8 Language models in clinic and science

AUTHOR
R. Heller

R. Heller · Date: 2026-08-23 · Last changed: 2026-11-09

At a glance

Guiding question: Which statement can be checked after generation?

For: clinicians, researchers, teachers and students who use, review or publish text produced with language models · **Prerequisites:** Chapter 7 (tracing a statement to its source); no programming experience is needed for the core reading

In the course: W2 · Language models and evidence · **Data:** synthetic (two stored answers and a twelve-sentence corpus written for this book; teaching file `data/b08-stored-answers.csv`); *published* findings only where cited · **Core reading time:** about 11 minutes

8.1 Case and decision

The daughter of a 78-year-old woman writes through the patient portal of a fictitious trauma department. Her mother has a hip fracture and will be operated on tomorrow. “Can anything be done so that she does not get confused after the operation? Should she take melatonin?” The resident on duty uses the hospital’s writing assistant, which stores two answers to the same question. Both sound competent, both cite literature, and they disagree. The resident has twenty minutes before the evening round. Every sentence that goes into the reply will be read as the hospital’s advice and may shape what the family does tonight. The decision is which generated sentences, if any, may be used — and the resident is entitled to use none of them.

8.2 Core reading

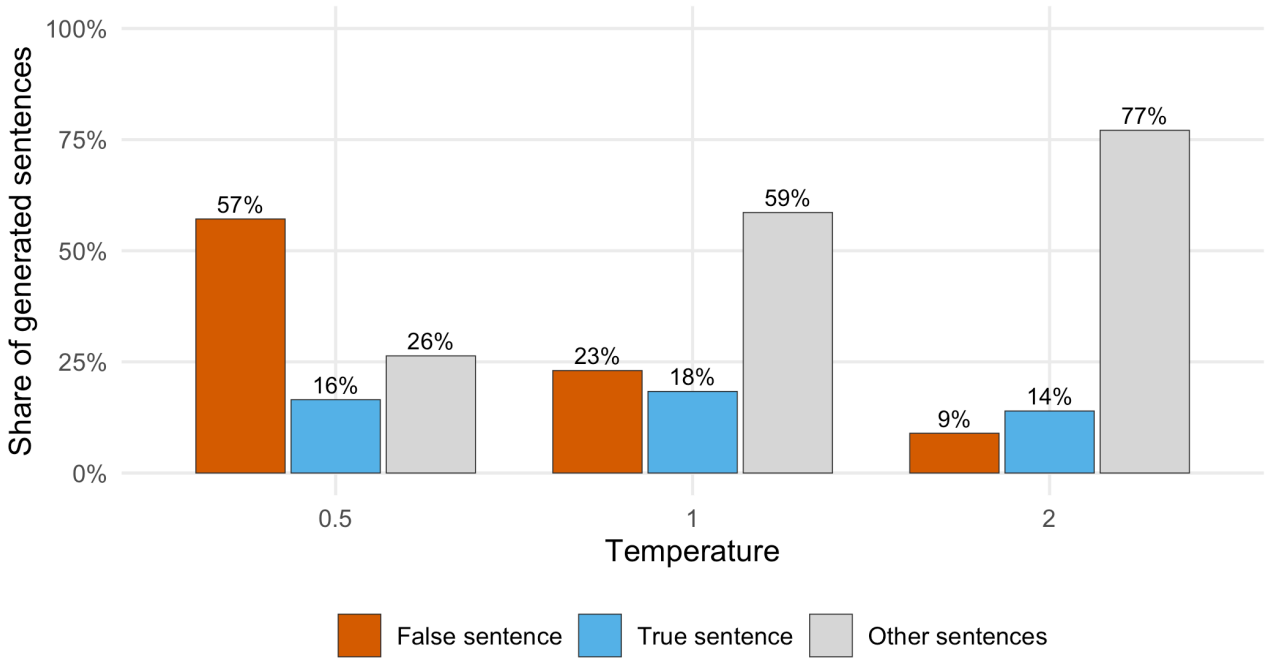
Core reading · required for W2 · about 11 minutes · ends before ‘Lab’

How a language model writes. A language model splits text into *tokens* — words or pieces of words — and learns, from very large collections of text, how likely each token is to follow the tokens before it. To write, it repeatedly picks a next token from this probability distribution and appends it. Modern models compute these probabilities with the transformer architecture (Vaswani et al. 2017), which lets every token draw on a long stretch of preceding text (Methods in R: Transformers). The principle can be seen in a model that is tiny by comparison: a *bigram model*, which predicts the next word only from the current word, using counts from a corpus (Jurafsky and Martin 2026, chap. 3). The lab builds one from twelve synthetic sentences about delirium. One of them states that melatonin does *not* prevent delirium after hip surgery; others say that various measures help prevent delirium after hip surgery, and that melatonin helps patients sleep.

Following the most frequent continuations, the model writes: “melatonin helps prevent delirium after hip surgery”. Every word pair in this sentence occurs in the corpus; the sentence as a whole does not, and it reverses the only statement the corpus makes on the question.

This is the central point of the chapter. The model is trained to produce *probable* continuations, not *true* ones. Plausibility and truth often coincide, which is why language models are useful, but nothing in the mechanism checks the second. Fluent, confident and wrong text is a known property of such systems, discussed as “hallucination” (Ji et al. 2023).

Temperature. Before sampling, the probabilities can be reshaped. Dividing the model’s internal scores by a *temperature* below 1 sharpens the distribution towards the most probable tokens; a temperature above 1 flattens it. In general, emphasising the most probable tokens gives text that people rate as more accurate and coherent but also more repetitive, while more weight on less probable tokens gives more diverse but less factual text (Jurafsky and Martin 2026, sec. 7.6.3); the decoding strategy alone can change the quality of generated text markedly (Holtzman et al. 2019). But probability is not truth. Figure 8.1 shows what happens in our small model, where the most frequent continuation is the false one: a low temperature makes the error more likely, not less.



2,000 generated sentences per temperature; start word 'melatonin'; synthetic corpus

Figure 8.1: Sentences generated by the bigram model from the start word “melatonin” at three temperatures (2,000 generated sentences per temperature). Bars: share that reproduces the true corpus sentence, share that produces the false sentence “melatonin helps prevent delirium after hip surgery”, and all other sentences. Unit: generated sentence; data status: synthetic.

Two different failures with sources. Generated text often carries references, and they fail in two distinct ways. A reference can be *fabricated*: the authors, title or journal do not exist. In one study of 636 references produced by two versions of a widely used chatbot for short literature reviews, 55% of the references from the older version and 18% of those from the newer version were fabricated, and many of the real ones contained substantive errors (Walters and Wilder 2023). A reference can also be *real but not supporting*: the paper exists, is on the right topic, and says

something different from the sentence it is attached to. The second failure is harder to catch, because a quick check of title and DOI succeeds. Systems that first *retrieve* documents and then generate an answer from them (Lewis et al. 2020) reduce the first failure but not the second; a retrieved paper can still be summarised wrongly.

The two checks must therefore be kept apart. *Existence* is bibliographic: does the DOI resolve, do authors, title, journal and year match? *Support* is substantive: does the cited passage say what the sentence claims, for the population and outcome in question? Table 8.1 applies both checks to the five claims in the two stored answers.

Table 8.1: Evidence card for the two stored answers: existence and support are judged separately; the action follows from both. Data status: the answers are synthetic; the findings named under ‘support’ are published.

Claim (answer)	Source given	Source exists?	Source supports claim?	Verdict	Action in the reply
A1 Melatonin reliably prevents postoperative delirium	de Jonghe et al. 2014	yes	no: no reduction in incidence (29.6% vs 25.5%)	real source, contradicts claim	reject
A2 Glasses, hearing aids and family visits help orientation	none	not applicable	not assessable without a source	unsourced	adapt only after finding a source
B1 Measures for orientation, sleep, vision, hearing, fluids and mobility reduce delirium in older hospital patients	Inouye et al. 1999	yes	partly: 852 general-medicine patients aged 70+, not surgical	supported with caveat	adapt: name the population
B2 Daily geriatrician visits reduced delirium from 50% to 32% in hip fracture patients	Marcantonio et al. 2001	yes	yes: 20/62 vs 32/64 patients, one centre, 126 patients	supported	adopt in plain language
B3 Lavender aromatherapy halves the risk of delirium after hip surgery	Hartmann and Weiss 2021	no	not assessable	fabricated reference	reject

The melatonin claim in answer A is the instructive case. The cited trial exists and is exactly on topic: older patients with hip fracture, melatonin or placebo, delirium as primary outcome. Its result was that melatonin did not reduce the incidence of delirium — 29.6% with melatonin versus 25.5% with placebo (de Jonghe et al. 2014). The generated sentence turns this result into its opposite. Nothing in this misuse is the trial authors’ doing; the error lies entirely in the generated sentence. The lavender claim in answer B cites a paper and a journal that could not be found in Crossref or the NLM Catalog, and its DOI does not resolve; it was invented for this book to show what a

fabricated reference looks like. The two remaining claims of answer B are supported, one of them with a caveat about the population studied.

The most important assumption for using generated text is that a generated sentence is a hypothesis about the literature, not a report of it; it becomes usable only after its source has been found and read. In clinical communication, a further condition applies: the person who signs the letter is responsible for every sentence in it, whatever produced the first draft. Benchmarks of clinical knowledge show that models can answer many examination-style questions while human evaluation still finds important gaps (Singhal et al. 2023); such results describe average behaviour, not the reliability of the sentence in front of you.

Question for the workshop: For each of the five claims in Table 8.1, decide whether you would adopt, adapt or reject it in the reply to the daughter. Is there a claim you would reject even though its source is real and supportive — and is rejecting the whole generated text a defensible choice?

8.3 Lab

8.3.1 Step 1 · Store the two answers with their provenance

The two answers below are *synthetic*. They were written for this book to reproduce typical patterns of generated text; they are not output of any language model, and the “settings” describe a fictitious assistant. Each answer is stored with a SHA-256 checksum so that later readers can confirm that the text they assess is the text that was assessed.

```
library(digest)
question <- paste("My mother is 78 and will have surgery for a hip fracture tomorrow.",
                  "Can anything be done so that she does not get confused after the
        operation?",
                  "Should she take melatonin?")
answer_a <- paste(
  "Confusion after surgery, called delirium, is common in older adults with a hip
        fracture.",
  "The most effective step is melatonin: a randomised trial in hip fracture patients showed
        that",
  "melatonin reliably prevents postoperative delirium (de Jonghe et al., CMAJ 2014).",
  "Your mother should take melatonin every evening, starting the evening before surgery.",
  "Bringing her glasses and hearing aids and having family visit can also help her stay
        oriented.")
answer_b <- paste(
  "Confusion after an operation, called delirium, affects many older patients.",
  "Measures that address orientation, sleep, vision, hearing, fluids and mobility have been
        shown",
  "to reduce delirium in older hospital patients (Inouye et al., NEJM 1999).",
  "In a randomised trial in hip fracture patients, daily visits by a geriatrician reduced
        delirium",
  "from 50% to 32% (Marcantonio et al., J Am Geriatr Soc 2001).",
  "A 2021 study also found that aromatherapy with lavender oil halves the risk of delirium
        after hip",
  "surgery (Hartmann and Weiss, Journal of Orthogeriatric Nursing 2021;
        doi:10.5555/jon.2021.0147).",
  "Please ask the ward team which of these measures they offer.")

stored <- data.frame(
  answer = c("A", "B"),
```

```
  setting_described = c("fictitious assistant, temperature 0.2", "fictitious assistant,
        temperature 1.0"),
  question = question,
  text = c(answer_a, answer_b),
  data_status = "synthetic; written for this book; not produced by a language model")
stored$sha256 <- vapply(stored$text, digest, character(1), algo = "sha256", serialize =
        FALSE)
write.csv(stored, "../data/b08-stored-answers.csv", row.names = FALSE)
stopifnot(nrow(read.csv("../data/b08-stored-answers.csv")) == 2,
          all(nchar(stored$sha256) == 64))
substr(stored$sha256, 1, 16)        # first 16 characters of each checksum
```

```
#> [1] "79592eb3bc08717b" "06eb23553cd7c8ca"
```

8.3.2 Step 2 · A bigram model from twelve sentences

```
corpus <- c(
  "delirium after hip surgery is common in older patients",
  "geriatric visits help prevent delirium after hip surgery",
  "early mobilisation helps prevent delirium after hip surgery",
  "regular sleep helps prevent delirium in older patients",
  "melatonin does not prevent delirium after hip surgery",
  "melatonin helps some patients sleep",
  "melatonin helps older patients sleep",
  "glasses and hearing aids help patients stay oriented",
  "family visits help patients stay oriented",
  "delirium is often missed in older patients",
  "pain control helps prevent delirium after hip surgery",
  "older patients often sleep poorly in hospital")

# Tokens are words here; <s> and </s> mark the start and end of a sentence
bigram_table <- function(sentences) {
  pairs <- do.call(rbind, lapply(strsplit(sentences, " "), function(w) {
    w <- c("<s>", w, "</s>")
    data.frame(from = head(w, -1), to = tail(w, -1))
  }))
  table(pairs$from, pairs$to)                    # counts: rows = current word, columns =
        next word
}
counts <- bigram_table(corpus)

# Next-word probabilities, optionally reshaped by a temperature T: p^(1/T), renormalised.
# This equals dividing the log-probabilities ("logits") by T before the softmax.
next_word <- function(word, temperature = 1, cnt = counts) {
  k <- cnt[word, ]; k <- k[k > 0]
  p <- k / sum(k)
  q <- p^(1 / temperature)
  q / sum(q)
}
greedy <- function(start, cnt = counts, max_len = 15) {   # always take the most probable
        next word
  out <- start; w <- start
  repeat {
    p <- next_word(w, cnt = cnt); w <- names(p)[which.max(p)]
    if (w == "</s>" || length(out) >= max_len) break
    out <- c(out, w)
  }
  paste(out, collapse = " ")
}
greedy_sentence <- greedy("melatonin")
false_sentence <- "melatonin helps prevent delirium after hip surgery"
true_sentence <- "melatonin does not prevent delirium after hip surgery"
```

```
stopifnot(greedy_sentence == false_sentence, !(false_sentence %in% corpus), true_sentence
        %in% corpus)
round(next_word("melatonin"), 2); round(next_word("helps"), 2)
```

```
#>  does helps
#>  0.33  0.67

#>   older prevent    some
#>     0.2     0.6     0.2
```

The model's probability for the false sentence (start word given) is the product of its word-to-word probabilities:

```
sentence_prob <- function(s, cnt = counts) {
  w <- c(strsplit(s, " ")[[1]], "</s>")
  prod(mapply(function(a, b) next_word(a, cnt = cnt)[b], head(w, -1), tail(w, -1)))
}
probs <- c(false = sentence_prob(false_sentence), true = sentence_prob(true_sentence))
stopifnot(probs["false"] > probs["true"])
round(probs, 3)
```

```
#> false  true
#> 0.229 0.190
```

The false sentence is more probable than the true one, because "melatonin helps" and "helps prevent" are frequent pairs in the corpus. Nothing in the counts encodes which statement about melatonin and delirium is correct.

8.3.3 Step 3 · Temperature and generation

```
generate <- function(start, temperature, cnt = counts, max_len = 15) {
  out <- start; w <- start
  repeat {
    p <- next_word(w, temperature, cnt)
    w <- sample(names(p), 1, prob = p)
    if (w == "</s>" || length(out) >= max_len) break
    out <- c(out, w)
  }
  paste(out, collapse = " ")
}
set.seed(8081)
temperatures <- c(0.5, 1, 2)
runs <- lapply(temperatures, function(t) replicate(2000, generate("melatonin", t)))
temp_summary <- data.frame(
  temperature = temperatures,
  false_share = sapply(runs, function(s) mean(s == false_sentence)),
  true_share = sapply(runs, function(s) mean(s == true_sentence)),
  distinct = sapply(runs, function(s) length(unique(s))))
temp_summary$other_share <- 1 - temp_summary$false_share - temp_summary$true_share
stopifnot(temp_summary$false_share[1] > 0.5, temp_summary$false_share[3] <
        temp_summary$false_share[1],
          temp_summary$distinct[3] > temp_summary$distinct[1])
knitr::kable(temp_summary, digits = 3, row.names = FALSE,
             col.names = c("Temperature", "False sentence", "True sentence", "Distinct
        sentences", "Other"),
             caption = "2,000 sentences generated from “melatonin” at each temperature
        (synthetic).")
```

2,000 sentences generated from “melatonin” at each temperature (synthetic).

Temperature	False sentence	True sentence	Distinct sentences	Other
0.5	0.572	0.165	94	0.263
1.0	0.230	0.184	200	0.586
2.0	0.090	0.140	275	0.771

The following code draws Figure 8.1 in the core reading.

```
library(ggplot2)
plot_df <- data.frame(
  temperature = factor(rep(temp_summary$temperature, 3)),
  outcome = factor(rep(c("False sentence", "True sentence", "Other sentences"), each = 3),
                   levels = c("False sentence", "True sentence", "Other sentences")),
  share = c(temp_summary$false_share, temp_summary$true_share, temp_summary$other_share))
ggplot(plot_df, aes(x = temperature, y = share, fill = outcome)) +
  geom_col(position = position_dodge(width = 0.8), width = 0.75, colour = "#333333",
           linewidth = 0.2) +
  geom_text(aes(label = cdsr_percent(share, 0)), position = position_dodge(width = 0.8),
            vjust = -0.4, size = 3.2) +
  scale_fill_manual(values = c(`False sentence` = cdsr_colours[["red"]],
                               `True sentence` = cdsr_colours[["skyblue"]],
                               `Other sentences` = "#D9D9D9")) +
  scale_y_continuous(labels = function(x) cdsr_percent(x, 0), limits = c(0, 1)) +
  labs(x = "Temperature", y = "Share of generated sentences", fill = NULL,
       caption = "2,000 generated sentences per temperature; start word 'melatonin';
         synthetic corpus") +
  theme_cdsr()
```

8.3.4 Step 4 · The evidence card: existence and support, judged separately

Each claim receives two separate judgements. The verdict follows from their combination, and the action follows from the verdict. “Reject” is a legitimate action — for a single sentence or for the whole answer.

```
card <- data.frame(
  claim = c("A1 Melatonin reliably prevents postoperative delirium",
            "A2 Glasses, hearing aids and family visits help orientation",
            "B1 Measures for orientation, sleep, vision, hearing, fluids and mobility
         reduce delirium in older hospital patients",
            "B2 Daily geriatrician visits reduced delirium from 50% to 32% in hip fracture
         patients",
            "B3 Lavender aromatherapy halves the risk of delirium after hip surgery"),
  source = c("de Jonghe et al. 2014", "none", "Inouye et al. 1999", "Marcantonio et al.
         2001",
             "Hartmann and Weiss 2021"),
  exists = c("yes", "not applicable", "yes", "yes", "no"),
  supports = c("no: no reduction in incidence (29.6% vs 25.5%)",
               "not assessable without a source",
               "partly: 852 general-medicine patients aged 70+, not surgical",
               "yes: 20/62 vs 32/64 patients, one centre, 126 patients",
               "not assessable"))
verdict <- function(exists, supports) {
  ifelse(exists == "no", "fabricated reference",
  ifelse(exists == "not applicable", "unsourced",
  ifelse(startsWith(supports, "no"), "real source, contradicts claim",
  ifelse(startsWith(supports, "partly"), "supported with caveat", "supported"))))
```

```
}
card$verdict <- verdict(card$exists, card$supports)
action_for <- c("real source, contradicts claim" = "reject",
                "fabricated reference" = "reject",
                "unsourced" = "adapt only after finding a source",
                "supported with caveat" = "adapt: name the population",
                "supported" = "adopt in plain language")
card$action <- unname(action_for[card$verdict])          # the action follows from the
        verdict
stopifnot(card$verdict[1] == "real source, contradicts claim",
          card$verdict[5] == "fabricated reference",
          sum(card$action == "reject") == 2)
```

The following code prints Table 8.1 in the core reading.

```
knitr::kable(card[, c("claim", "source", "exists", "supports", "verdict", "action")],
             col.names = c("Claim (answer)", "Source given", "Source exists?", "Source
        supports claim?",
                           "Verdict", "Action in the reply"), row.names = FALSE)
```

The two judgements in the card rest on checks that every reader can repeat. Existence: the DOIs of the three real studies were resolved through Crossref and their abstracts read in PubMed (de Jonghe et al. 2014; Inouye et al. 1999; Marcantonio et al. 2001); the DOI given for the lavender study returns “not found” at doi.org and Crossref, and neither the paper nor a journal of that name was found in Crossref or the NLM Catalog (checked 8 November 2026). Support: the numbers in the card come from the abstracts’ results sections. Tools for the existence check at scale — DOI lookups, retraction databases — are covered in Scientometrics in R: Retraction Analysis; the support check cannot be automated away, because it requires reading.

8.3.5 Step 5 · Check the patient letter for readability

The resident’s first draft is accurate but written for colleagues. The revision keeps the supported content, removes the rejected claims, offers a choice and ends with a question that checks understanding (teach-back, Chapter 21). Both letters are *synthetic*; their clinical content would still have to be confirmed against the local protocol (*open*).

```
draft <- paste(
  "Dear [patient's name], postoperative delirium is an acute neuropsychiatric syndrome
        characterised by",
  "fluctuating disturbances of attention and cognition, with a considerable incidence
        following hip fracture",
  "surgery in geriatric populations. Pharmacological prophylaxis with melatonin has not
        demonstrated efficacy",
  "in a randomised controlled trial in this population. Multicomponent non-pharmacological
        interventions",
  "addressing orientation, sensory impairment, sleep, hydration and mobilisation are
        associated with a",
  "reduced incidence in hospitalised older patients, and proactive geriatric consultation
        reduced the",
  "incidence in a randomised trial in patients with hip fracture.")
revised <- paste(
  "Dear [patient's name], after a hip operation some older people become confused for a few
        days.",
  "Doctors call this delirium. It usually gets better, but it can slow down recovery.",
  "In a study of almost four hundred older people with a broken hip, melatonin did not
        prevent it.",
```

```
  "Simple things can help. Please bring your glasses and hearing aids. Ask your family to
       visit.",
  "Our team can also ask a doctor for older people to see you every day. You can say yes or
       no to this.",
  "Could you tell us in your own words what you will bring and whom you will call if you
       feel confused?")

count_syllables <- function(word) {                # approximation: groups of vowels,
       silent final e
  w <- tolower(gsub("[^A-Za-z]", "", word))
  if (nchar(w) == 0) return(0)
  n <- lengths(regmatches(w, gregexpr("[aeiouy]+", w)))
  if (grepl("[^l]e$", w) && n > 1) n <- n - 1
  max(1, n)
}
readability <- function(text) {
  sentences <- unlist(strsplit(text, "(?<=[.?!])\\s+", perl = TRUE))
  words <- unlist(strsplit(gsub("[^A-Za-z' -]", " ", text), "\\s+")); words <- words[words
       != ""]
  syl <- vapply(words, count_syllables, numeric(1))
  c(sentences = length(sentences), words = length(words),
    words_per_sentence = length(words) / length(sentences),
    syllables_per_word = mean(syl),
    long_words = mean(syl >= 3),
    flesch = 206.835 - 1.015 * length(words) / length(sentences) - 84.6 * mean(syl))  #
       Flesch (1948)
}
jargon <- c("postoperative", "neuropsychiatric", "incidence", "pharmacological",
       "prophylaxis",
           "efficacy", "multicomponent", "geriatric", "hospitalised", "mobilisation")
jargon_hits <- function(text) sum(sapply(jargon, function(j) grepl(j, tolower(text), fixed
       = TRUE)))
letters_tab <- rbind(draft = c(readability(draft), jargon_terms = jargon_hits(draft)),
                     revised = c(readability(revised), jargon_terms =
       jargon_hits(revised)))
stopifnot(letters_tab["revised", "flesch"] > letters_tab["draft", "flesch"] + 30,
          letters_tab["revised", "jargon_terms"] < letters_tab["draft", "jargon_terms"])
round(letters_tab, 2)
```

```
#>         sentences words words_per_sentence syllables_per_word long_words flesch
#> draft           3    82              27.33                2.5       0.49 -32.41
#> revised        10   108              10.80                1.4       0.08  77.59
#>         jargon_terms
#> draft             10
#> revised            0
```

The Flesch Reading Ease score rises from -32 to 78 (higher means easier; the formula combines sentence length and word length) (Flesch 1948); scores below zero occur for text with very long sentences and long words. The syllable count is an approximation, and no formula shows whether a reader has understood the letter. That is what the final question in the revised letter is for: the reply of the patient or her daughter, in their own words, is the real test.

8.3.6 Step 6 · Document origin, model version and use

The reply is clinical documentation; the generated drafts are part of its origin. A short record, stored with the case, makes the process reviewable. For research, the reporting guideline TRIPOD-LLM covers studies that evaluate or use large language models (Gallifant et al. 2025), and medical journals following the ICMJE recommendations expect authors to disclose at submission whether

and how AI-assisted technologies were used; the humans remain responsible for the content (International Committee of Medical Journal Editors 2026).

```
library(jsonlite)
record <- list(
  task = "draft reply to a patient-portal question on delirium prevention",
  tool = "fictitious hospital writing assistant (teaching case; in real use: product, model
       identifier, version)",
  deployment = "in real use: local, institutional or external service; whether personal
       data left the hospital",
  date_of_generation = "not applicable (synthetic answers written for this book)",
  input = question,
  personal_data_in_input = "none (age and diagnosis only, no identifiers)",
  settings = stored$setting_described,
  stored_output_sha256 = stored$sha256,
  sentences_rejected = card$claim[card$action == "reject"],
  sentences_adapted = card$claim[startsWith(card$action, "adapt")],
  sources_checked = c("10.1503/cmaj.140495", "10.1056/NEJM199903043400901",
                      "10.1046/j.1532-5415.2001.49108.x", "10.5555/jon.2021.0147 (not
       found)"),
  checked_by = "resident on duty; countersigned by consultant",
  disclosure = "noted in the case record; the letter is signed by the responsible
       physician")
stopifnot(length(record$sentences_rejected) == 2)
cat(toJSON(record, pretty = TRUE, auto_unbox = TRUE))
```

```
#> {
#>   "task": "draft reply to a patient-portal question on delirium prevention",
#>   "tool": "fictitious hospital writing assistant (teaching case; in real use: product,
model identifier, version)",
#>   "deployment": "in real use: local, institutional or external service; whether personal
data left the hospital",
#>   "date_of_generation": "not applicable (synthetic answers written for this book)",
#>   "input": "My mother is 78 and will have surgery for a hip fracture tomorrow. Can
anything be done so that she does not get confused after the operation? Should she take
melatonin?",
#>   "personal_data_in_input": "none (age and diagnosis only, no identifiers)",
#>   "settings": ["fictitious assistant, temperature 0.2", "fictitious assistant, temperature
1.0"],
#>   "stored_output_sha256":
["79592eb3bc08717bbcb61a29408416d98b76bcb0a1478766096b62be1b24961c",
"06eb23553cd7c8ca16f1ebc4880d0b7e84e2fc71d652d0cb906afdc49400fba9"],
#>   "sentences_rejected": ["A1 Melatonin reliably prevents postoperative delirium", "B3
Lavender aromatherapy halves the risk of delirium after hip surgery"],
#>   "sentences_adapted": ["A2 Glasses, hearing aids and family visits help orientation", "B1
Measures for orientation, sleep, vision, hearing, fluids and mobility reduce delirium in
older hospital patients"],
#>   "sources_checked": ["10.1503/cmaj.140495", "10.1056/NEJM199903043400901",
"10.1046/j.1532-5415.2001.49108.x", "10.5555/jon.2021.0147 (not found)"],
#>   "checked_by": "resident on duty; countersigned by consultant",
#>   "disclosure": "noted in the case record; the letter is signed by the responsible
physician"
#> }
```

8.4 Counter-check

Change the evidence the model can draw on. Retrieval-augmented systems add documents to what the model sees before it writes (Lewis et al. 2020). Our bigram analogue: add two sentences

that restate the trial result to the corpus — “melatonin does not prevent delirium in older patients” and “melatonin does not reduce delirium after hip surgery”.

Before you run the code, predict: What will the greedy sentence be? At temperature 1, will the false sentence disappear?

```
corpus_retrieved <- c(corpus, "melatonin does not prevent delirium in older patients",
                      "melatonin does not reduce delirium after hip surgery")
counts_retrieved <- bigram_table(corpus_retrieved)
greedy_retrieved <- greedy("melatonin", cnt = counts_retrieved)
set.seed(8082)
gen_retrieved <- replicate(2000, generate("melatonin", 1, cnt = counts_retrieved))
share_false_retrieved <- mean(gen_retrieved == false_sentence)
stopifnot(greedy_retrieved == true_sentence, share_false_retrieved > 0,
          share_false_retrieved < temp_summary$false_share[2])
c(greedy = greedy_retrieved, false_share_t1 = round(share_false_retrieved, 3))
```

```
#>                                                 greedy
#> "melatonin does not prevent delirium after hip surgery"
#>                                          false_share_t1
#>                                                  "0.13"
```

With the added evidence, the most probable sentence becomes the true one. Yet at temperature 1 the false sentence is still generated in 13% of runs, compared with 23% before. Retrieval shifts the probabilities; it does not remove the recombination that produced the error. The check of every sentence against its source therefore remains necessary even with a retrieval system — a point that holds, with far more complex models, in practice.

8.5 Consequences for patients and operations

Patients. The harm of answer A would have been concrete: a family relying on melatonin, possibly neglecting measures with better support. The revised letter also changes the relationship: it offers a choice and asks for feedback instead of instructing. Patients are entitled to know when text addressed to them was drafted with a language model and who checked it; how this is communicated is part of the hospital’s policy (Chapter 21). The WHO guidance on the ethics and governance of large multi-modal models in health is a starting point for these questions (World Health Organization 2024).

Data protection. The question as entered contained age and diagnosis but no name. Entering identifiable patient data into an external service is a processing of health data — a special category under Article 9 of the General Data Protection Regulation (GDPR) (European Parliament and Council of the European Union 2016) (English text on EUR-Lex). It needs a legal basis, a processor agreement under Article 28 where a provider processes data on the hospital’s behalf, and an assessment under Chapter V if data leave the European Economic Area. Which services are approved is a local decision (*open*; Chapter 23).

Security. Text that a model reads can contain instructions. A retrieved document or an attached file can steer the output — indirect prompt injection — and an assistant with access to records can leak them. These risks and their controls are treated in Chapter 25.

Regulation. Under Article 50 of the AI Act (European Parliament and Council of the European Union 2024), applicable from 2 August 2026, providers must ensure that people are informed when they interact directly with an AI system, unless this is obvious, and must mark generated text and other synthetic content in a machine-readable way; for systems already on the market before that date, the marking obligation applies from 2 December 2026 (Article 111(4), inserted by Regulation (EU) 2026/1744 (European Parliament and Council of the European Union 2026)). Deployers who publish generated text to inform the public on matters of public interest must disclose this unless the text has undergone human review and someone holds editorial responsibility. Article 4, as replaced by the 2026 amendment, requires providers and deployers to take measures to support the AI literacy of their staff. (Read on EUR-Lex; status: November 2026.) Whether an assistant that answers patient-specific medical questions is itself a medical device depends on its intended purpose (*open*; Chapter 22).

8.6 Look up and transfer

Reviewer question: “For every reference in the manuscript, has someone confirmed both that it exists and that the cited passage supports the sentence it is attached to — and do the authors disclose which AI-assisted tools were used, for what, and with which version?”

Self-test: A generated discussion paragraph for a manuscript reads: “Deep learning has been shown to outperform clinicians in image-based diagnosis [Nagendran et al., BMJ 2020].” The DOI resolves and the paper is a systematic review of studies comparing deep learning with clinicians. Would you keep the sentence?

Answer

Annotated sources:

- Jurafsky and Martin (2026) explains n-gram language models (chapter 3) and temperature sampling (section 7.6.3), the two mechanisms of the lab.
- Vaswani et al. (2017) introduces the transformer architecture on which current language models are built.
- Holtzman et al. (2019) shows how decoding choices change the quality and diversity of generated text.
- Ji et al. (2023) surveys hallucination, the production of fluent but unsupported content.
- Walters and Wilder (2023) quantifies fabricated and erroneous references produced by a chatbot.
- Lewis et al. (2020) describes retrieval-augmented generation, the idea behind the counter-check.
- Singhal et al. (2023) evaluates clinical knowledge of large language models and finds gaps on human evaluation.
- de Jonghe et al. (2014), Inouye et al. (1999) and Marcantonio et al. (2001) are the three real studies checked in the evidence card; the first is the real source that does not support the claim attached to it.
- Flesch (1948) defines the Reading Ease formula used for the patient letter.

- Gallifant et al. (2025) and International Committee of Medical Journal Editors (2026) set the reporting and disclosure expectations for AI use.
- The AI Act (European Parliament and Council of the European Union 2024), its 2026 amendment (European Parliament and Council of the European Union 2026) and the GDPR (European Parliament and Council of the European Union 2016) are the binding texts behind the consequences; the WHO guidance (World Health Organization 2024) is orientation.

8.7 Further reading

- Chapter 7 — how to trace a statement from a brief or brochure back to the original study.
- Chapter 25 — prompt injection, data leakage and other security risks of generative systems.
- Chapter 21 — teach-back and shared decisions when explaining uncertainty to patients.
- Scientometrics in R: Text Mining Bibliographic Corpora — tokens, n-grams and word counts on real literature corpora.
- Scientometrics in R: Embeddings for Scientometrics — the vector representations that retrieval systems use to find documents.
- CTTIR Tutorials: Transformers: Overview — the architecture behind current language models.
- Strategy in R: LLM Agents and Strategic Reasoning — what changes when language models act rather than only write.
- Methods in R: Use of LLM tools — an example of a disclosure statement that names tools, tasks and the checks applied.