
Building a Conversational Archive

*How five programs turn recorded television
into a searchable question-and-answer library*

A non-technical overview

Prepared for the project owner · August 2026

SUMMARY

This paper describes a working system that listens to political talk shows and turns them into something a computer can hold a conversation with. Television is recorded automatically on a schedule. The captions that scroll along the bottom of the screen are captured and cleaned. The conversation is broken into its natural back-and-forth units — someone asks, someone answers — and each exchange is given a subject. A person reviews the results. The answers are then rebuilt into full, readable replies matched to the exact seconds of video where they were spoken. Finally the long spoken questions are reduced to the short ones a searcher would actually type.

The output is a library of questions paired with real answers, given by real people, on the record, with the video to prove it. The paper follows one exchange from broadcast to finished entry, and explains why each stage exists.

Systems described: CCProc · TVShowDialogueCreator · AIMLEditor · AIMLMultiSentenceCreator ·
AIMLPatternReducer

1. What the System Is For

Most conversational software makes its answers up. Ask it a question and it assembles a reply out of patterns it has absorbed, with no particular person standing behind the words. The system described here takes the opposite approach. Every answer it can give was actually said, out loud, by a named person, on a broadcast, at a moment that can be pointed to on a recording.

The raw material is daily political television — programs such as *Washington Journal*, *The McLaughlin Group*, *Hardball*, *Andrea Mitchell Reports*, *PoliticsNation*, *MTP Daily*, *All In*, and *The Beat*. These shows are almost entirely question and answer: a host asks, and a guest, panelist, or telephone caller responds. That structure is the whole opportunity, because the material already arrives in the shape the system wants to store it in. The work is to find where each exchange begins and ends, to work out what it was about, and to make the question askable by someone who was not watching.

Doing this by hand would be impossibly slow. A single hour-long program contains dozens of exchanges, and the archive grows every day television is broadcast. The work is therefore divided among five programs that run one after another, each handing its results to the next, with a human reviewer placed at the one point in the chain where human judgment genuinely pays for itself.

The pipeline at a glance

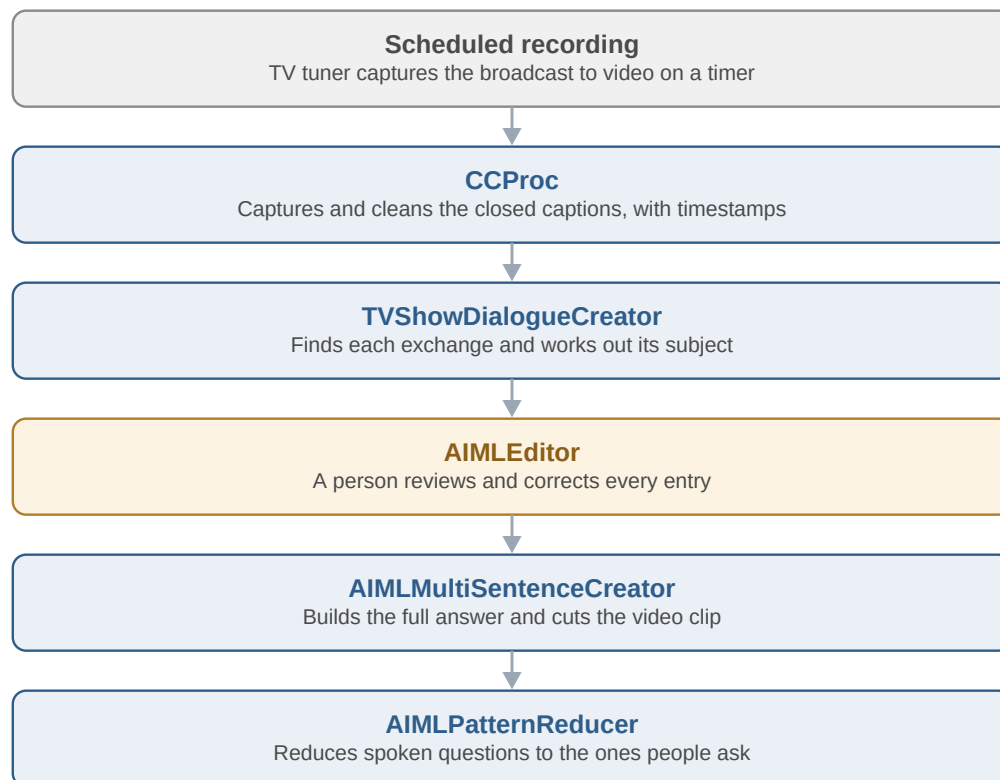


Figure 1. The five processing stages, in running order. The shaded stage is the human review step.

2. The Raw Material: Recording the Broadcast

Everything begins with a television tuner — a Hauppauge WinTV-HVR-955Q — set to record on a schedule. Programs air at fixed times, so the capture runs unattended: the tuner wakes, records the hour to disk, and stops. Nobody has to be present. Over months this builds a substantial library of raw episodes organised by show and date.

Two things are captured, and both matter. The first is the picture and sound, which the finished archive will eventually point back to. The second is the closed-caption text — the transcription broadcasters are required to carry for viewers who are deaf or hard of hearing. That caption stream is the system’s transcript, and it arrives free with the broadcast.

Captions are a gift and a problem in equal measure. The gift is that they are produced live by human stenographers and are broadly accurate about who said what. The problem is everything else: they arrive in capital letters, with little or no punctuation, often lagging a second or two behind the speech, sometimes garbled, and with no marking of where one person stops talking and another starts. A caption stream is a river of words. Turning that river into discrete, attributable exchanges is the central challenge of the entire system, and the reason it takes five programs rather than one.

WHY NOT JUST USE SPEECH RECOGNITION?

The system does have a speech-to-text path, and uses it — but as a fallback rather than a primary source. Broadcast captions are typed by professionals who understand the context, know the speakers’ names, and can spell an unusual surname correctly the first time. Automatic transcription of political television, with its crosstalk, its proper nouns, and its telephone callers on poor lines, is markedly worse. So captions are the default, and the audio is transcribed only where captions are missing or unusable.

3. Stage One — CCProc: Cleaning the Captions

CCProc takes the recorded broadcast and produces a usable transcript from the caption stream. This sounds mechanical, and much of it is, but the details determine whether everything downstream succeeds.

Preserving the clock. The single most valuable thing CCProc does is keep the timing. Every fragment of text is stored with the moment in the recording at which it appeared. This is what makes the finished archive more than a collection of sentences: because the timing survives every subsequent stage, a finished answer can be traced back to the exact seconds of video in which it was spoken, and that segment can be cut out and attached to it. An answer in this archive is not a claim about what someone said — it is a pointer to them saying it.

Making the text readable. Caption text as broadcast is uppercase, unpunctuated, and broken into arbitrary lines reflecting how fast the stenographer was typing rather than the shape of the sentences. CCProc works out where sentences actually end, applies punctuation and capitalisation, and reassembles the fragments into coherent lines. The punctuated version is kept separately from the raw one, because both stay useful — the raw text is closer to what was broadcast, the punctuated text is what a reader will eventually see.

Handling the gaps. Captions fail: a segment drops out, a feed garbles, a caller’s line is too poor to follow. Where CCProc finds a hole, the corresponding audio is transcribed by speech recognition instead. The two sources are kept apart so later stages know which transcript they are working from.

What leaves this stage is a clean, punctuated, fully timestamped transcript of an entire episode. It is readable — but it is still an undifferentiated stream of talk. Nothing yet knows it contains questions.

4. Stage Two — TVShowDialogueCreator: Finding the Conversation

This is where a transcript becomes a conversation. The program reads an episode from beginning to end, performing three judgments over and over: where one exchange ends and the next begins, which part is the question and which the answer, and what the exchange is about.

4.1 Breaking the stream into exchanges

The program works forward looking for the next question, then looks ahead for the response belonging to it. This is harder than it sounds. Televised conversation does not proceed in tidy alternation. Hosts interrupt. Guests answer a question asked two minutes earlier. Rhetorical questions expect no answer. A question is asked, abandoned halfway, and reformulated.

So the program does not simply take the next sentence ending in a question mark. It looks for an answer with genuine content — a response with a real subject that says something, rather than “Right.” or “Sure, Chris.” If the immediate reply is empty in this sense, it keeps looking. An exchange is only worth recording if the answer is worth reading.

4.2 Different shows, different rules

Much of this stage exists because the programs being recorded are not built the same way, and one set of rules handles them badly. **The standard interview format** — host and guest, one at a time — is the baseline the general logic covers. *Washington Journal* is built around telephone calls from the public: segments open with the host framing a topic, then run through a queue of callers who introduce themselves by name and city before making a point. The system detects where one caller’s segment ends, strips the recurring introductory formulas that carry no content, and recognises the name-and-location opening marking a new caller. *The McLaughlin Group* is a roundtable where several panelists argue at once and turn-taking is aggressive; it gets its own handling for tracking which panelist is responding to which prompt.

This is a recurring theme: there is a general method, and there are hand-built exceptions where the general method produces poor results. That is not inelegance. It is the shape any system takes when it is judged on the quality of its output rather than the tidiness of its design.

4.3 Knowing who is speaking

Captions rarely say who is talking. Sometimes a speaker change is marked, sometimes not, and on a roundtable it is frequently wrong. Since an answer without an attributable speaker is close to worthless here, the system does not rely on the captions. Instead it listens: the audio for the segment is extracted and passed to voice biometrics software, which compares it against stored voice profiles. Regular hosts are identified reliably because the system has heard them hundreds of times. Guests can be enrolled on first appearance and recognised afterwards. Callers are at minimum distinguishable from the host, which is what matters on a call-in show. The voice, not the transcript, is the authority on who spoke.

4.4 Finding the topic

Every exchange needs a subject. Without one the archive is a heap of unconnected questions; with one, related exchanges can be grouped, pronouns can later be resolved, and the same question asked about two different subjects stays distinct. The system attacks this with a sequence of methods, each tried in turn until one succeeds.

THE TOPIC CASCADE — TRIED IN ORDER

1. **The host says it outright.** Broadcast hosts announce their subjects: “Let us talk about the debt ceiling.” What follows such a phrase is the topic. Cheapest and most reliable, so tried first.
2. **The guest introduction.** “Joining me now is the senator from Connecticut.” The system recognises this pattern and identifies the person named, which frequently establishes what the segment will be about.
3. **Language analysis of the answer.** The answer is sent to a natural-language service that identifies the people, places, organisations, and concepts in it; the most prominent becomes the topic.
4. **Language analysis of the question.** If the answer yields nothing usable, the same treatment is applied to the question that prompted it.
5. **The surrounding conversation.** As a last resort the system widens its view to the material before and after, on the assumption that a segment stays on one subject even when an individual exchange within it does not name that subject.

The ordering reflects both confidence and cost. The early methods are simple text recognition: fast, free, and highly reliable when they apply. The later ones require calls to an outside language service, which is slower and metered. The system therefore tries to avoid them, and enforces a daily limit on how many such calls it will make.

4.5 Working within limits, and surviving interruption

Two practical constraints shape this stage. The first is the metered service: the program counts its calls and stops at the day’s allowance rather than running up an unbounded bill. The second is time. Processing a backlog of recorded television takes many hours, and any long-running job will eventually be interrupted. The program therefore records its position continuously, noting which show and episode it has reached, and resumes there when restarted. Work already done is never repeated — which matters when the expensive part of the work is metered.

What leaves this stage is a set of draft entries: for each exchange, a question, an answer, the subject, and the conversational context it sat in, all still carrying their timestamps and recorded in a central database.

5. Stage Three — AIMLEditor: The Human in the Loop

The third stage is a person at a screen. After two stages of automation, draft entries are presented one at a time for review — the only point in the pipeline where human judgment is applied directly to individual entries.

The reviewing screen shows everything needed to judge one entry at once: the question, the answer, the assigned subject, the conversational context, and the original timestamped transcript it was drawn from, so the reviewer can see what was actually said around it. Navigation runs forward and backward through the entries of an episode and from one episode to the next. The available actions are deliberately few, because the reviewer will perform them thousands of times and speed matters more than expressiveness:

The reviewer can **accept** the entry as generated and move on, which is the common case and takes one keystroke; **rewrite** the question or answer where the automation produced something awkward; **correct or clear the subject** where the topic cascade guessed badly, on a dedicated screen, since it is the field most often wrong; **skip** the entry entirely, because not every exchange on television is worth keeping and discarding is as important an editorial act as correcting; or **run automated editing**, applying routine corrections in bulk so attention goes to genuine judgment calls rather than mechanical fixes.

Spell-checking runs here too, which matters more than it might appear: live captioning produces misspellings constantly, particularly of proper nouns, and those errors would otherwise be preserved permanently in the archive.

WHY A HUMAN, AND WHY HERE?

The reviewer sits at exactly the right point in the chain. Everything before is mechanical work a person would do slowly and no better — parsing captions, tracking timestamps, cutting the stream into units. Everything after is expensive work that should only be performed on entries known to be worth it: building full answers, extracting and encoding video, generating question variants.

Placing review before that expensive work means no effort is spent perfecting an entry that was never any good. It also makes the human corrections permanent — edited text is stored separately from automatically generated text, so later processing builds on the corrected version and no subsequent automated pass can silently undo a human decision.

6. Stage Four — AIMLMultiSentenceCreator: Building the Full Answer

This stage takes reviewed entries and turns them into finished ones. It does three substantial things.

Restoring the whole answer. Earlier stages tend toward short answers, because a short answer is easier to isolate and verify. But people on television do not answer in single sentences. They qualify, elaborate, give an example. A one-sentence extract is often accurate and useless at the same time — it captures the assertion and loses the reasoning. This stage rebuilds the answer at its natural length, extending across the following sentences for as long as the speaker is still answering the same question. Knowing where to stop is the whole difficulty: extend too little and the answer is a fragment; extend too far and it runs into the next question, or another speaker, and the entry becomes wrong rather than merely incomplete.

Re-deriving the timing. Once the answer spans several sentences, its start and end must be recalculated. The system computes these against the original caption timings and deliberately pads them — by a couple of seconds, or five, or ten, depending on the situation. The padding matters because captions lag speech: a clip cut to exact caption timing will clip the first word off the front and hang a beat too long at the end. The padding compensates for a known, systematic delay.

Cutting the video and verifying the speaker. With reliable timings, the corresponding segment is extracted from the recorded video and stored as a clip that can be played back alongside the answer text. This is what makes the archive evidential rather than anecdotal. The speaker is then checked once more against the audio of the final clip. This is not redundant: the answer has grown since the earlier check, and the extension may have crossed into another person's speech. Verifying the clip that actually exists is the only check that establishes the finished entry is correctly attributed.

Because video extraction and encoding are the slowest operations in the system, this stage is built to run as several simultaneous jobs on different portions of the material. What emerges is a finished entry: a question, a full answer in readable prose, the subject, the identified speaker, the source episode and broadcast date, and a video clip of the answer being given.

7. Stage Five — AIMLPatternReducer: Making Questions Askable

The final stage runs differently from the four before it. Rather than working through one episode start to finish, it works across the whole accumulated library, picking up where it last left off and moving forward through everything added since. It solves a problem that only becomes apparent once the archive is large enough to be used in earnest.

The questions in the archive are real questions asked by real broadcasters, and broadcasters do not ask short questions. A televised question is a performance. It carries context for the audience, restates what the guest said earlier, hedges, names the person being addressed, and frequently bundles two or three questions into one sentence. Something like:

“So given everything that has happened this week, and I want to get your reaction to this, do you think the leadership actually has the votes, and can they hold their caucus together through Friday, Chris?”

Nobody searching the archive will ever type that. They will type “Does the leadership have the votes?” The archive holds an excellent answer to that question, and without help it will never be found. This stage bridges the gap, converting long spoken questions into the short natural ones a person would actually ask — and generating several where the original contained several.

What the reduction involves

- **Splitting bundled questions.** Where one utterance contains more than one question — joined by “and,” or by a second question word — it is separated into individual questions, each becoming its own way into the same answer.
- **Resolving pronouns.** A dedicated pronoun resolver runs as part of this stage. Spoken questions are full of “he,” “she,” “they,” and “that” — perfectly clear in the moment, meaningless in isolation. The resolver looks backward through the conversation to find what each pronoun referred to and substitutes the actual subject, consulting, where the pronoun is gendered and the antecedent is a personal name, a service that determines whether a given first name is male or female.
- **Repairing question form.** Spoken questions are often not well-formed once lifted out of context. A past-tense verb may need converting to its base form with “did” inserted, so a fragment becomes a properly shaped question that reads correctly on its own.
- **Trimming what carries no meaning.** Trailing qualifiers, dangling prepositional phrases, and hanging clauses are removed — but only where what remains still ends properly. A question ending on a preposition or stray adverb reads as broken, so the ending is checked before any cut is committed.
- **Removing direct address.** Broadcasters constantly name the person they are speaking to: “... does that scan to you, Chris?” The name is conversational furniture and is stripped, with care taken not to remove a name that is part of the actual subject matter.
- **Expanding coordinated pairs.** A question about “the House and the Senate” becomes two questions, one about each, since a searcher will ask about only one.
- **Protecting idioms.** Figurative phrases must not be taken apart by the trimming rules, because “hold their caucus together” loses its meaning if cut in half. Recognised idioms are kept intact throughout.

Each reduced question is stored as an additional way of reaching the same answer, so a single recorded exchange may become reachable through half a dozen naturally phrased questions. The original question, as actually spoken, is kept as well. Like the earlier stages, this one works under a daily allowance for the language service it depends on, and resumes from its last position when interrupted.

8. The Complete Transformation, Traced End to End

The preceding sections describe each stage in isolation. This section follows a single exchange all the way through, which is the clearest way to see how the parts fit together. The example is composed to be representative rather than drawn from a specific broadcast.

STEP 1 · THE BROADCAST

A public-affairs program airs at nine in the morning. The tuner, running on a schedule, records the hour to disk. Nobody is watching. Roughly thirty-one minutes in, the host turns to a guest and asks about a pending vote; the guest answers for about forty seconds.

STEP 2 · THE CAPTIONS ARE CAPTURED AND CLEANED (CCProc)

The caption stream is extracted from the recording. As broadcast, the passage is uppercase, unpunctuated, and split across lines at arbitrary points, each fragment carrying the moment it appeared on screen. CCProc reassembles the fragments into sentences, restores punctuation and capitalisation, and preserves every timing. The passage is now readable prose — but still an undifferentiated stream, indistinguishable from the thirty minutes before it.

STEP 3 · THE EXCHANGE IS ISOLATED (TVShowDialogueCreator)

Reading forward, the program identifies a question and searches ahead for a substantive reply. The guest's first words are "Well, look —", which carries no content, so the search continues to the sentence where the guest actually begins answering. The boundary of the exchange is set from the end of the previous answer to the end of this one.

The audio for the passage is extracted and compared against stored voice profiles. The host is recognised immediately; the guest is matched to a profile enrolled during an earlier appearance. The exchange is attributed from the voice, not from the transcript.

STEP 4 · THE SUBJECT IS DETERMINED (the topic cascade)

The first method is tried: does the host announce the subject outright? There is no "let us talk about" phrase here, so it fails. The second: is this a guest introduction? No — the guest was introduced several minutes earlier. The third sends the answer text to the language service, which identifies the legislation under discussion as the most prominent entity in the passage. That becomes the subject and the cascade stops. One metered call was spent; the day's allowance drops by one.

A draft entry now exists — question, answer, subject, context, speaker, and timings — written to the central database.

STEP 5 · A PERSON REVIEWS IT (AIMLEditor)

The entry appears on the review screen beside the original timestamped transcript. The reviewer reads both. The question is fine. The answer's opening clause is a fragment left over from the guest's hedging, and is trimmed. The subject is correct but awkwardly phrased, so it is rewritten to the form the legislation is commonly known by. A misspelled surname, introduced by the live captioner, is corrected.

The reviewer accepts the entry. The corrections are stored separately from the automatically generated text, so everything downstream builds on the edited version and no later automated pass can overwrite these decisions.

STEP 6 · THE FULL ANSWER AND THE CLIP ARE BUILT (AIMLMultiSentenceCreator)

The approved one-sentence answer is extended forward. The guest's next two sentences continue the same answer — a qualification and a supporting example — and are absorbed. The third sentence begins a new point, and the extension stops.

Start and end times are recalculated across the three sentences and padded at both ends to compensate for the lag between speech and caption. The corresponding video segment is cut and stored as a clip. The speaker check runs once more, on the audio of the clip that actually exists — confirming the extension did not stray into another person's speech.

The entry is now finished: a question, a full three-sentence answer in readable prose, the subject, the identified speaker, the broadcast date, and a video clip of the answer being given.

STEP 7 · THE QUESTION IS MADE ASKABLE (AIMLPatternReducer)

In the final stage the long spoken question is reduced. As asked, it opened with a framing clause about the week's events, referred to the guest's party as "they," bundled two questions together with "and," and closed by addressing the guest by first name.

The framing clause is trimmed. The pronoun resolver works backward through the conversation to the party actually being discussed and substitutes it in. The bundled questions are split into two. The trailing first name is removed as direct address. Each resulting question is checked to confirm it ends properly and reads as a complete question on its own.

Two short, natural questions now point at this answer, alongside the original as it was spoken. Someone typing either of them — in words the broadcaster never actually used — reaches a real answer, given by a named person, on a known date, with the video to prove it.

Seven steps, five programs, one human decision point. Multiplied across dozens of exchanges per episode and hundreds of episodes, this is how a television schedule becomes a conversational archive.

9. What the Finished Archive Contains

Each entry holds a question, an answer, and the evidence the answer was really given — specifically: the question as actually asked on air, together with shorter reduced forms a person might plausibly type; the answer at its natural length, punctuated, readable, and human-reviewed; the subject the exchange concerns; the conversational context, which distinguishes the same words spoken in two different discussions; the speaker, established from the voice; the provenance — which program, which broadcast, and the position within it; and the video clip of the answer being spoken.

Entries are stored in the conventional structure used for conversational agents, which is why the programs are named as they are: a pattern to match against what the user asks, a template holding the reply, a topic, and a record of what was said just before. That format was designed for hand-authored chatbots, where a person writes both question and answer. Here it is being filled from broadcast reality instead — an unusual and rather appealing use of it.

WHAT THIS MAKES POSSIBLE

Because every answer is attributed, dated, and backed by video, the archive supports things a generative system cannot honestly offer. You can ask what a particular figure said about a subject and receive their actual words. You can follow how the answers to the same question changed over months of broadcasts. You can compare what two people said about the same matter in the same week. And in every case the response is not a summary or a paraphrase — it is a recording of a person saying it.

10. Observations on How the System Is Built

Cheap methods first, expensive methods only when needed. The topic cascade is the clearest example, but the principle is everywhere: simple text recognition before language analysis, captions before speech recognition. The pattern holds because the cheap methods are not merely cheaper — when they apply they are usually more reliable. Recognising that a host said “let us talk about the debt ceiling” is both free and certain. Inferring the same subject from analysis is neither.

Everything can be interrupted and resumed. Every long-running program keeps a note of where it reached and consults it on startup. This is not defensive programming for its own sake. Processing months of television takes hundreds of hours, and any process running that long will be interrupted. Because the most valuable work is metered and paid for, repeating it is not just slow but costly. Resumability is a budget decision as much as a reliability one.

Human corrections are permanent. Edited text is kept apart from generated text throughout. Once a person has fixed an entry, later automated passes build on the fixed version and cannot silently revert it. In a system that will reprocess its own archive repeatedly over years, this is what makes review worth doing at all — without it, every improvement to the automation would risk quietly discarding accumulated human judgment.

The evidence is the point. Timestamps are preserved through every stage at real cost in complexity, and video is cut and attached at the end. None of that is necessary to produce a question-and-answer library. It is necessary to produce one that can be checked. That priority — verifiability over convenience — is the system’s defining commitment.

11. Limitations and Directions

- **The archive can only answer what was broadcast.** Its coverage is that of the programs recorded, and its perspective is theirs, including that medium’s preoccupations and blind spots.
- **Human review is the throughput ceiling.** Four stages run unattended; one requires a person to look at every entry. As the recording schedule grows, that stage limits how much material can be finished.
- **Caption quality sets an upper bound.** Where captions are poor everything downstream inherits the damage; the speech-recognition fallback mitigates but does not eliminate this.
- **Reduction is hard to get uniformly right, and voice identification degrades under crosstalk.** The accumulation of special-case rules in the reduction stage reflects how many ways human speech finds to complicate the task; roundtable programs remain the hardest case for attribution, which is precisely why those formats needed dedicated handling.

The most promising direction is the review stage. Modern language models are considerably better at judging whether an extracted exchange is coherent, whether an answer is complete, and whether a topic is well-chosen than the rule-based methods available when this system was first built. Using such a model as a first-pass reviewer — not to replace the human, but to sort entries so attention goes to the doubtful ones — would directly relieve the main bottleneck without surrendering the verifiability that gives the archive its value.

12. Conclusion

The system does something that sounds simple and is not: it watches television and remembers what was asked and what was answered. The difficulty lies in the gap between how conversation actually happens and how it must be stored to be useful. People interrupt, trail off, refer to things by pronoun, bundle three questions into one sentence, and address each other by name mid-thought. Captions arrive uppercase, unpunctuated, and late. Nothing about the raw material is tidy.

Five programs close that gap in sequence: captions are captured and made readable; the stream is cut into exchanges, attributed by voice, and given subjects; a person reviews the result; the answers are restored to full length and matched to the video; and the questions are reduced to the form a searcher would actually use. What comes out is a library of real answers from real people, each traceable to the moment it was spoken.

That last property is the one worth emphasising. It would be far easier to build something that produces plausible answers about what politicians think. This system was built to produce checkable ones instead — and every piece of complexity described here, from the preserved timestamps to the voice verification to the padded video clips, exists to serve that single commitment.

Appendix · Terms Used in This Paper

Closed captions	The text transcription carried with a television broadcast, typed live by a stenographer. The system's primary source of text.
Timestamp	A record of when something appeared in the recording, preserved through every stage so finished answers can be matched back to video.
Exchange	One question and the answer responding to it — the basic unit the archive stores.
Pattern and template	The stored question and the stored answer, in the conventional structure used for conversational agents.
Reduction	Converting a long spoken question into the short, natural forms a person would actually type when searching.
Voice biometrics	Identifying a speaker by comparing their voice against stored profiles — used here because captions are unreliable about who is talking.
Language service	An outside service identifying the people, places, and concepts in a passage of text. Metered, and therefore used sparingly.