

# Fuzzy Variable Tightening

The correction pass that cuts every AI-written value down to size and forces it to fit the exact sentence it has to live inside.

---

## 01 When to use this

Use this immediately after your first enrichment run, every single time. The first pass is always too wordy and at least one value will be structurally broken. This is the note that teaches the actual craft, because the difference between personalization that works and personalization that gets you marked as a bot is decided here.

## 02 The prompt

The fuzzy variables you just wrote are too long, and some of them do not fit the sentence they have to sit inside. Rewrite all of them against the rules below and write the result to a new sheet.

Here is the exact copy each variable has to live in. Write for insertion at that point in the sentence, not as a standalone statement about the person:

"Extremely excited to have you with us, especially because {{shortPlausibleReasonWhySignedUp}}."

"I think <PRODUCT\_NAME> is going to be useful for you because {{shortCustomReasonWhyItsUseful}}."

"I know a big challenge for you right now is {{paraphrasedChallenge}}."

Rules:

1. Length. A reason slot is about 10 words. A paraphrase slot is about 5 words. Shorter is better than longer every time.
2. Socket fit. Read the value back inside the full sentence before you accept it. It must be grammatically correct and it must not repeat words that already appear in my copy. Never start a value by naming the person or restating the sentence stem.
3. Second person. Write to them as you, not about them in the third person.
4. No em dashes anywhere. They read as AI.
5. No marketing language, no adjectives you would not say out loud.

Example of a broken value: "Marcus's biggest challenge is that leads show up in bursts". That is a report about Marcus, so it collapses the sentence.

Example of a fixed value: "keeping up with how fast AI moves".

Output the new sheet with each variable shown twice: bare, and rendered inside its full sentence, so I can check fit at a glance.

## 03 What it is actually doing

Two things go wrong on a first enrichment pass, and they are different problems with different fixes.

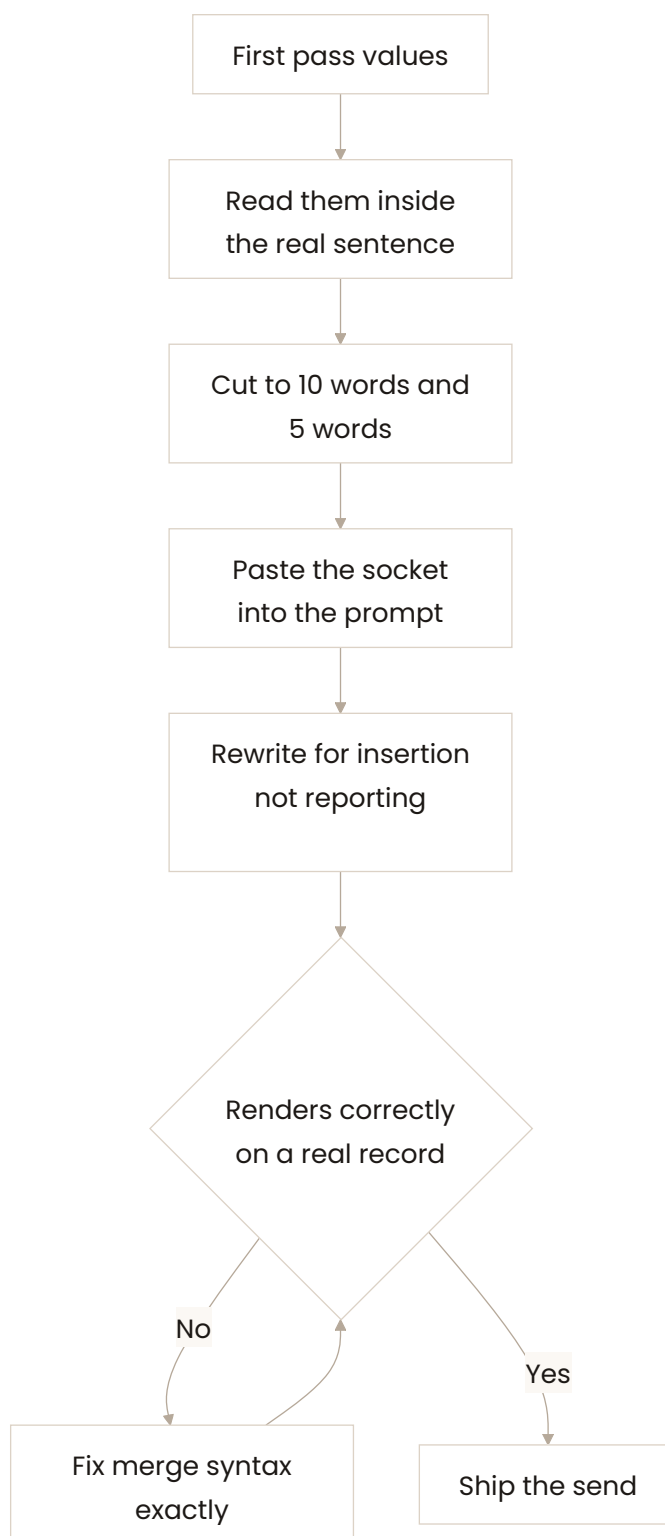
The first is length. Claude writes fifteen or twenty words when ten would do. This matters more than it sounds. The share of the email that is AI-written is what triggers a reader's suspicion. Fewer AI-written words as a share of the total means a lower chance anyone thinks a bot wrote it. So you cut hard: about ten words for a reason, about five for a paraphrase.

The second is fit. Nick's first run produced the value "Marcus's biggest challenge is that leads show up in bursts". Read it inside the sentence it was built for: "I know a big challenge for you is Marcus's biggest challenge is that leads show up in bursts". It is broken. The model wrote a report about the person instead of a fragment for a hole.

The fix is one line of prompt. Paste the actual surrounding sentence from your email and tell the model it is writing for insertion at that exact point. Once the model can see the socket, it writes for the socket.

Then you verify against a real send. Nick previewed the email against a live subscriber and found the platform's merge-field syntax was not what he assumed, so every variable rendered blank. The values were perfect and the email was empty. Preview before you trust it.

## 04 How it flows



---

## 05 Step by step

---

### 01 Read the first pass rendered

Paste a few values into the real email and read the whole sentence out loud. Problems that are invisible in a spreadsheet column are obvious in a sentence.

---

### 02 Set a word budget per slot

About 10 words for a reason slot, about 5 for a paraphrase slot. State the number in the prompt. Vague words like short get you fifteen words back.

---

### 03 Paste the socket into the prompt

Give the model the full sentence with the placeholder still in it. This is the single change that fixes structurally broken values.

---

### 04 Ban third person and restating

Tell it to write in second person and never to begin by naming the person or repeating the sentence stem. That is what caused the Marcus failure.

---

### 05 Ban em dashes explicitly

Em dashes are one of the loudest AI tells in cold and warm email right now. Say no em dashes in the prompt, because the model defaults to them.

---

### 06 Ask for a rendered column

Have it output each value twice, bare and inside its full sentence. You can then scan a hundred rows for fit in under a minute.

---

### 07 Preview against a real subscriber

Send or preview the actual email to one real record. Confirm the variables render. Nick found the platform needed exact lowercase merge syntax that differed from what he assumed.

---

---

## 06 Words explained

---

### Socket fit

Whether the AI-written fragment is grammatically correct inside the sentence you built around it. A good phrase in the wrong socket still breaks the email.

---

### Merge syntax

The exact bracket format a sending platform expects for a placeholder. Different platforms use different formats and they are usually case sensitive.

---

### AI tell

A pattern readers now associate with machine writing. Em dashes, flawless grammar, and words like leverage are the current ones.

---

### Second person

Writing to the reader as you, rather than about them as he or she. It keeps the fragment inside the flow of your sentence.

---

### Correction pass

A second run over output you already have, aimed only at fixing it. Cheaper and more reliable than regenerating from scratch.

---

---

## 07 Watch out for

---

- The model writes a report about the person unless you show it the surrounding sentence. That is the number one failure.
  - Saying keep it short does nothing. Give a word count per slot or you get long values back.
  - Em dashes appear by default in AI copy and readers now clock them instantly. Ban them by name.
-

- Perfect values render blank if the merge syntax is wrong. Nick lost a preview to a case-sensitive format mismatch.
- Never judge fit from the spreadsheet column. Always judge it inside the sentence it has to fill.

---

## 08 How it connects

---

### Newsletter Fuzzy Variable Enrichment

This is the second half of that build. The enrichment prompt produces the values and this prompt makes them usable.

---

### Cold Email Personalization

The same length and socket-fit rules decide whether a cold email reads human, where the stakes are higher because a bad send burns a domain.

---

### Follow-Up Cadence SOP

The no em dashes rule and the preference for short human-shaped lines carry straight into the follow-up template pool.

---

### Fuzzy Variables for Personalized Copy

The framework note where the ten-word and five-word budgets and the write-for-insertion rule are recorded.

---

---

**Credit.** Framework and approach by Nick Saraev, from the fuzzy variable correction pass in the copy personalization build of his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.