들어 다음 값은 계산 API가 처리합니다.

```
학생부 교과 환산점수
수능 영역별 반영 점수
탐구 과목 선택과 가산점
지원 자격 충족 여부
전형별 모집 인원과 경쟁률 비교
```

LLM은 계산 결과를 자연어로 설명하고, 해당 결과가 사용된 모집요강을 인용하는 역할을 맡습니다.

Windows에서 시작하는 순서

현재 PC에서 시작할 때는 Docker Desktop과 Ollama를 준비합니다.

```
# Ollama 설치 후 모델 준비
ollama serve
ollama pull qwen2.5      # 상담용 LLM (한국어 지원, 예시)
ollama pull bge-m3       # 임베딩 모델 (다국어, RAG 검색용)
```

LLM(상담 답변)과 임베딩(문서 검색) 모델은 역할이 다르므로 각각 등록해야 합니다. 한국어 상담 품질이 중요하면 qwen2.5, exaone 계열 등을 비교해 고르고, 임베딩은 다국어에 강한 bge-m3를 권장합니다.

Dify는 공식 Docker Compose 구성을 사용해 실행하고, Dify의 모델 설정에서 Ollama 엔드포인트를 연결합니다. 이 PC는 Windows Docker Desktop이므로, Dify 컨테이너에서 호스트의 Ollama에 접근할 때는 `http://host.docker.internal:11434`를 엔드포인트로 사용합니다. (Linux에서 host 네트워크로 실행하는 경우에는 `http://172.17.0.1:11434` 등 Docker 브리지 주소를 사용합니다.)

설치가 끝나면 다음 순서로 검증합니다.

1. Ollama API가 응답하는지 확인합니다.
2. Dify에서 LLM 모델과 임베딩 모델을 등록합니다.
3. 모집요강 PDF 한 개를 지식베이스에 넣습니다.
4. 문서에 실제로 있는 질문과 없는 질문을 각각 테스트합니다.
5. 답변에 출처가 표시되는지 확인합니다.

품질 검증 기준

상담 시스템을 공개하기 전에 질문과 기대 답변을 고정한 평가 세트를 만듭니다.

- 지원 자격을 정확히 판단하는가
- 연도와 전형을 혼동하지 않는가
- 표의 숫자를 잘못 읽지 않는가
- 근거 문서와 페이지를 표시하는가
- 자료에 없는 질문에 모른다고 답하는가
- 계산 API 결과와 설명이 일치하는가

모집요강이 개정되면 기존 문서를 삭제하거나 비활성화하고, 문서 기준일과 버전을 갱신해야 합니다.

단계별 확장

1단계: 문서 상담 MVP

Dify와 Ollama를 연결하고 최근 모집요강 10~20개로 출처가 포함된 Q&A를 검증합니다.

2단계: 전형 계산

Python API를 연결해 대학별 환산 점수와 지원 자격을 계산합니다.

3단계: 맞춤 상담

학생 프로필, 희망 대학, 상담 이력을 저장하고 안정·적정·도전 지원 전략을 생성합니다.

4단계: 운영 서비스

로그인, 권한, 관리자용 문서 갱신, 평가 대시보드, 최신 자료 알림을 추가합니다.

참고 문서

- Dify 지식 검색 테스트
- Dify 공식 사이트
- Flowise 공식 문서
- RAGFlow 공식 문서
- AnythingLLM 공식 문서

추천 결론

빠른 검증은 AnythingLLM도 가능하지만, 대학입시 전문 상담처럼 질문 순서, 검색 조건, 계산 API, 출력 표시를 계속 확장할 계획이라면 Dify + Ollama로 시작하는 것이 가장 균형이 좋습니다. 검색 흐름을 완전히 코드 수준으로 제어해야 할 때 Flowise를 추가하고, 복잡한 PDF 표 분석이 병목이 될 때 RAGFlow를 검토합니다.