A translation pipeline for feature-film screenplays

Jake Tae-moon Jo · Engineering lead · https://jake-jo-portfolio.vercel.app/en/work/screenplay-translation/

Designed a screenplay-translation pipeline in which 11 AI agent roles translate and judge, and translated four feature scripts end to end five times in two directions.

4 films, 5 full runs	One day	83%	0
two directions, end to end	full pipeline, one feature	auditor backtest recall, 3rd attempt	hallucinations, all 5 runs (own gate)

Results

Four feature scripts were translated end to end five times in two directions.

Every change request from the production company was shipped within 24 hours.

Four feature scripts were translated end to end five times in two directions: Vietnamese → Korean four times (two drafts of Work A, Work B, Work C) and Korean → Vietnamese once (Work D). The outputs are an industry-formatted script PDF, an editable DOCX and a report that opens with the list of losses.

The first full feature (95 pages) was finished on the night of the day the empty folder was created. Works C and D each went through the whole pipeline in one day (by file timestamps; Work C includes about an hour stopped at a usage limit). Running several models together would cover this; it is not applied now, to keep costs down.

All five runs had zero hallucinations (by the pipeline's own gates).

The first Korean → Vietnamese script (Work D) ran on the same role definitions, with no new roles. English stopped at a one-sequence trial and the dialogue-only external benchmark; there is no full English translation.

Period

Sep – Oct 2026

Role

Only human on the build team; designed and ran the AI agent team

Team

Me plus 11 AI agent roles

Platforms

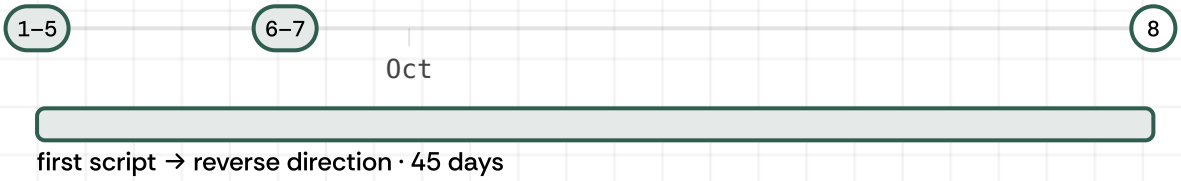
An operator-run harness inside Claude Code (not a web service); outputs a script PDF, a DOCX and a report

Context

A film production company. Work started from the source script, earlier drafts and a human Korean translation

From the first script to the reverse direction

Four feature scripts were translated end to end five times in two directions.



1	Sep 16, 2026	**First commits** Design, 11 agents, orchestrator, intake tools, backtest
2	Sep 16, 2026	**Auditor backtest passed** 10 of 12 mutations (83%), no confirmed false positives, third attempt
3	Sep 16, 2026	**Parallel sections dropped** Sequential relay within a language (my call)
4	Sep 16, 2026	**Work A (draft 6.5), full Korean** 95 pages, the night of the day the empty folder was created
5	Sep 17, 2026	**Work A (draft 7.0), full Korean** 94 pages; steps that found nothing were cut
6	Sep 26, 2026	**Work B, full Korean** 86 pages
7	Oct 1, 2026	**Work C, full Korean** 89 pages, one day
8	Oct 31, 2026	**Work D, full Vietnamese** 103 pages, first Korean → Vietnamese run, one day

Problem and context

The job was to turn "don't translate literally; keep the same context and emotion in the other language" into checks that could be measured.

The client is a film production company. They wanted feature screenplays carried into another language without literal translation, keeping the same context and emotion. A sad scene could not come out less sad, and a joke could not die in a word-for-word rendering.

Three target languages were planned: Vietnamese, Korean and English. Full scripts were translated in two directions, Vietnamese → Korean and Korean → Vietnamese; English stopped at a one-sequence trial (4 scenes) and the dialogue-only external benchmark.

The design started from one observation: without measurement, nobody can tell whether that requirement was met. So before translating, what had to survive (emotional intensity per beat, comic beats, setups and payoffs) was fixed as numbers and lists, and the translation alone was then checked backwards to see whether it survived.

The starting material was Work A's source, a Korean translation made by a person (the reference translation below) and a newer draft not yet translated. The harness had three things to get right: keep the revision marks between drafts so the changed scenes in the next draft can be picked out, flag the loss whenever two distinct source concepts are merged into one, and keep the script format intact.

What was built is not a web service but a translation harness an operator runs inside Claude Code. A web service was deferred until "the translation quality itself is judged excellent".

My role

I designed and ran the 11 agent roles and their gates and made the three calls that changed direction; each film's market, purpose and notation were recorded separately as the production company's decisions.

I was the only human on the build team. I am the only git author, and all 148 commits carry an AI co-author mark. Nothing records which parts I wrote by hand and which the agents wrote, so the role reads "Only human on the build team; designed and ran the AI agent team".

What I designed: 11 agent roles, an orchestrator skill that calls them stage by stage, 9 ground rules and 26 tools (17 for the translation harness, 9 for the external benchmark). The roles sit in three layers: source anatomy (AI director, structure writer, detail writer, forms-of-address keeper), transfer (translator, cultural adviser, performance reviewer) and judging (auditor, format keeper, AI readers, AI investors). Nine run on the Opus family and two on Sonnet. The orchestrator is a skill: the main session runs the stages and calls each agent by naming its definition file.

Call records survive only for sessions from Sep 26 on. The Work C session called agents 33 times and the Work D session 51 times. Of Work D's 51 calls, 47 named one of 10 role definitions (8 each for the AI director, the translator and the auditor, among others) and 4 used no definition. The format keeper has no calls in the surviving records; the format-agent step was cut when the steps were trimmed on Sep 17.

The three decisions below that changed direction or standards are on record as the "user's", and I was the one running those sessions. On Sep 16 I pointed out that the sections translated in parallel contradicted each other and had them redone in sequence. The same day I set the principle that a proven fix is not finished in one script but goes into the whole pipeline. On Sep 17, noting "this is taking far too long", I had the steps that found nothing new removed. Market, purpose and naming for each script are put to a person; anything unanswered uses the previous script's decision as the default and goes to the report's user (production company) confirmation section as pending. The reports list 4 pending items for Work B, 3 for Work C and 8 for Work D.

Human decisions rank above the AI director. The order is "user decision > director's locks > lexicon", and the director never re-judges a

user decision.

From source script to final output

Tools clean the source and agents dissect it into scene contracts; then a fresh translation runner takes each section in turn, and a section must pass its gates before the next one starts.

① **Intake** ☐System
Watermark cleaning, tone-mark check, structure parsing; any failure stops the run

② **Source dissection** ◇Agent
Structure, detail and address-form roles work in parallel: structure map, emotion intensity, lexicon, ledger

③ **Scene contracts and locks** ◇Agent
The AI director merges them and locks the strings every section must write the same way, before translation

④ **Section translation** ◇Agent
A fresh runner per section, given only its own source and packet plus the previous section's final script

⑤ **Section gates** ☐System
Checks for missing scenes, handoff leaks and quotes, hallucinations, and locks

⑥ **Judging after assembly** ◇Agent
The auditor infers the scene contracts from the translation alone and compares them with the source; AI audience and investor simulations

⑦ **Human decisions** ○Person
Market, purpose, naming; undecided items go to the report as awaiting confirmation

⑧ **Final output** ☐System
Script PDF, editable DOCX, and a report with the loss list first

Loops back

↺ ⑤ Section gates → ④ Section translation: if blocked, retranslate the section

Who decided what

People set the direction and the standards, the AI director settled conflicts between agents, and the machine gates blocked through the tools' exit codes.

People

Dropped parallel sections for a sequential relay (Sep 16)
Proven fixes go into the whole pipeline (Sep 16)
Cut the steps that found nothing new (Sep 17)
Market, purpose and naming per script; anything left open goes to the report as pending

11 AI agent roles

Source anatomy 4 · transfer 3 · judging 4 (Opus family 9, Sonnet 2)
The AI director settles agent conflicts and never re-judges a human decision
12 AI readers and 5 AI investor profiles are AI simulations, and the reports label them so
Surviving session logs: 33 calls for Work C, 51 for Work D (10 of the 11 roles)
AI co-author mark on all 148 commits

Machine gates

Stop if the source fails the tone-mark check

Section gate blocks missing scenes, hand-off leaks and quote marks, and is itself tested with planted faults

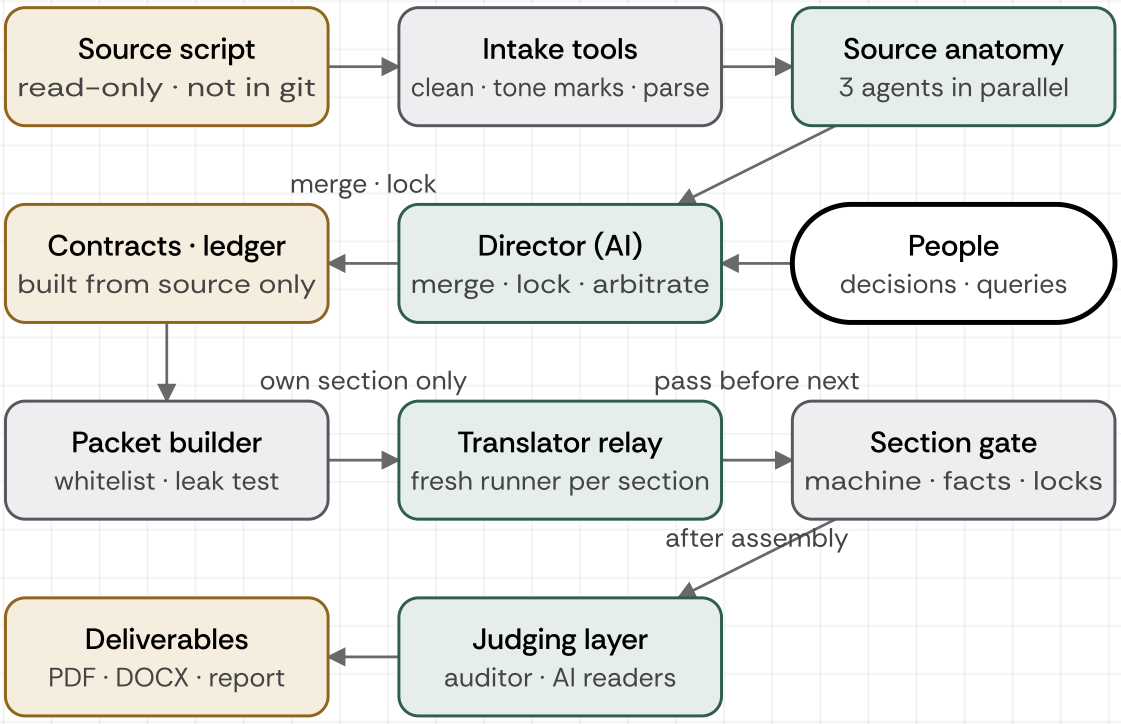
Any failed hallucination check means automatic rejection (no arbitration)

No CI, PRs or hooks

System architecture

Pipeline architecture

Each translator sees only its section's source and packet plus the finished earlier script and hand-off, runs in order; the next one starts only after the gate passes.



Source script

The source script (PDF or DOCX) is read-only and never goes into git.

Intake tools

Tools. Watermark cleanup, a Vietnamese tone-mark integrity check (density, variety, broken characters) and structure parsing. Any failure stops the run. In Work A's source, 194 spots of watermark contamination were found and removed.

Source anatomy

Agents. A structure writer, a detail writer and a forms-of-address keeper dissect the source in parallel: a structure map, emotional intensity (1–10) and comic beats per beat, a lexicon of names and forms of address, and a state ledger.

Contracts · ledger

For each scene, its dramatic function, emotional arc, subtext, setups and payoffs and the lines that must survive are fixed in JSON. Contracts are built from the source only; translation output is never an input. The state ledger (who knows what, and so on) is also built in full from the source up front, and translators only read it.

Director (AI)

An agent (AI). It merges the anatomy into scene contracts, locks the strings every section must render the same way before translation starts, and arbitrates audit and screening results in one pass. It never re-judges a human decision.

People

People set the market, purpose, distribution and naming. Human decisions sit above the AI director's rulings. Source-level problems go to the writer as queries.

Packet builder

Tool. Only whitelisted layers 1–5 go in. "Information not in this packet does not exist. Do not invent it; report it." Whether later plot leaks is tested by having an agent that does not know the answer guess the twist.

Translator relay

Agents. Within a language, sections are translated in order, with a fresh runner for each. A runner gets only its section's source and packet plus the finished script and the hand-off from earlier sections, never the previous runner's notes. Languages run in parallel.

Section gate

Tools plus the auditor. ① Machine gate (blocks missing scenes, hand-off leaks and quote marks) ② hallucination gate (dialogue blocks per speaker, proper nouns, numbers, ledger) ③ director's lock revisions and retroactive replacement. A blocked section is translated again. The gate tests itself by planting faults.

Deliverables

The Fountain script becomes an industry-formatted PDF and an editable DOCX, with a report that opens with the list of losses.

Judging layer

Agents. The auditor reads only the translation, reconstructs the scene contracts, then compares them with the source contracts; an intensity gap of 3 or more is a failure. Four AI readers and five AI investor profiles read only a copy of the translation. This is not audience research. The director rules on the results in one pass, with one correction round.

`backtest/relay_gate.py` · The gate tests itself: it plants quoted dialogue, a deleted scene and an altered locked string in a copy of the script, and if it misses any of them its results are not trusted. (Docstring and names translated to English, print and bookkeeping lines left out, the scene-deletion regex folded into drop_scene)

```
def mutate(group):
    """Plant violations in a copy of the script; all must be caught."""
    real = bp.RUN
    base = (real / f"ko_{group}.fountain").read_text(encoding="utf-8")
    ok0 = gate(group)
    start, end = bp.GROUPS[group]
    mid = (start + end) // 2
    present = [a for a in dict.fromkeys(CHECKED)
               if len(a) >= 4 and base.count(a) == 1]
    cases = {
        "dialogue quotes": base.replace("\n", '\n  "fake line"\n', 1),
        f"scene {mid} deleted": drop_scene(base, mid),
    }
    if present:
        s = present[0]
        cases["locked string altered"] = base.replace(s, s[::-1])
    tmp = pathlib.Path(tempfile.mkdtemp())  # … plus the hand-off file
    misses = 0
    for name, t in cases.items():
        if t == base: misses += 1; continue  # mutation did not apply
        (tmp / f"ko_{group}.fountain").write_text(t, encoding="utf-8")
        bp.RUN = tmp
        with contextlib.redirect_stdout(io.StringIO()) as buf:
            ok = gate(group)
        caught = (not ok) or (WARN in buf.getvalue())
        misses += not caught
    bp.RUN = real
    return ok0 and not misses
```

Key decisions

Two of the six started from my calls, sequential sections and cutting steps that found nothing; four are agent defaults, set by structure and checks.

Feature-length context through closed packets and a ledger built from the source up front

Agent default · no sign-off on record

Problem

Split a feature into scenes and translate them one by one, and the translator fills the missing information by inventing it. Hand over the whole script, and it cannot focus on one scene. How much to show the translator had to be decided.

Options considered

- Give each scene alone: gaps breed hallucinations ("hallucination is gap-filling under uncertainty")
- Give the whole script: attention thins, details from other scenes bleed in, and cost explodes
- Build the ledger while translating: an early mistranslation poisons the ledger, and every later scene stacks on it
- ✓ Closed packets plus a read-only ledger built from the source up front

Decision

The translator gets a packet holding only four layers with different lifetimes: what stays fixed for the whole film, state that only grows, what this scene alone needs, and beat summaries of the scenes on either side. The ledger (who knows what, forms of address, props that have appeared, characters who have died) is built for every scene from the source before translation, and the translator only reads it. The packet has one escape hatch: "Information not in this packet does not exist. Do not invent it; report it."

Result

The packet builder turned into a whitelist of layers 1–5 that admits only what may go in. A rule was added to be suspicious when nothing at all is reported. Whether a packet leaks later plot is tested by having an agent that does not know the answer guess the twist (see Quality).

Cost & remaining risk

In scripts without a twist, the leak filter switched off and later plot leaked (Oct 1, and again Oct 31). The leak check uses the same rules that build the packet, and the record itself calls it "only a necessary condition".

Sections in parallel, tried for real, then a sequential relay My input

Problem

One feature (Work A, 91 scenes) is too long to translate in one go. At first, five sections were translated at the same time.

Options considered

Translate the sections within a language at the same time (the first run, actually tried)

✓ Sections within a language in order; languages in parallel

Decision

Translating at the same time broke the feature. Two sections each wrote their own version of a recurring lyric, and the ending section could not match a line that pays off at the end after being repeated three times. One form of address appeared in four variants, and emphasis was marked three different ways. The record says "the one who plants and the one who harvests wrote at the same time without knowing each other". I pointed this out and had the sections switched to sequential. Each section gets a fresh translator, and only the finished script and the hand-off from earlier sections are passed on. The reason given: "Sequential is stronger for both context and leaks. The earlier script contains nothing from later on."

Result

Three things changed with it. Strings every section must render the same way (forms of address, recurring lines and the like) were locked before translation, and the locks went from 25 to 60 by the seventh revision. Packets switched from a blacklist to a whitelist. A three-layer gate went in between sections. Translated again in sequence, Work A's full 6.5 Korean script (95 pages) was finished that night.

Cost & remaining risk

Waiting on the previous section made runs longer. On the 7.0 draft each section took one to two hours, and the record says "one run took too long", which led to cutting steps (decision 5).

Hallucination as a checkable fact: machine checks and automatic rejection Agent default · no sign-off on record

Problem

The translator could invent characters, props or numbers not in the source. In a relay, a hallucination in one section spreads to the next.

Options considered

Send it to the AI director for arbitration, like emotional drift

✓ Block it with machine checks and reject automatically (no arbitration)

Decision

The reasoning was "drift is a judgment between fidelity and localization, but an invented entity is never the right answer", so there is no arbitration. The design document says "prompts do not guarantee it; only mechanical checks plus a blocking policy do". There are four checks: dialogue blocks per speaker, whether proper nouns and props are within the source and the lexicon, numbers, and consistency with the ledger. If any one fails, that section is translated again. A gate stands between every pair of sections, and the machine gate tests itself by planting faults (code below).

Result

All five full runs had zero hallucinations (by the pipeline's own gates). Counting dialogue blocks was a script's job in the design, which called it "the most reliable detector". But the script was off by 10, and four attempts to replace it with a speaker-alignment tool failed. So the auditor does the counting, and the claim in the design document was corrected.

Cost & remaining risk

Hallucinations that subtly twist subtext are not caught by these checks. That is left to re-measuring emotional intensity and to the blind AI screening.

Grade the auditor with planted mutations

Agent default · no sign-off on record

Problem

How to measure translation quality had to be decided. A human reference translation existed, and the judge of each translation was itself an AI auditor.

Options considered

Score the translator by similarity to the reference (BLEU-style): "different and wrong become indistinguishable", and any mistakes the reference may contain harden into the answer

A registry that turns backtest failures into correction rules for the agents: the sample is too small, and with both auditor and translator being Claude the measurement is self-referential

✓ Plant faults in the reference on purpose and measure how many the auditor finds

Decision

The auditor's recall and false-positive rate are what get measured; the design document notes that "the false-positive rate is more decisive". The pass bar is recall of 80% or more and false positives of 10% or less, and an auditor below it is not used for real work.

Failures go into a single append-only failure log, and a failure becomes one line in a definition file only when it repeats across several scenes.

Result

The first attempt failed at 58%. The second, with more judging criteria, was still 58%. The third widened the check to every beat and stage direction and passed at 83% (10 of 12), with no confirmed false positives. The record reads "where to look works better than what to look for". The machine-check layer flagged none of 9 good free renderings.

Cost & remaining risk

Auditor and translator are the same model family, so this grading is still self-referential. On an external benchmark it lost to a single prompt on the same model (see the retrospective).

Steps that found nothing new come out by the same standard that put them in My input

Problem

On Work A's 7.0 draft one run took too long (one to two hours per section). What each step actually found had been measured on that 7.0 run.

Decision

On Sep 17 the steps that found nothing new on the 7.0 run were removed: the consistency audit, the full continuity audit, the format agent (14 minutes, zero violations), the retroactive re-audit of locked strings, the second correction round (4 lines changed per round) and the repeat leak test. Steps that had paid off stayed. One leak test caught a 15.5-point leak, and the machine gate found a bug that skipped locks. The principle: "remove steps by the same standard used to add them".

Result

Five full audits became one plus machine checks, correction rounds went from two to one, and screening and auditing now run at the same time. After that, Works C and D each went through the whole pipeline in one day (by file timestamps).

Cost & remaining risk

Improvements found during a run were to be applied in one batch after it finished. On Oct 31 Work D paid for that: 11 kinds of deferred defects came back section after section and step after step (a capitalization issue in five sections in a row, field names confused six times, and so on).

Genre as judging criteria in each script's settings

Agent default · no sign-off on record

Problem

Each script was a different genre. Whether to keep separate translators and auditors per genre had to be decided.

Options considered

Genre-specific agents: 11 roles × N genres

✓ The same 11 roles, plus genre failure conditions in each script's settings

Decision

Genre-specific agents were rejected for three reasons. The combinations explode and 95% of each definition would be duplicated. Most films are not a single genre. Genre knowledge belongs in the judging criteria. The exception: a new role is added only for a skill no existing role has (lyric metre, period accuracy, action clarity, technical jargon).

Result

Four genres were handled by the same 11 roles plus each script's genre failure conditions. If even one AI reader hits a genre failure condition in the screening, that scene is reworked.

Quality and verification

What got graded was the auditor, and it was tested first on scripts broken on purpose.

What got graded was the auditor. Twelve faults of nine types were planted in the reference translation, and the test measured how many the auditor found. The pass bar is recall of 80% or more and false positives of 10% or less; an auditor below it is not used for real work. It passed at 83% on the third attempt (figure below).

Audits of the translation are blind. The auditor reads only the translation, rewrites the scene contracts from it and saves them first, then opens the source contracts to compare. If it opened the contracts first, the audit is void and runs again. Emotional intensity is re-measured from 1 to 10 for every beat; a gap of 3 or more from the source is a failure and 2 is a warning. This is reverse inference at the level of the scene contract.

Whether meaning gets lost across cultures is checked separately. The cultural adviser writes neutral questions about what a reader of the source takes away, and AI readers who saw only the translation answer

them. Answers are sorted into arrived, partial and lost; a lost meaning goes back to the AI director even if the item is locked.

The checking tools are checked too. The section gate plants quoted dialogue, a deleted scene and an altered locked string in a copy of the script, and if it misses any of them its results are not trusted. Packet leaks are measured by giving a fresh agent that does not know the answer either the source alone or the source plus the packet, and having it guess the twist. The hit rate has to stay within the source-only baseline plus 10 points.

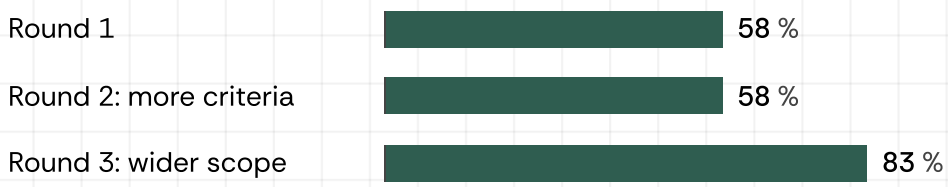
The four AI readers and five AI investor profiles read only a copy of the translation kept outside the project. As the reports say, this is not audience research, so this write-up does not count screening or investment verdicts as results. Zero hallucinations and matching dialogue-block counts are also the verdicts of the pipeline's own gates. The record itself notes the limit that one model family is grading its own family's translations.

There was no CI, no PRs and no hooks. What enforces the gates is the wording of the skill and the role definitions, plus the tools' exit codes (the section gate blocking, the tone-mark check failing). Re-run on Nov 10 for this write-up, 11 of the 12 tool self-tests passed; one could not run because the DOCX library was not installed. These are not a test suite with counted assertions. The regression results from the Oct 2 batch of fixes (self-tests 8/8, gate self-checks 8/8) come from the work log and were not re-run.

Auditor backtest recall

Adding judging criteria in round 2 changed nothing; widening the check to every beat and stage direction in round 3 cleared the 80% bar.

Source: Share of 12 planted mutations (9 types) the auditor found, strictly matched to the labels. No confirmed false positives in any round. Pass bar: recall ≥ 80%, false positives ≤ 10%.



One feature, start to finish

This follows one beat from a trial sequence (4 scenes) where the translation drew the opposite reaction to the source. It all happened on Sep 16, 2026, and most of the calls were made by AI agents (the AI director and AI readers). Only the structure is given here.

1 **Screening: a beat that flipped (12:49)**

On a beat where all four AI readers of the source reacted as intended, two AI readers of the English version and, before any fix, all four of the Korean version laughed. The suspected cause was word choice.

2 **A precedent disproved (12:51)**

The AI readers' reactions showed that a mimetic-word rule from the precedent book was draining the tension in this scene; from then on, precedents were also something to verify.

3 **First fix, failed (13:38–13:40)**

Korean mimetic words were cut from 16 kinds to 4, yet the AI readers who laughed went from 4 to 3 in Korean and from 2 to 3 in English. The laugh had moved to the translator's new substitutes, and both languages failed the same way. The AI director had a gate that checks one scene first, so it stopped before the other 21 fixes were made.

4 **The skipped check, run (14:43)**

The fidelity audit, never once run on a real translation, was run. It found zero intensity failures. The cause was not the translator: guidance revised after the screening had never reached the scene contracts written before it.

5 **Contracts fixed, translated again (14:46 → 15:04)**

In Korean, the AI readers who laughed went from 4 to 2, and those who read the genre as intended went from 1 to 4. None found it flat or too dark. The remaining laugh sat in the same spot in both languages, and the AI readers of the source did not laugh there.

6 **Source-level ruling (15:32)**

Four issues that translation could not solve were sorted: 1 handed to staging, 2 sent to the writer as queries, 1 kept as in the source. The translator was frozen on those spots and they were excluded from the translation quality score. The reason: "leave a failure the translation did not cause on the translator, and the next run fixes the wrong place again". Because it was the riskiest ruling, a re-screening at the storyboard stage was made mandatory.

7 **Made into procedure**

A fourth destination for audit findings was added: "source-level". Findings go there when source and translation readers diverge while the translation is accurate, or when source readers show the same problem. Two things went into the correction round as well: changed scenes are always re-audited, and repeated findings are watched for.

Retrospective

On an external benchmark the pipeline lost to the same model with a single prompt, and twice a "pass" turned out to mean a check had not run at all.

On an external benchmark, the pipeline lost to a single prompt. On Sep 17 the dialogue alone was translated and compared against the official subtitles of three public features (131 pairs per language). The pass criteria were hash-locked before looking at the data. Against the official subtitles, the pipeline won (win rate 76.7% in Korean, 84.4% in English). The same model with a single prompt beat the official subtitles by more (82.4%, 88.5%). Head to head, the pipeline's win rate was 32.1% in Korean and 24% in English, with the upper confidence bound below 50%: a loss (by the primary judge; by the other judge it was lower still, 23.3% in Korean and 15.6% in English). The result file lists four limits alongside, three of which are that the judge is the same model family as the translator, there were three films of the planned six, and dialogue-only mode is a lower-bound estimate for the full pipeline. The plan was a judge from another company's models, but no key was available, so the same family was used. No write-up interpreting the result was made.

Twice, a "pass" was not a pass of any check. The first leak check confirmed "zero remaining" with the same function that had filtered the packet. The record says "if the checking tool and the generating tool are the same, it is not a check", and the check became a blind guessing test

by an agent that does not know the answer. On Oct 31, in the first Korean → Vietnamese script, a tool that assumed Vietnamese source text skipped every lock comparison and showed "pass" without a warning. The lesson written down afterwards: "on the first script in a new language pair, first check whether zero findings means zero or not run".

A tool deliberately left out at first became the core of the gates. The design document said writing a script parser and page counter in Python was "left out on purpose… if it becomes a bottleneck, a 150-line stdlib script then". In practice the first batch of commits already held three intake tools, including the parser (85 lines), and those tools became the core of the gates and, with the external-benchmark tools, grew to 26 (4,816 lines). Counting that needs judgment, though, was not handed to a machine (decision 3).

Records of finished runs were overwritten twice. On Oct 1, while taking a regression baseline for tool fixes, the originals of some of Work B's packets were lost; an output-location option and a rule that "finished run folders are records" were added. On Oct 31 the orchestrator again overwrote Work D's first-section packet and gate results. This time they were rebuilt from the revision backups, checked to be byte-identical, and restored.

How I work

This project was also built with a role-based AI agent team. The team and its gates differ from project to project; this one's are under "My role" above. Below is the method as written up while building the EWA platform.

See how I set up and ran the work on the 2026 EWA platform →

https://jake-jo-portfolio.vercel.app/en/process/

Contact

whxoans@gmail.com
+82 10-9233-7440