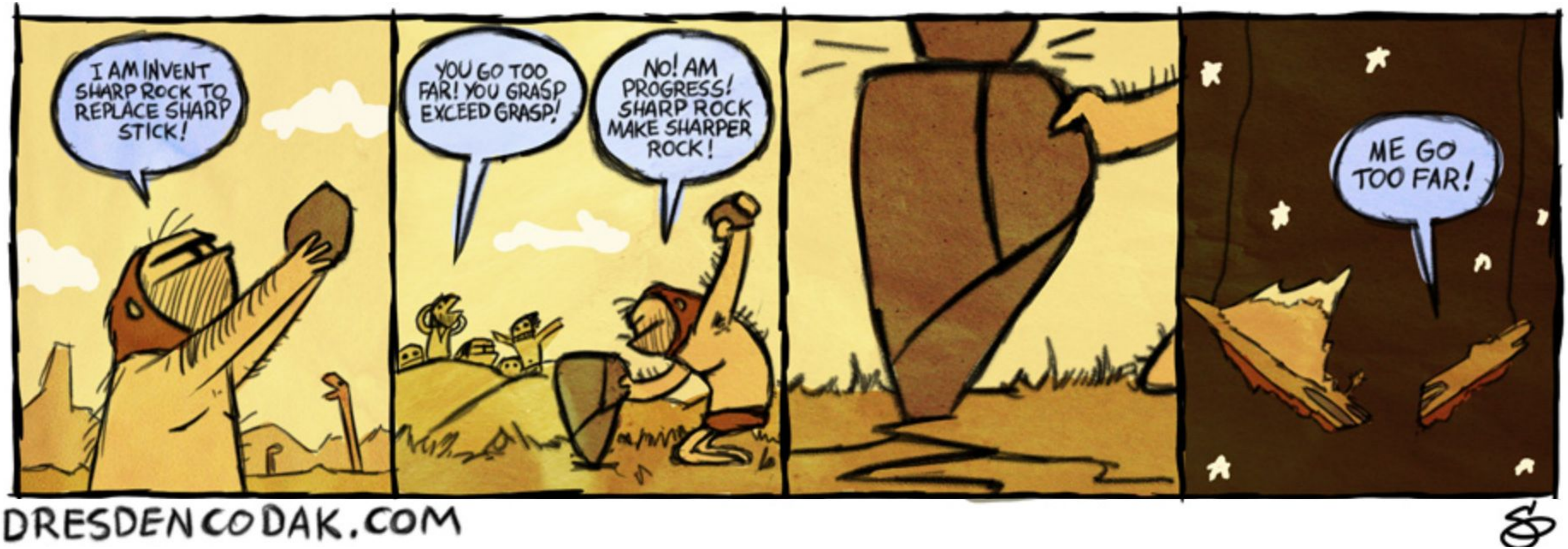
LLMs and other language tools in technical writing

The fundamentals of technical writing | BUT 2024

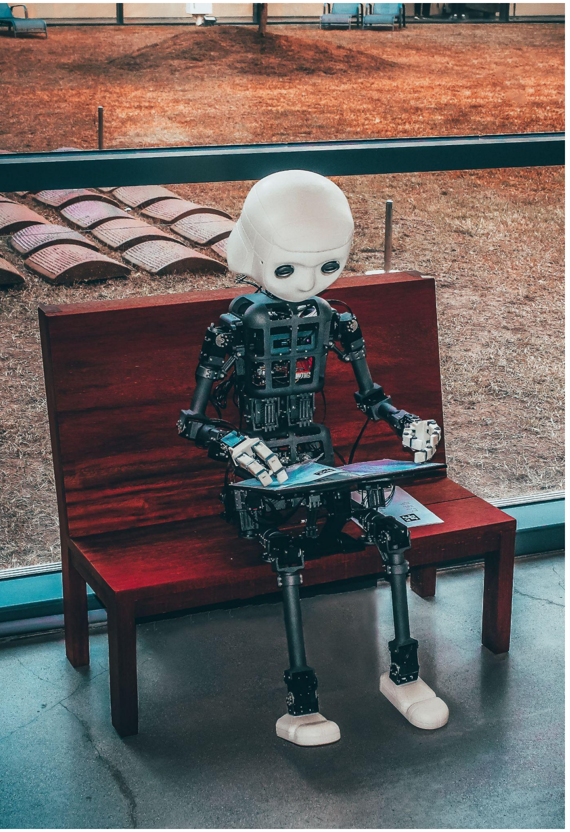
The Red Hat Customer Content Services team

What we’ll discuss today

- Writing aids
- How do LLMs work?
- How to use LLMs and other tools in technical writing
- Risks and shortcomings
- Practical exercises


I AM INVENT SHARP ROCK TO REPLACE SHARP STICK!
YOU GO TOO FAR! YOU GRASP EXCEED GRASP!
NO! AM PROGRESS! SHARP ROCK MAKE SHARPER ROCK!
ME GO TOO FAR!
DRESDENCODAK.COM

Source: https://dresdencodak.com/2009/09/22/caveman-science-fiction/



Which writing aids do you use?

Source: https://unsplash.com/@santesson89

Writing aids

Linters

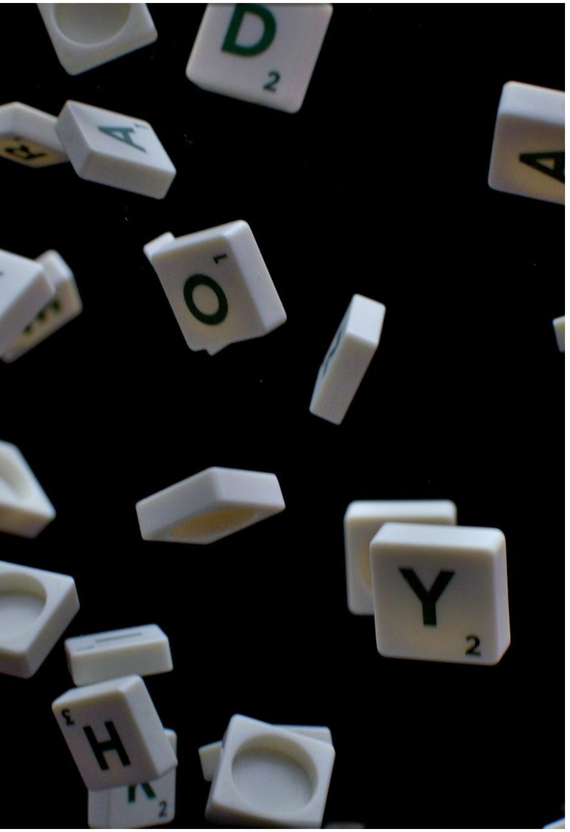
- Vale

Other tools

- Spellcheck
- Grammarly
- Hemingwayapp

Large language models

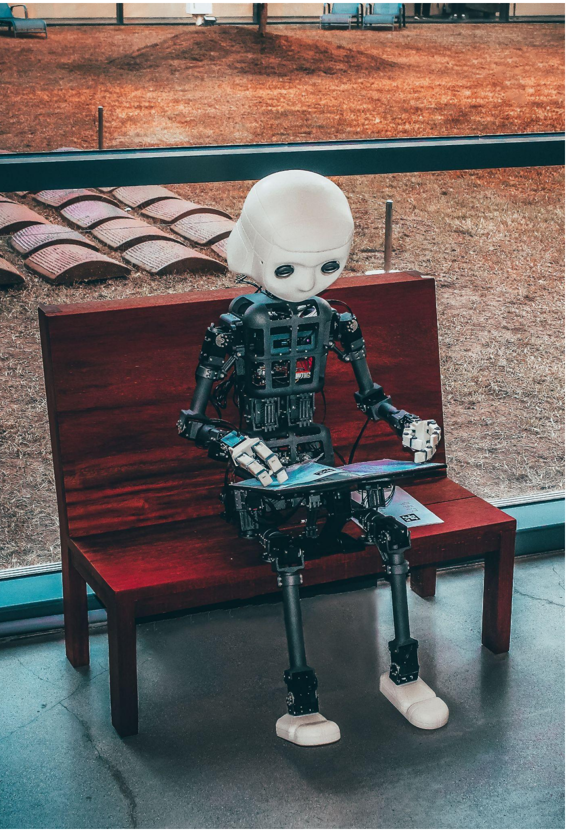
- GPT-3.5, 4, 5?
- ~~Bard~~ Gemini
- LLaMA 2
- Alpaca
- ~~Bing AI~~ Copilot



How do large language models work?

- LLMs use the **transformer architecture** for sequence processing.
- Transformers use **self-attention** to weigh word importance efficiently.
- Models are **pre-trained** on vast text data (billions to trillions of tokens).
- Then, they are **fine-tuned** for specific tasks.
- They have billions of **parameters** learned during training.
- They iteratively **predict the next token** based on context.

Source: https://unsplash.com/@tot87



How do you use AI tools?

Source: https://unsplash.com/@santesson89

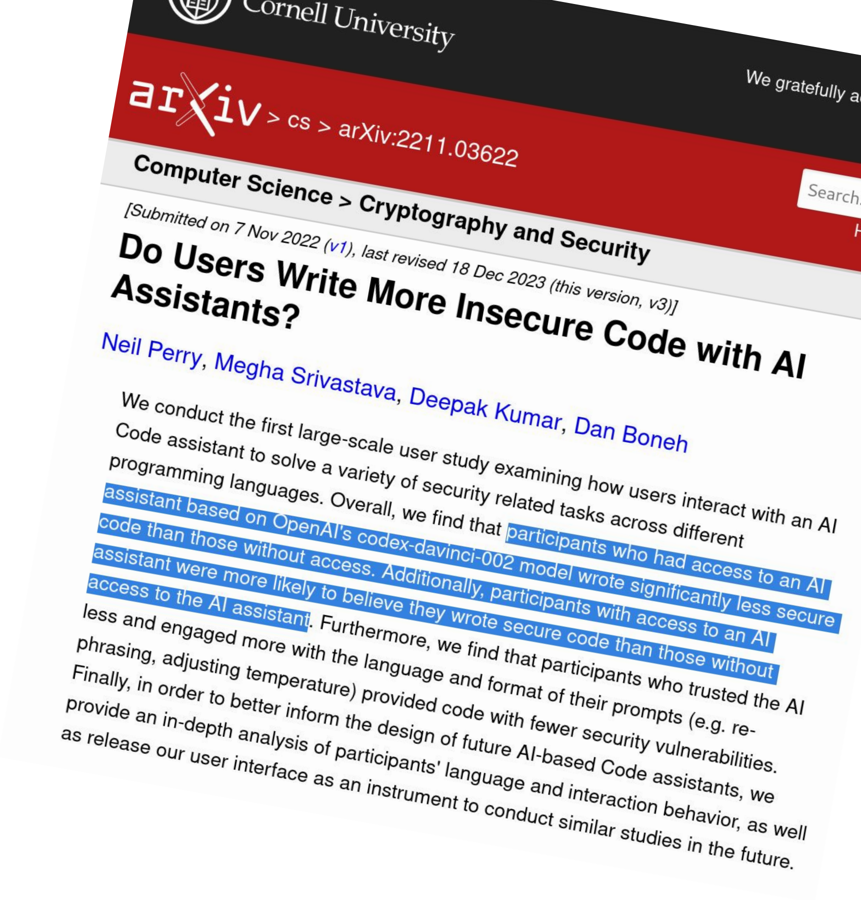
LLM use cases in TW

- Brainstorming: “Give me 5 headings for this paragraph”
- Research, explanations: “How does SELinux work?”
- Description of code: “What does this yaml file do?”
- Code snippets or regex: “Create a script that…”
- Formatting: “Convert this csv into a markdown table”
- Ideas for visual elements (runwayml.com)

Source: unsplash.com/@rechaoktaviani

Limitations of LLMs

- No real time information
- No fact checking
- Not for generating conclusions
- No information about indexed data
- Legal limitations
- Security concerns



Computer Science > Cryptography and Security

[Submitted on 7 Nov 2022 (v1), last revised 18 Dec 2023 (this version, v3)]

Do Users Write More Insecure Code with AI Assistants?

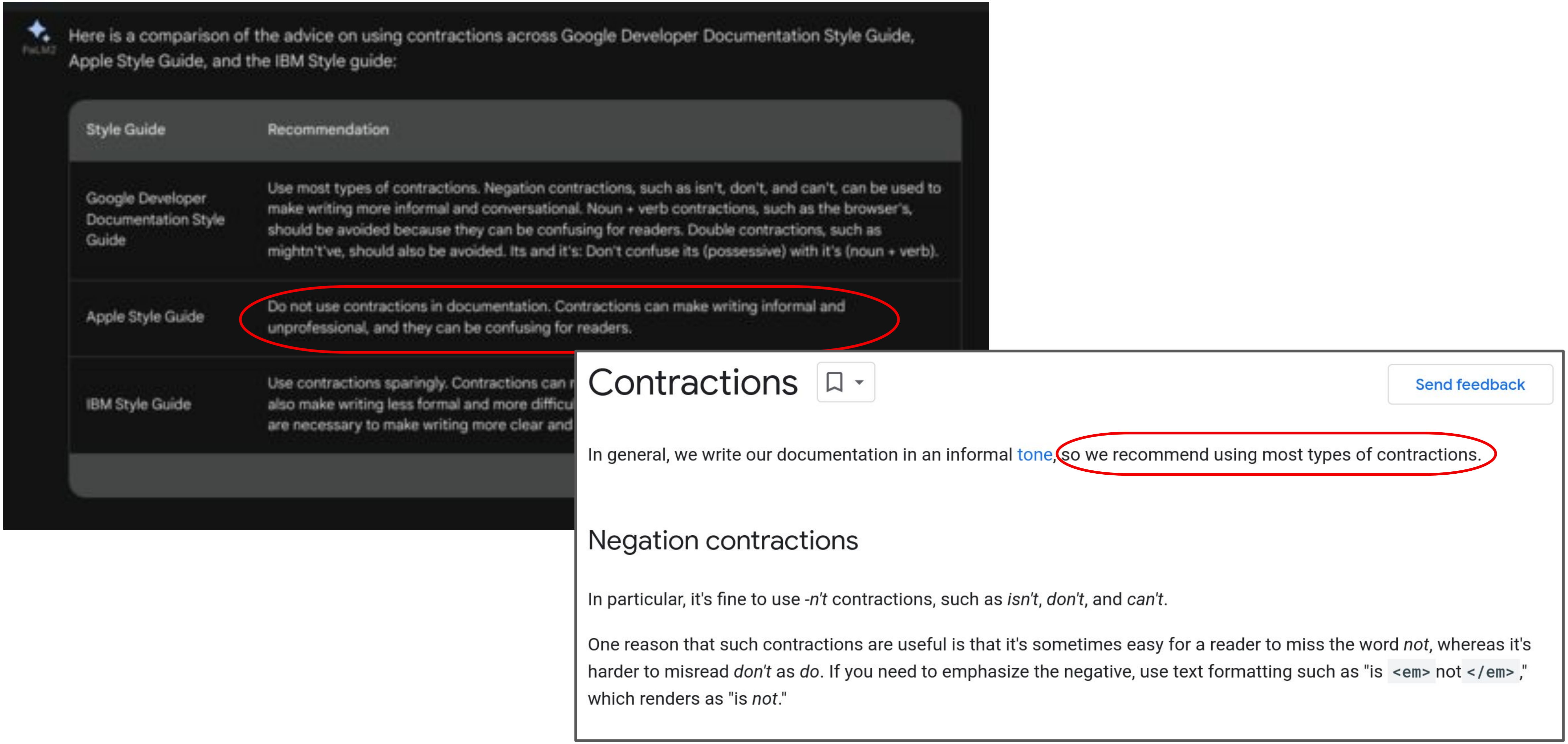
Neil Perry, Megha Srivastava, Deepak Kumar, Dan Boneh

We conduct the first large-scale user study examining how users interact with an AI Code assistant to solve a variety of security related tasks across different programming languages. Overall, we find that participants who had access to an AI assistant based on OpenAI's codex-davinci-002 model wrote significantly less secure code than those without access. Additionally, participants with access to an AI assistant were more likely to believe they wrote secure code than those without access to the AI assistant. Furthermore, we find that participants who trusted the AI less and engaged more with the language and format of their prompts (e.g. re-phrasing, adjusting temperature) provided code with fewer security vulnerabilities. Finally, in order to better inform the design of future AI-based Code assistants, we provide an in-depth analysis of participants' language and interaction behavior, as well as release our user interface as an instrument to conduct similar studies in the future.

Sources: unsplash.com/@jannerboy62
https://arxiv.org/abs/2211.03622
https://dl.acm.org/doi/10.1145/3491101.3519665

Example use case

Retrieving information about TW style?



Here is a comparison of the advice on using contractions across Google Developer Documentation Style Guide, Apple Style Guide, and the IBM Style guide:

Style Guide	Recommendation
Google Developer Documentation Style Guide	Use most types of contractions. Negation contractions, such as isn't, don't, and can't, can be used to make writing more informal and conversational. Noun + verb contractions, such as the browser's, should be avoided because they can be confusing for readers. Double contractions, such as mightn't've, should also be avoided. Its and it's: Don't confuse its (possessive) with it's (noun + verb).
Apple Style Guide	Do not use contractions in documentation. Contractions can make writing informal and unprofessional, and they can be confusing for readers.
IBM Style Guide	Use contractions sparingly. Contractions can r also make writing less formal and more difficu are necessary to make writing more clear and

Contractions

In general, we write our documentation in an informal tone, so we recommend using most types of contractions.

Negation contractions

In particular, it's fine to use *-n't* contractions, such as *isn't*, *don't*, and *can't*.

One reason that such contractions are useful is that it's sometimes easy for a reader to miss the word *not*, whereas it's harder to misread *don't* as *do*. If you need to emphasize the negative, use text formatting such as "is `<em>` not `</em>`," which renders as "is *not*."

Which text is written by a human TW?

A	B	C
Red Hat Enterprise Linux 9: GIMP 2.99.8 Technology Preview Available This release introduces GNU Image Manipulation Program (GIMP) version 2.99.8 as a Technology Preview in Red Hat Enterprise Linux 9. GIMP 2.99.8 delivers a comprehensive set of image editing and manipulation capabilities, enabling users to: ● **Perform basic image editing tasks:** Crop, resize, adjust color and brightness, apply filters, and more. ● **Create and manipulate artwork:** Draw vector graphics, paint with various brushes, and work with layers for complex compositions. ● **Automate repetitive tasks:** Utilize scripting capabilities to automate workflows and save time. **Important Note:** GIMP 2.99.8 is provided as a Technology Preview. This means it is still under development and may contain bugs or stability issues. It is not intended for production use. This Technology Preview allows users to explore the capabilities of GIMP 2.99.8 before its official introduction in Red Hat Enterprise Linux 9. Feedback from users will be crucial in shaping the final offering. **Installation:** GIMP 2.99.8 is available through the Technology Preview repository. You can install it using the following command: `sudo dnf install gimp-techpreview` **Disclaimer:** Red Hat does not provide support for Technology Preview features. Use them at your own risk and discretion.	**Feature: GNU Image Manipulation Program (GIMP) 2.99.8** **Description:** The GIMP package version 2.99.8 is now available as a Technology Preview in Red Hat Enterprise Linux 9. This pre-release version provides a glimpse into the upcoming GIMP 3 release. Users can explore and test this version using the RPM package for installation. **Key Points:** ● **Technology Preview Status:** ○ GIMP 2.99.8 is presented as a Technology Preview, showcasing features and enhancements that will be fully integrated into the official GIMP 3 release in future versions of RHEL 9. ● **Pre-Release Version:** ○ This package serves as a pre-release version, allowing users to experience and provide feedback on the latest developments in GIMP. **Recommended Action:** ● Users interested in previewing the upcoming features in GIMP 3 can install the GIMP 2.99.8 package. ● As a Technology Preview, this version is not intended for production use but offers users an opportunity to familiarize themselves with the upcoming GIMP release. **Note:** GIMP 2.99.8 is a forward-looking Technology Preview, giving users early access to the evolving capabilities of GIMP 3. Feedback from the community during this preview phase is valuable in refining and enhancing the user experience before the official release.	**GIMP available as a Technology Preview in RHEL 9** GNU Image Manipulation Program (GIMP) 2.99.8 is now available in RHEL 9 as a Technology Preview. The `gimp` package version 2.99.8 is a pre-release version with a set of improvements, but a limited set of features and no guarantee for stability. As soon as the official GIMP 3 is released, it will be introduced into RHEL 9 as an update of this pre-release version. In RHEL 9, you can install `gimp` easily as an RPM package.

Which text is written by a human TW?

A	B

Deprecated Functionality: Change in Default Settings for SSSD Local User Caching
Description: In Red Hat Enterprise Linux 9.4, a change in default settings has been made regarding the caching of local users by the System Security Services Daemon (SSSD). As part of this adjustment, SSSD will no longer cache local users by default.
Changes:
● **Default Settings Update:**
○ Local user caching, previously enabled by default, is deprecated in this release.
○ SSSD will no longer cache local users by default, impacting the default behavior of the system.
Impact:
● Users relying on the previous default behavior of local user caching may experience a change in SSSD's behavior.
● This deprecation aligns with evolving security and performance considerations.
Recommended Action:
● Administrators relying on local user caching in SSSD are advised to review their configurations.
● If local user caching is still required, administrators should explicitly configure SSSD to enable this functionality.
Note: This deprecation is a proactive step toward optimizing default settings in SSSD. While the default behavior has changed, users have the flexibility to tailor SSSD configurations based on their specific requirements. The deprecation aims to align with evolving best practices and ensure a more streamlined experience for administrators managing user caching in SSSD.
	The SSSD implicit files provider domain is disabled by default
	The SSSD implicit `files` provider domain, which retrieves user information from local files such as `/etc/shadow` and group information from `/etc/groups`, is now disabled by default.
	To retrieve user and group information from local files with SSSD:
	Configure SSSD. Choose one of the following options:
	1. Explicitly configure a local domain with the id_provider=files option in the sssd.conf configuration file.
	`[domain/local]`
	`id_provider=files`
	`...`
	2. Enable the files provider by setting enable_files_domain=true in the sssd.conf configuration file.
	`[sssd]`
	`enable_files_domain = true`
	Configure the name services switch.
	`# authselect enable-feature with-files-provider`

Which text is written by a human TW?

A

Simplified testing with infinispan-dev-mode.xml file

Infinispan Server distribution now includes `infinispan-dev-mode.xml` file that provides you with necessary configuration to simplify development experience with Infinispan and Quarkus projects. The `infinispan-dev-mode.xml` configuration file enables `BASIC` client authentication for the Hot Rod client and cross-site configuration.

To start the Infinispan Server with this configuration, complete the following steps:

1. Copy the server/ directory and rename the directory to server2/.
2. Start the first server using the infinispan-dev-mode.xml file.

```
./bin/server.sh -c infinispan-dev-mode.xml
```

3. In another terminal, start the second server using the same configuration.

```
./bin/server.sh -c infinispan-dev-mode.xml -s server2 -Dinfinispan.site.name=site2 -o 100 -Djgroups.mcast_port=46656
```

B

Infinispan Development Mode Configuration File (infinispan-dev-mode.xml)

This document provides a brief overview of the `infinispan-dev-mode.xml` file and how to work with it.

What is it?

The `infinispan-dev-mode.xml` file is an XML configuration file used with Infinispan, an open-source data grid platform. This specific file is designed for development environments and configures various aspects of Infinispan, including:

- **Networking:** How Infinispan nodes communicate with each other.
- **Caching:** Defines cache configuration and behavior.
- **Security:** Sets up authentication and authorization for accessing the cache.
- **Server:** Configures server interfaces, ports, and endpoints.

How to Use It:

1. **Location:** This file is typically located in the Infinispan configuration directory.
2. **Editing:** You can edit this file using a text editor to modify various settings for your development environment.
3. **Comments:** Comments within the file (denoted by ``) explain specific configurations and provide instructions for running the configuration in a multi-server setup.

Important Note:

This is a development configuration file and is not intended for use in production environments. For production environments, a different configuration with stricter security measures should be used.

EXERCISE: Now you try it!

Cause: Too restricted checks in openssl

Consequence: random tls connection failures

Workaround (if any): just reconnect

Result: handshake succeeds

gemini.google.com

bing.com/chat

chat.openai.com

How did you do compared to basic LLMs?

Naive Gemini	GPT 3.5 with a custom “style guide”	Naive ChatGPT
OpenSSL Strictness Causing TLS Connection Failures (Resolved) **Issue:** Recent updates to OpenSSL have introduced s lead to **random TLS connection failures** in s **Impact:** Users might experience unexpected connectio establishing secure connections (TLS). **Resolution:** This issue has been addressed in the latest Op Upgrading to the latest version is recommende secure connections. **Workaround (Temporary):** If updating OpenSSL is not immediately possible, a temporary workaround is available. Simply **reconnecting** after encountering a failed connection attempt may allow the handshake to succeed. **Note:** This workaround is not a long-term solution and should only be used until the system can be updated with the latest OpenSSL version. **Recommendation:** We strongly recommend upgrading to the latest OpenSSL version to benefit from the improved security measures and avoid potential connection issues.	**Release Note: Improved OpenSSL Connection Checks** **Issue:** If you encounter a connection failure, please try reconnecting. **Result:** With the updated connection checks, TLS handshakes will now succeed reliably. Please ensure that you download and install this release to benefit from the improved OpenSSL connection checks.	**Release Note: OpenSSL Update (Version X.X.X)** **Issue Addressed:** SL implemented TLS connection process hese connection attempting to X.X.X, OpenSSL moother handshake process and eliminating the occurrence of random TLS connection failures. Now, connections will consistently succeed without requiring users to reconnect. This update enhances the stability and reliability of TLS connections, providing a seamless experience for users.

.OpenSSL sometimes fails to connect with TLS when FFDHE is used

TLS connections that use finite-field Diffie-Hellman key exchange (FFDHE) sometimes fail. This is due to overly restrictive checks in OpenSSL. To work around this problem, reconnect and the handshake succeeds.

More RN prompts!

```
Cause: policy does not contain rules
to allow confined users run sudo
crontab

Consequence: confined user in the
staff_r role cannot use sudo crontab
to edit other users crontabs

Fix: a rule was added to the policy
Result: confined user in the staff_r
role can use sudo crontab to edit
other users crontabs
```

```
Cause: There was an error on login
into the system when the sshd
configuration file is larger than 256K

Consequence: system is remote
unreachable

Fix: The limitation was lifted

Result: User can login to the system
when the sshd configuration file is
larger than 256K
```

More RN prompts!

Solutions

```
.SELinux policy allows `staff_r` confined
users to run `sudo crontab`

Previously, the SELinux policy did not
contain rules to allow confined users to
run the `sudo crontab` command. As a
consequence, confined users in the
`staff_r` role could not use `sudo
crontab` to edit other users’ `crontab`
schedules. This update adds a rule to the
policy, and as a result, `staff_r` users
can use `sudo crontab` to edit other
users’ `crontab` schedules.
```

```
.Large SSHD configuration files no
longer prevent login

Previously, when the SSHD
configuration file was larger than 256
KB, an error occurred when logging
into the system. As a consequence,
remote systems were unreachable. This
update removes the file size
limitation, and therefore users can
log in to the system when the SSHD
configuration file is larger than 256
KB.
```

Even more RN prompts!

Feature – Ports used by keylime are now labeled as "keylime_port_t" in keylime SELinux policy. For ports with this label is allowed tcp connection.

Reason – Keylime SELinux policy before allowed to connect to all undefined ports and also most of the ports used by keylime were in group of undefined.

Result – It increase granularity of keylime SELinux policy, and port security can be targeted more strict.

Cause: ssh and sshd added the RSAMinSize parameter, which the ssh and sshd roles were not expecting.

Consequence: The user is not able to set this parameter. Automated tests for parameter checking were failing.

Fix: Add support for the RSAMinSize parameter to the ssh and sshd roles.

Result: User is able to set the RSAMinSize parameter.

Even more RN prompts!

Solutions

```
.Ports for Keylime have stricter rules in the
SELinux policy

Ports used by Keylime are now labeled as
`keylime_port_t` in the Keylime SELinux
policy. The policy now allows TCP connections
for ports with this label. This is because the
previous Keylime SELinux policy allowed
connecting to all undefined ports and also
most of the ports used by Keylime were in the
undefined group. As a result, this update
increases the granularity of the Keylime
SELinux policy, and port security can be more
strict and better targeted.
```

```
.Minimal RSA key bit length option in
the `ssh` and `sshd` RHEL System Roles

Accidentally using short RSA keys might
make the system more vulnerable to
attacks.
With this update, you can set RSA key
minimal bit lengths for OpenSSH clients
and servers by using the `RSAMinSize`
option in the `ssh` and `sshd` RHEL
System Roles.
```

In-class exercise: Generating technical documentation with ChatGPT 4

Scenario:

"Document how to manage and configure passkey authentication devices in the Identity Management (IdM) environment."

1. Go to ChatGPT 4 https://chat.openai.com/

2. Copy and paste the scenario above.

2. Save the first response generated by the AI in a gdoc.

3. Experiment with prompting to enhance the AI-generated documentation.

4. Save the best response in the same gdoc.

5. Prepare for discussion to share the reasoning behind selecting the best response and also the strategies you used to generate the documentation.

Exercise: Tips for prompting

"Document how to manage and configure passkey authentication devices in the Identity Management (IdM) environment."

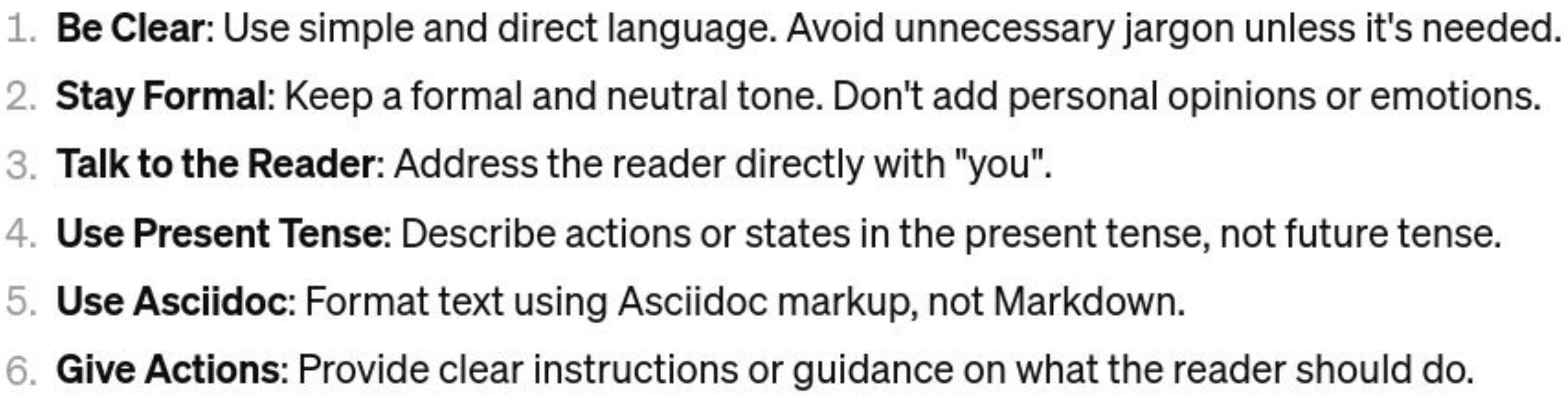
- Provide contextual details.
- Be clear and specific in your prompts.
- Focus on one aspect at the time (content format, structure, language).
- Work one section at a time (heading, abstract, procedural steps, verification…).
- Provide examples of the style or tone you're aiming for.
- Refine the prompts using different wording to get better results.
- Feed ChatGPT with content or reference material.

Exercise: Tips for prompting

Bonus: Custom instructions (system prompt)

Go to **Your profile** > **Customize ChatGPT** to customize the responses.

For example, create your own style guide:



1. **Be Clear:** Use simple and direct language. Avoid unnecessary jargon unless it's needed.
2. **Stay Formal:** Keep a formal and neutral tone. Don't add personal opinions or emotions.
3. **Talk to the Reader:** Address the reader directly with "you".
4. **Use Present Tense:** Describe actions or states in the present tense, not future tense.
5. **Use Asciidoc:** Format text using Asciidoc markup, not Markdown.
6. **Give Actions:** Provide clear instructions or guidance on what the reader should do.

You need TW skills even for AI tools

- User focus
- Minimalism
- Stylistics, applying a style guide

Source: unsplash.com/@rechaoktaviani

Technical writing is not just writing

- Research of new topics
- Planning, content strategy, user/content journey
- Verification and testing
- User experience

Source: unsplash.com/@rechaoktaviani

Thank you