강의 한 편에 일주일, 지금은 20 분

자막을 '받아쓰기'가 아니라 '정렬'로 만들어 강의 57 편을 제작한 기록

- 응모 분야: 업무 생산성 개선을 위한 AI 활용
- 제안자: 유예찬 ((주)스마트포크 · AI 교육 콘텐츠 제작)

---

① 어떤 상황에서 AI 를 활용했나

AI 교육 영상을 만드는 일을 합니다. 대본을 쓰고, 음성을 입히고, 화면을 붙이고, 자막을 답니다. 이 중에서 가장 오래 걸리는 건 촬영이 아니라 **1 차 편집**이었습니다. 그리고 1 차 편집에서 가장 오래 걸리는 건 자막이었습니다.

이유는 한국어 기술 강의의 문장이 이렇게 생겼기 때문입니다.

> *"Claude Code 를 실행하면 CLI 가 뜨고, 거기서 MCP 서버를 붙일 수 있습니다."*

한 문장에 한글과 영어 고유명사가 섞여 있습니다. 자동 자막(STT, 받아쓰기)을 돌리면 여기서 반드시 깨집니다. Claude Code 가 "클로드 코드"로 적히거나 "크라우드 코드"가 되고, CLI 는 "씨엘아이"가 됩니다. 화면에 나가면 안 되는 글자들입니다. 결국 자막 파일을 열어 전수 수정해야 했고, 강의 한 편에 자막 줄이 수백 개입니다.

무음 제거도 같은 문제였습니다. 강의 음성에는 문장과 문장 사이 빈 구간이 계속 생깁니다. 그대로 두면 늘어지고, 일괄로 잘라내면 말이 붙어 버립니다. 마침표 뒤의 쉼과 쉼표 뒤의 쉼은 남겨야 할 길이가 다른데, 편집 프로그램의 자동 무음 제거는 그 구분을 하지 못합니다. 사람이 한 컷씩 판단해야 했습니다.

강의를 한두 편 만들 때는 견딜 만했습니다. 시리즈가 쌓이면서 이 작업이 병목이 됐습니다.

② 어떤 AI 를, 어떤 방식으로 썼나

발상의 전환: 받아쓰지 말고 맞춰라

핵심은 도구를 바꾼 게 아니라 **문제를 다시 정의한 것**입니다.

자동 자막이 틀리는 이유는 AI 가 "무슨 말인지 알아맞히려" 하기 때문입니다. 그런데 저는 이미 **대본을 가지고 있습니다.** 알아맞힐 필요가 없습니다. 필요한 건 "이 글자가 몇 초에 나오는가"뿐입니다.

그래서 받아쓰기(STT) 대신 **강제 정렬(forced alignment)**을 씁니다. 음성과 대본을 함께 넣고, 각 글자가 음성의 몇 초 지점에 해당하는지만 계산합니다. 화면에 찍히는 글자는 **언제나 제 원본 대본 그대로**입니다. 받아쓰기 오류가 원천적으로 발생할 수 없는 구조입니다.

그래도 남은 문제, 그리고 음차본

정렬 엔진도 결국 음성을 듣습니다. 음성은 "클로드 코드"라고 발음하는데 대본에는 Claude Code 라고 적혀 있으니, 둘을 짝지을 때 헷갈립니다.

그래서 파일을 **두 벌** 만듭니다.

파일	내용	쓰임
<강의>.align.txt	**음차본** — "클로드 코드를 실행하면 씨엘아이가 뜨고"	정렬(음성과 맞추기)에만 사용
<강의>.txt	**표기본** — "Claude Code 를 실행하면 CLI 가 뜨고"	화면 자막 글자로 사용

두 파일은 문장과 토큰 순서가 1:1 로 대응합니다. **소리는 음차본에 맞추고, 글자는 표기본에서 가져옵니다.** 음차본에서 표기본을 만드는 변환 사전도 코드에 두어, 새 용어가 나오면 한 줄 추가하면 됩니다 (("이천이십육년", "2026 년")).

사용한 도구

용도	도구	비고

강제 정렬	stable-ts (Whisper large-v3), WhisperX 대안	오픈소스
무음 감지	FFmpeg silencedetect	오픈소스
타임라인 생성	OpenTimelineIO (FCP7 xmeml)	오픈소스
음성 합성	ElevenLabs / Google Gemini TTS	강사 본인 음성
아바타 영상	HeyGen	유료 라이선스
파이프라인 설계·구현	**Claude Code**	아래 서술

마지막 줄이 중요합니다. 이 자동화 자체를 **AI 로 만들었습니다.** 먼저 "무엇을 만들 것인가"를 명세서(SPEC.md)로 쓰고, Claude Code 에게 그 명세를 주어 구현하게 했습니다. 정렬 엔진 선택, 캐시 구조, 컷 판단 규칙을 대화하며 다듬었습니다. AI 를 **결과물 생산에만 쓴 게 아니라 생산 설비를 만드는 데** 썼습니다.

6 단계 파이프라인

```
1 무음감지  2 정렬  3 컷플랜  4 자막+검수  5 렌더  (6) Premiere XML
```

- 정렬은 **한 번만** 수행하고 결과를 캐싱합니다. 이 한 번의 결과가 자막(4)과 컷 판단(3)을 **둘 다** 구동합니다.
- 컷 규칙은 문장부호 맥락별로 다릅니다. 마침표 뒤와 쉼표 뒤에 남기는 길이가 다르게 설정됩니다.
- 모든 임계값은 설정 파일(config.yaml)에 있습니다. **하드코딩 금지**를 설계 원칙으로 명문화했습니다.
- 컷 지점마다 미세 페이드를 넣어 이음매의 클릭·팝 잡음을 없앱니다.
- 출력은 렌더링해서 굳힌 파일이 아니라 **편집 가능한 타임라인(XML)**으로도 낼 수 있습니다. 자동화 결과를 사람이 이어서 손볼 수 있어야 하기 때문입니다.

③ 무엇이 달라졌나

강의 한 편이 나오기까지

	자동화 전	자동화 후
강의 한 편 제작	**일주일에 한 편 나올까 말까**	**20 분**

제작 주기의 단위가 '주'에서 '분'으로 바뀌었습니다.

이 차이가 실제로 무엇을 의미하는지는 누적 수치로 드러납니다. 지금까지 이 파이프라인으로 만든 강의가 57 편입니다. 예전 속도였다면 57 주, 1 년이 넘게 걸렸을 분량입니다.

그리고 이건 단순히 "빨라졌다"가 아닙니다. 주 1 편이 한계일 때는 **만들 수 있는 강의만 만들게 됩니다.** 시리즈를 기획해도 완주할 수 있을지 알 수 없으니 애초에 짧게 잡습니다. 제작 시간이 20 분이 되자 25 강짜리 시리즈를 계획하고 끝내는 일이 가능해졌습니다. 속도가 바뀌니 만들 수 있는 것의 종류가 바뀌었습니다.

실제 산출 (2026 년 9 월 기준)

항목	수치
이 파이프라인으로 제작한 강의	**57 편**
생성된 완성 산출물	83 개
생성된 자막(SRT)	97 개
시리즈	원론 25 강, 총론 7 강, 웹소설 5 강, 클로드코드 계열, 중급·고급·오픈클로 등

질적 변화

자막 수정이 사라졌습니다. 화면 글자가 언제나 원본 대본이므로, 받아쓰기 오타를 고치는 작업 자체가 없어졌습니다. 사람이 확인해야 하는 건 _review.md 에 따로 모이는 **정렬 신뢰도가 낮은 구간뿐**입니다. 전수 검사가 표적 검사로 바뀌었습니다.

작업의 성격이 바뀌었습니다. 전에는 "자막을 만드는" 일이었고 지금은 "규칙을 조정하는" 일입니다. 컷이 너무 붙었다 싶으면 설정값 한 줄을 바꿔 3~5 단계만 다시 돌립니다. 정렬은 캐시가 있어 다시 하지 않습니다.

실패가 재현 가능해졌습니다. 예전에는 편집이 어긋나면 어디서 틀어졌는지 알 수 없었습니다. 지금은 단계별 중간 결과가 JSON 으로 남아, 정렬이 문제인지 컷 규칙이 문제인지 바로 가릅니다.

④ 무엇이 독창적인가

1. **받아쓰기가 아니라 정렬.** 자동 자막의 정확도를 높이려 한 게 아니라, 정확도가 필요 없는 구조로 바꿨습니다. 원본 대본이 있는 상황에서는 AI 에게 "알아맞히기"를 시킬 이유가 없습니다.
2. **음차/표기 분리.** 한국어 기술 강의의 한·영 혼용은 흔한 문제인데, 대개 후처리(치환 사전)로 해결하려 합니다. 후처리는 새 용어마다 깨집니다. 저는 정렬용 텍스트와 표기용 텍스트를 **처음부터 두 벌로 분리**해 문제가 생기지 않게 했습니다.
3. **한 번의 정렬로 두 가지를 구동.** 자막과 컷 판단은 보통 별개 작업입니다. 같은 정렬 결과를 공유하게 만들어, 자막과 영상의 싱크가 구조적으로 어긋날 수 없게 했습니다.
4. **문장부호 맥락별 차등 컷.** 일괄 무음 제거와 다릅니다. 마침표 뒤의 쉼은 호흡이고 쉼표 뒤의 쉼은 군더더기라는 판단을 규칙으로 옮겼습니다.
5. **평탄화하지 않는 출력.** 자동화의 결과를 최종본으로 굳히지 않고 편집 가능한 타임라인으로 내보냅니다. 자동화가 사람의 판단을 대체하는 게 아니라 사람이 판단할 지점까지 데려다주는 구조입니다.
6. **유료 API 의존 금지를 설계 원칙으로.** 핵심 공정(정렬·무음·타임라인)은 전부 오픈소스입니다. 유료 도구는 음성·아바타 등 대체 가능한 부분에만 씁니다. 남이 따라 하려 할 때 비용이 장벽이 되지 않도록 한 선택입니다.

⑤ 다른 사람이 따라 할 수 있나

이 항목이 이 사례의 핵심이라고 생각합니다. **"해봤다"가 아니라 "넘겨줄 수 있다"**를 목표로 만들었습니다.

문서 3 종

문서	역할
SPEC.md	무엇을 만들 것인가 — 설계 원칙, 입출력 계약, 기술 스택 선택 이유
README.md	어떻게 쓰는가 — 설치, 입력 배치, 실행 명령, 권장 첫 실행 순서
CLAUDE.md	작업 런북 — 강의 한 편을 만드는 전체 절차와 분기

따라 하기 쉽게 만든 장치

- **단계별 독립 실행.** --stage 1-2, --stage 3-5 처럼 필요한 구간만 돌립니다. 전체를 이해하지 못해도 한 단계씩 검증하며 들어올 수 있습니다.
- **중간 캐시.** 무거운 정렬을 반복하지 않습니다.
- **위험 구간 우선 검증 순서를 문서화.** README 에 "가장 먼저 1-2 단계만 돌려 정렬 정확도부터 판정하라"고 적어 두었습니다. 전체 파이프라인에서 진짜 위험한 곳이 어디인지 미리 알려 주는 것입니다.
- **실패 모드까지 문서화.** 10 분 넘는 음성은 후반부 싱크가 밀린다 파트별 정렬로 우회. GPU 메모리가 부족하면 5 분 단위 자동 분할. 겪은 실패와 그 우회법을 그대로 적어 두었습니다.
- **설정으로 분리.** 모든 임계값이 설정 파일에 있어, 코드를 읽지 않고도 자기 강의 스타일에 맞출 수 있습니다.
- **확장을 전제로 설계.** 런북 첫머리에 "팀원은 이 자동화를 베이스로 자기 강의 스타일에 맞게 확장한다"고 적혀 있습니다.

실제로 같은 방식으로 만든 다른 자동화(전자책 제작 툴킷)를 동료 배포용 독립 저장소로 분리해 넘긴 전례가 있습니다. 이 파이프라인도 같은 경로로 넘길 수 있습니다.

⑥ 책임 있는 활용을 위해 지킨 것

- **사용 도구와 라이선스를 전부 명시했습니다.** 오픈소스(FFmpeg, stable-ts/Whisper, OpenTimelineIO)와 유료 도구(ElevenLabs, HeyGen)를 구분해 표기했습니다.
- **음성과 초상.** 합성 음성은 강사 본인의 음성을 본인 동의 하에 사용했고, 아바타는 유료 라이선스 범위 안에서 사용했습니다. 타인의 음성·초상을 무단 합성한 부분이 없습니다.
- **원고 저작권.** 강의 대본은 전부 본인이 작성한 것입니다. 자막에 나가는 글자가 원본 대본 그대로이므로 출처가 명확합니다.
- **개인정보.** 파이프라인이 다루는 데이터는 본인의 강의 음성과 대본뿐이며, 제 3 자의 개인정보가 포함되지 않습니다.
- **사실 확인.** 강의 내용 중 사실 주장은 별도 검증을 거칩니다. AI 가 생성한 문장을 검증 없이 강의에 넣지 않는 것을 작업 규칙으로 두고 있습니다.
- **AI 의 역할을 감추지 않습니다.** 이 파이프라인은 사람의 검수를 없애는 것이 목적이 아닙니다. 검수해야 할 지점을 좁혀 주는 것이 목적이고, 그래서 출력이 편집 가능한 형태로 나옵니다.

맺으며

이 사례에서 실제로 바뀐 것은 도구가 아니라 질문이었습니다.

"AI 자막의 정확도를 어떻게 올릴까"를 붙들고 있는 동안에는 답이 없었습니다. **"나는 이미 대본을 가지고 있는데 왜 AI 에게 알아맞히라고 하고 있지?"**로 질문을 바꾸자 문제가 사라졌습니다.

AI 를 잘 쓰는 일은 더 좋은 모델을 찾는 일이 아니라, AI 에게 시키지 않아도 되는 일을 알아보는 일에 가깝다고 생각합니다.