Avoid ambiguity in language

Last modified on 11/25/2026 3:01 pm EST

In 2026, CT Smith performed a number of tests to see how common AI tools handled some of the tried-and-true writing rules about passive voice and avoiding ambiguity. She documented her findings and made her testing suite available in the Github luciddocs repo.

Her biggest surprise was that AI tools generally handle passive voice fairly well.

But she wasn't surprised to learn that they didn't handle ambiguity well. The cases and tips below are a summary of her findings.

Avoid scope hedging

Scope hedging is when you wrap clear instructions in qualifiers like "generally," "where applicable," "in most cases," or "depending on your configuration. We generally do it to cover our butts in case we're missing an edge case or variable. (See what I did there? I hedged my scope!)

When you hedge a scope, AI basically perpetuates that vagueness, including it in every response. In CT's testing, this degraded AI accuracy by 46%.

Remove those vague scope hedges in favor of specific edge cases or exceptions. If you don't have a specific edge case or exception and were doing it just out of caution, stop. Write it specifically without the hedge and wait for the support tickets or complaints that will help surface the edge cases you need to document.

You may want to write the statement as it generally applies and then include an alert div style or a summary/details block to go into details on the exception.

Avoid vague quantifiers

Quantifier vagueness is when you use an approximate or vague value instead of a specific one, such as "periodically" instead of "every 90 days."

When you use a vague quantifier, AI either repeats the vague language or invents a specific value that doesn't exist. In CT's testing, this degraded AI accuracy by 44%

Replace vague or approximate values with specific values. If it’s a value that may vary, provide an example, for example: “Rotate your API keys every 90 days or based on the guidance from your security team.”

Define your terms

Copyright © 2015 – 2025 Silly Moose, LLC. All rights reserved.
Powered by KnowledgeOwl knowledge base software.

Undefined terms include acronyms, abbreviations, or internal terms that have no context for their meaning on the page. For example, SSO can mean "Single Sign-On," "Sorry Sold Out," or "Sydney Symphony Orchestra."

When you don't define a term or acronym and it has a specific meaning in your documentation, AI may parrot the abbreviation or acronym back, refuse to answer because it couldn't decode the terms, or invent its own definition for the term. In CT's testing, this degraded AI accuracy by 35%

Use glossary terms or other methods to define acronyms or abbreviations. Be sure you redefine the acronym or abbreviation, even if it's the old-school parenthetical format the first time you use it, for example: "Your knowledge base (KB) houses all your documentation."

For RAG/chunking, you may want to define or manually insert the glossary term in each subsequent section.

Refer to Add glossary definitions in articles manually for more detailed instructions on adding glossary terms and definitions this way.

Remove ambiguous pronouns

Ambiguous pronouns are when you have words like "it" or "this" that may refer to different things on the same page or in the same section. Human readers usually use context from the whole page to figure out what "it" means, but an AI tool looking at a single chunk may not have the appropriate context to figure it out.

When you use an ambiguous pronoun, AI will either confidently answer with the wrong "it" or will parrot the vagueness. In CT's testing, this degraded AI accuracy by 31%

Make sure your pronouns clearly map to a concept in each section. If necessary, replace the vague pronoun with something more specific, either calling the thing by its full name or by a more specific reference, like "This generic setup process..." instead of "This...".

Copyright © 2015 – 2025 Silly Moose, LLC. All rights reserved.
Powered by KnowledgeOwl knowledge base software.