The Gopher Times

Opus 6 - Gopher news and more - Oct. 2022

Sentient Regex

tgtimes

Can there be a sed one-liner that implements Artificial Intelligence? Depending on how you define Artificial Intelligence, it may!

How does it work for you? How more accurate than this is machine learning going to become to answer our existential questions?

```
sed -r 's/Is ([^y]*)?/Absolutely, (1)./
s/Is (.*y.*)?/I do not think that (1)./'
```

fold, fmt, par: get your text in order

katolaz

If you happen to read plain text files (e.g., phlog posts), you have probably noticed that, especially on gopher, the lines of a text file tend to be wrapped all to a similar length. Some authors are very strict on the matter, and like all the lines to be "justified" (i.e., all adjusted to have exactly the same length, by inserting a few spaces to get the count right). Some other authors (including myself) just do not allow any line to be longer than a certain amount of characters (in this case, as you might have noticed, the magic number is 72). But how to they manage to do that?

Most common editors have a command to format a paragraph ('M-q' in Emacs, 'gwip' or '{gq}' in vim normal mode, etc.). But obviously, there are several Unix tools that can help you getting the right formatting for your files. We are talking of fold(1), fmt(1), and par(1), so keep reading if you want to know more.

The oldest one is probably fold(1) (and it is also the only one to be defined in the POSIX standard...). It will just break each line to make it fit a given length in characters (by default, 72, which is indeed a magic number). Let's see how to wrap the lines of this post at 54 characters:

```
$ fold -w 54 20190213_fold.txt | head -10
    fold, fmt, par: get your text in order
======================================
If you happen to read plain text files (e.g., phlog po
sts), you have
probably noticed that, especially on gopher, the lines
 of a text file
tend to be wrapped all to a similar length. Some autho
rs are very strict
on the matter, and like all the lines to be "justified
$
```

Notice that fold(1) did not really think twice before breaking "posts" or "authors" across two lines. This is pretty inconvenient, to say the least.

You can actually force fold(1) to break stuff at blank spaces, using the '-s' option:

```
$ fold -w 54 -s  20190213_fold.txt |head -10
   fold, fmt, par: get your text in order
======================================

If you happen to read plain text files (e.g., phlog
posts), you have
probably noticed that, especially on gopher, the
lines of a text file
tend to be wrapped all to a similar length. Some
authors are very strict
on the matter, and like all the lines to be
$
```

Nevertheless, the output of fold(1) is still quite off: it breaks lines at spaces, but it does not "join" broken lines to have a more consistent formatting. This is where fmt(1) jumps in:

```
$ fmt -w 54  20190213_fold.txt |head -10
   fold, fmt, par: get your text in order
======================================

If you happen to read plain text files (e.g., phlog
posts), you have probably noticed that, especially on
gopher, the lines of a text file tend to be wrapped
all to a similar length. Some authors are very strict
on the matter, and like all the lines to be
"justified" (i.e., all adjusted to have exactly the
same length, by inserting a few spaces to get the
$
```

Now we are talking: fmt(1) seems to be able to to "the right thing" without much effort, and it has a few other interesting options as well. Just have a look at the manpage. Simple and clear.

Last but not least, par(1) can do whatever fmt(1) and fold(1) can do, plus much, much more. For instance:

```
$ par 54 < 20190213_fold.txt  | head -10
   fold, fmt, par: get your text in order
==========================================

If you happen to read plain text files (e.g., phlog
posts), you have probably noticed that, especially on
gopher, the lines of a text file tend to be wrapped
all to a similar length. Some authors are very
strict on the matter, and like all the lines to be
"justified" (i.e., all adjusted to have exactly the
same length, by inserting a few spaces to get the
$
```

will give more or less the same output as fmt(1). But:

```
$ par 54j < 20190213_fold.txt  | head -10
   fold,   fmt,   par:   get  your   text   in   order
==========================================

If you  happen to read  plain text files  (e.g., phlog
posts), you have probably  noticed that, especially on
gopher, the  lines of a  text file tend to  be wrapped
all  to  a  similar  length.  Some  authors  are  very
strict on  the matter,  and like all  the lines  to be
"justified" (i.e.,  all adjusted  to have  exactly the
same  length, by  inserting a  few spaces  to get  the
$
```

will additionally "justify" your lines to the prescribed width, while: something like:

```
$ head file.h
 *
 * include/linux/memory.h -  generic memory definition
 *
 * This is mainly for topological representation. We define the
 * basic "struct memory_block" here, which can be embedded in per-arch
 * definitions or NUMA information.
 *
 * Basic handling of the devices is done in drivers/base/memory.c
 * and system devices are handled in drivers/base/sys.c.
 *
$
```

can be easily transformed into:

```
$ par 40j < file.h
 *
 * include/linux/memory.h    -   generic
 *memory definition
 *
 * This   is   mainly  for   topological
 * representation.  We define  the basic
 * "struct memory_block" here, which can
 * be  embedded in  per-arch definitions
 * or NUMA information.
 *
```

```
 * Basic handling of the devices is
 * done in drivers/base/memory.c and
 * system devices are handled in
 * drivers/base/sys.c.
 *
 * Memory block are exported via
 * sysfs in the class/memory/devices/
 * directory.
 *
 *
$
```

Pretty neat, right?

To be honest, par is not the typical example of a unix tool that "does exactly one thing", but it certainly "does it very well" all the things it does. The author of par(1) felt the need to apologise in the manpage about the style of his code and documentation, but I still think par(1) is an awesome tool nevertheless.

fold(1) appeared in BSD1 (1978-1979)

fmt(1) appeared in BSD1 (1978-1979)

par(1) was developed by Adam Costello in 1993, as a replacement for fmt(1).

GNU tar(1) extraction is quadratic

tgtimes

When implementing something from the ground, it gets possible to build-up a simple home-baked file format or protocol looking perfect without any cruft and legacy. Easy to implement, fast to adopt, supporting everything you need from it, and not much more... Likely an alternative to a huge elephant in the room: the current standard in place used by everyone, huge, with many extensions with many use-cases...

Why bother, then, with implementing the huge and difficult file format or protocol? Maybe because it would be used by many software, and writing data in this slightly more bloated format would help making it compatible with all the software that already support it.

In this compromise, a limit can be drawn, across which the big and bloated format or protocol is dropped in favor of a simpler, more reasonable, less time-wasting alternative, eventually home-brewed.

The result is a new tar implementation written for the single special-case of a 1.1 TiB file! [1]

1 https://mort.coffee/home/tar/

BYTE Magazine Covers

tgtimes

The BYTE magazine lives among the legends of computer magazines.

Being a paper glossy magazine, it had fancy covers. Our usual data archivist heroes, Archive.org, have a large collections of covers for these things. [1]

On another level of effort, someone with passion and patience, actually went through recreatinhg the scene coming from these covers, that never really existed... Until they did! [2]

>> In the 1970s and 1980s, Byte magazine featured covers with beautiful, surreal paintings by Robert F. Tinney. What if the scenes that Mr. Tinney imagined actually existed in real life? And what if, as Mr. Tinney was painting them, there was a photographer standing next to him, capturing the scene on film?

>> That's the idea behind this site. I created and photographed real-world objects and composited the images together in order to show what Mr. Tinney's images might look like in real life.

1 https://archive.org/details/byte-magazine

2 https://bytecovers.com/

An experiment to test GitHub Copilot's legality

seirdy

>> This article was posted on 2022-07-01 by Rohan Kumar [1] and is now republished on this newspaper, with permission (CC-BY-SA 4.0).

Preface

I am not a lawyer. This post is satirical commentary on:

- The absurdity of Microsoft and OpenAI's legal justification for GitHub Copilot.
- The oversimplifications people use to argue against GitHub Copilot (I don't like it when people agree with me for the wrong reasons).
- The relationship between capital and legal outcomes.
- How civil cases seem like sporting events where people win or lose , rather than opportunities to improve our understanding of law.

In the process, I intentionally misrepresent how the judicial system works: I portray the system the way people like to imagine it works. Please don't make any important legal decisions based on anything I say.

The only section you should take seriously is Context: the relevant technologies .

Introduction

GitHub is enabling copyleft violation **at scale** with Copilot. GitHub Copilot encourages people to make derivative works of source code without complying with the original code's license. This facilitates the creation of permissively-licensed or proprietary derivatives of copyleft code.

Unfortunately, challenging Microsoft (GitHub's parent company) in court is a bad idea: their legal budget probably ensures their victory, and they likely already have a comprehensive defense planned. How can we determine Copilot's legality on a level playing field? We can create legal precedent that they haven't had a chance to study yet!

A chat with Matt Campbell about a speech synthesizer gave me a horrible idea. I think I know a way to find out if GitHub Copilot is legal: we could use its legal justification against another software project with a smaller legal budget. Specifically, against a speech synthesizer. The outcome of our actions could set a legal precedent to determine the legality of Copilot.

Context: the relevant technologies Let's cover the technologies and actors at play before I start my evil monologue.

Exhibit A: GitHub Copilot

GitHub Copilot is a predictive autocompletion service for writing software. It's powered by OpenAI Codex, [2] a language model based on GPT-3. [3] It was trained using the source code of public repositories hosted on GitHub, regardless of their licensing. In response to a Request for Comments from the US Patent and Trademark Office, OpenAI claimed that Artificial Intelligence Innovation , such as code written by GitHub Copilot, should be considered fair use . [4]

Many of the code snippets it suggests are exact copies of source code from various GitHub repositories. For an example, see this tweet: I don't want to say anything but that's not the right license Mr Copilot. [5] by Armin Ronacher [6] It contains a screen recording of Copilot suggesting this Quake code. [7] When prompted to do so, it obediently fills in a permissive license. That permissive license violates the Quake code's GPL-2.0 license. Copilot provides no indication that a license violation is taking place.

GitHub performed its own research into the matter. [8] You can read about it on their blog: GitHub Copilot research recitation, [9] by Albert Ziegler. [10] I'm not convinced that it accounts for the fact that suggested code might have mechanical alterations to match surrounding text, while still remaining close enough to trained data to be a license violation.

Exhibit B: The Eloquence speech synthesizer

I recently had a chat with Matt on IRC about screen readers and different types of speech synthesizers. I mentioned that while I do like some variety, I always find myself returning to the underrated robotic voice of eSpeak NG. [11] He shared some of my fondness, and also shared his preference for a similar speech syn-

thesizer called Eloquence.

Downloads of Eloquence are easy to find (it's even included with the JAWS screen reader), but I struggle to find any official pages about the original Eloquence. Nuance acquired Eloquent Technology, the developer of Eloquence. Microsoft later acquired Nuance.

Eloquence sample audio

Matt recorded this sample audio clip of Eloquence reading some text. [12] The text is from the introduction of Best practices for inclusive textual websites. [13]

>> My primary focus is inclusive design. Specifically, I focus on supporting underrepresented ways to read a page. Not all users load a page in a common web-browser and navigate effortlessly with their eyes and hands. Authors often neglect people who read through accessibility tools, tiny viewports, machine translators, reading mode implementations, the Tor network, printouts, hostile networks, and uncommon browsers, to name a few. I list more niches in the conclusion. Compatibility with so many niches sounds far more daunting than it really is: if you only selectively override browser defaults and use plain-old, semantic HTML (POSH), you've done half of the work already.

I like the Eloquence speech synthesizer. It sounds similar to the robotic yet predictable voice of my beloved eSpeak NG, but with improved overall quality. Unfortunately, Eloquence is proprietary.

Exhibit C: Deep learning speech synthesis

Deep learning speech synthesis [14] is a recent approach to speech synthesizer creation. It involves training a deep neural network on voice samples, and using the trained model to generate speech similar to a real human voice. One synthesizer using deep learning speech synthesis is Mozilla's TTS. [15]

Zero-shot approaches could allow a pre-trained model to generate multiple different voices. YourTTS [16] is one such example. This could allow us to synthetically re-create a person's voice more easily.

My horrible plan

My horrible plan revolves around going through two different lawsuits to set some judicial precedents; these precedents could improve the odds of succeeding in a lawsuit against Microsoft for Copilot's licensing violations.

If this succeeds, we have new legal justification that GitHub Copilot is illegal; if it fails, we have still gained a means to legally re-create proprietary software. It's a win-win situation.

Part One: set a precedent

1. Train a modern text-to-speech (TTS) engine using the voice a proprietary one made by a company with a small legal budget. Keep the model's internals hidden.
2. Then release the final TTS under a permissive license. Remember, we're still keeping the machine-learning model hidden!

3. Wait for that company to file suit. [17]

4. Win or lose the case.

Part Two: use that precedent against Microsoft's Nuance

Our goal here is to get the same legal outcome as the low-stakes trial run of Part One.

Microsoft owns Nuance. Nuance previously bought Eloquent Technology, the developers of the Eloquence speech synthesizer.

1. Repeat Part One against Nuance speech synthesizers, including Eloquence. Go to court.

2. Have the ruling from Part One cited as legal precedent.

3. Achieve the same outcome as Part One, demonstrating that we have indeed set precedent that works against Microsoft's legal department.

Implications of the outcomes

If we *win* both cases: Microsoft has the legal high ground. Making a derivative of a copyrighted work using a machine-learning algorithm allows us to bypass copyright licenses.

If we *lose* both cases: Microsoft does not have the legal high ground. We have good judicial precedent against Microsoft to use when filing suit for Copilot's behavior.

Either way, it's an absolute win for free software. Taking down Copilot protects copyleft from enabling proprietary derivatives (and by extension, protects software freedom). But if we accidentally win these two low-stakes test cases, we still gain something else: we can liberate huge swaths of proprietary software, starting with speech synthesizers.

Update: on satire

This post isn't satire through-and-through like something from The Onion. Rather, my intent was to make some clear points, but extrapolate them to absurdity to highlight other problems. I don't think I was clear enough when doing this. I'm sorry.

Copilot has been found to suggest significant amounts of code that is dangerously similar to existing works. It does this without disclosing obligations that come with those works' licenses. Training a model on copyrighted works may not be wrong in and of itself; however, using that model to generate new works that are not sufficiently distinct from original works is where things get problematic. Copilot's users could apply proprietary licenses to the generated works, defeating the point of copyleft.

When a tool almost exclusively encourages problematic behavior, the makers of that tool should have put thought into its implications. GitHub and OpenAI have not demonstrated a sufficiently careful approach.

I don't think that going after a smaller player just to manipulate our legal system is a good thing to do. The fact that this idea seems plausible to some of my readers shows how warped our perception of the judicial system is. Even if it's accurate (I doubt it's accurate, but I'm not cer-

tain), it's sad. Judicial systems incentivise too much predatory behavior.

Corrections It's come to my attention that Eloquence may or may not still belong to Nuance. Further research is needed. Eloquent Technology was acquired by SpeechWorks in 2000.

1 https://seirdy.one/posts/2022/07/01/experiment-copilot-legality/
gemini://seirdy.one/posts/2022/07/01/experiment-copilot-legality/index.gmi

2 https://openai.com/blog/openai-codex/

3 https://en.wikipedia.org/wiki/GPT-3

4 See Comment Regarding Request for Comments on Intellectual Property Protection for Artificial Intelligence Innovation submitted by OpenAI to the USPTO.
https://www.uspto.gov/sites/default/files/documents/OpenAI_RFC-84-FR-58141.pdf

5 https://nitter.net/mitsuhiko/status/1410886329924194309
https://twitter.com/mitsuhiko/status/1410886329924194309

6 https://lucumr.pocoo.org/about/

7 https://github.com/id-Software/Quake-III-Arena/blob/master/code/game/q_math.c
At line 552

8 I doubt anybody worth their salt would count on a company to hold itself accountable, but at least they tried.

9 https://github.blog/2021-06-30-github-copilot-research-recitation/

10 https://github.com/wunderalbert

11 https://github.com/espeak-ng/espeak-ng/

12 https://seirdy.one/a/eloquence.mp3

13 https://seirdy.one/posts/2020/11/23/website-best-practices/

14 https://en.wikipedia.org/wiki/Deep_learning_speech_synthesis

15 https://github.com/mozilla/TTS

16 https://doi.org/10.48550/arXiv.2112.02418

17 If the stars align, you could file an anticipatory suit against the company. It's common for declaratory judgement regarding intellectual property rights.
https://en.wikipedia.org/wiki/Declaratory_judgment

Glenda adventure

sirjofri

>> Glenda found herself in a dark forest.

Do operating systems dream of electric bunnies? Nothing is certain about that, but it does not prevent you to try to imagine.

Sir Jofri offers us a piece of fiction built out of the reality of the plan 9 operating system. [1]

Where should this go next?

A story first published on the 9front Mailing List.

1 http://sirjofri.de/oat/tmp/glenda_adventure.txt

Space Weather Woman

tgtimes

As she names herself, Tamitha Skov [1] is the Space Weather Woman. You read it right! She have been doing, since now close to ten years, forecasts about how is space weather is going.

Just a nerd fantasy? Only a sci-fi artist on a periodic one woman show? Not at all! Knowing what the sun is blasting toward Earth can reveal more useful than it looks. This includes:

- personnal safety for some plane flights at high lattitude.
- GPS communication, something happening in the pocket of many individuals, some of them even unaware of the involvement of satellites in the process.
- Long distance radio communication, which include Amateur Radio operators, but also emergency services and militaries.
- Something that Starlink did not invent [2] is satellite-relayed communication, including satellite internet and voice phone transmission. Actually a lot of wind turbines are being given satellite internet, and see how a little disruption [3] in satellite internet access can disrupt their operation.

And all of these fancy things are benefiting from Tamitha Skov's efforts as a researcher, but also by informing in layman's terms what is going on outter space.

>> Weather phenomena like coronal mass ejections, solar flares, and solar particle events. [4]

Science is elegant.

1 https://www.spaceweatherwoman.com/
https://yewtu.be/c/TamithaSkov

2 WildBlue, Viasat, NordNet...
First amateur stellite launched in 1961.

3 https://hackaday.com/2022/06/02/the-great-euro-sat-hack-should-be-a-warning-to-us-a

4 https://en.wikipedia.org/wiki/Tamitha_Skov

A C64 4chan Browser

tgtimes

The sewers of Internet in a C64? The link appeared on various IRC channels such as #electronics or #osdev, and not one more word. The investigation is open. [1]

1 <No_File> https://imgur.com/H36LTRV BACK 2 ROOTS!

I Hate Modern Technology

ig0r

>> The "advance of technology" is a source of excitement as well as frustration. ig0r gives us a crystallised view of human stupidity offered daily by technology.

Modern technology sucks. This might be me behaving like a pathetic little angsty hipster or trying to LARP thinking I'm somehow cool, but I think it's a genuine problem.

Planned Obsolesence

Technology is being designed to fail.

Apple purposefully makes batteries fail on their devices and solders them in such that replacing the battery on an older device makes no sense, forcing the customer to buy a new device.

Lenovo's quality has gone down the shitter. Thinkpads used to be thick, bulky, and rugged such that a caveman could use it in place of a club. New models bend and creak, the hinges breaking after several years of use while older models still run like new.

The reality is companies want people to consume technology, not use it. They care about making a profit rather than giving users a good experience, hence poor quality of manufacturing to speed up distribution, consumption, and the filling of landfills.

Modern Software

Modern software is just bad. Here's a few reasons why...

- It's idiot proof, in that I have little control over settings and configuration
- Software has become synonymous with adware (see Microsoft putting ads into explorer)
- I have to pay money for it (fuck you, if I could copy-paste a car I would)

Smartphones

Smartphones are the most annoying little shits, and for some reason they've become ubiquitous.

Restaurants are starting to ditch regular menus in favor of QR codes to be scanned with smartphones. Why? Paper is more reliable. This is a step backwards in my opinion. What if I don't have a data plan? What if I don't carry a smartphone?

Also why does everything have to be an app? Why does my passport have to be an app? I'm perfectly happy carrying around paper ID (paper ID doesn't spy on my).

People are idiots

Most companies justify making technology suck more by saying it's 'easier' and more 'convenient' for normal people.

Stop making easy and more convenient. Nobody asked for that. We were happy when technology was hard.

Better recording of the IRC Now events

ircnow

Here is a link with a better recording than the one in the previous tgtimes opus [1]

As a teaser, here are some random contents from it:

- Independence from Silicon Valley
- Self-Governance with Free Software and Right to Code
- Live demo of OpenBSD system administration from the ground up.

1 https://media.libreplanet.org/u/libreplanet/m/ircnow-of-the-users-by-the-users-for-

MNT Pocket Reform OS support

tgtimes

All these laptop and portable devices come with either Windows, Apple iOS or OSX, Android, sometimes Chrome OS, and even more rarely Ubuntu installed upon.

But the open hardware commnity is rising, and calls for a change. The MNT Pocket Reform lists more exotic operating systems as officially supported, [1] or at least acknoledged and listed in the front page:

- Debian GNU/Linux
- Support for other distributions: Arch, Ubuntu, Void
- Plan 9 (9front)
- Genode
- OpenBSD (in development)

Are we seeing a year of the open hardware laptop coming?

1 https://mntre.com/media/reform_md/2022-06-20-introducing-mnt-pocket-reform.html

Darknet Diaries

tgtimes

The mysterious Dark Net. While not an official institution, this hypotetical place built its very own identity through popular culture and medias. Famous and infamous, the depths of the limbos are explored in the Darknet Diaries podcast, covering and reporting the day-to-day events of that suspicious eden of shadow. [1]

1 https://darknetdiaries.com/
https://en.wikipedia.org/wiki/Darknet_Diaries

The Modern Mechanical Turk

In 1770, long before the exploitation of electricity, a machine was built in the pretention of being able to play Chess. This machine named Mechanical Turk was nothing more than a moving puppet actuated by a small human, such as a child. A child who is good at chess, that is!

Actuating levers, the operator would make the puppet move, fooling the audience that technical advances occasionally make use of black magic.

Amazon called a software platform Amazon Mechanical Turk. [1] It offers management for harvesting food for machine learning: human description of images, videos, products, and other kind of canned thoughts that machine learning can make use of to build models.

Uber for Cyber. Human translators shouting at machines the language they got whispered through their life.

Ghostworker. Noun. 1. Worker performing activity that will only be appreciated as data feeding an algorhithm. 2. Worker with no access to who it provide work to, both employer and client are invisible to him. [2]

given the very large scale at which these data-harvesting structures are deployed, it means that you, web user, have experienced the Google and Cloudflare "captcha" block window. That window preventing you to submit a form unless you click on all buses, tracktors, crosswalks, traffic lights... to verify that you are indeed a human and not a bot trying to access the website. Instead of prooving its belonging to the mankind, at the opposite, the user is explaining to machines what is a bus, a tracktor, a crosswalk, or a traffic light.

Here is your Great Technological Singularity for the greatest common entertainment: Nothing more than a moving puppet, actuated by humans, barely even paid for it, if paid at all... [3]

1 https://en.wikipedia.org/wiki/Amazon_Mechanical_Turk

2 https://www.ghostwork.org/

3 https://en.wikipedia.org/wiki/Mechanical_Turk

Publishing in The Gopher Times

you

Want your article published? Want to announce something to the Gopher world?

Directly related to Gopher or not, reach us on IRC with an article in any format, we will handle the rest.

```
ircs://irc.bitreich.org/#bitreich-en
gopher://bitreich.org/1/tgtimes/
git://bitreich.org/tgtimes/
```

Did you notice the new layout? We now can jump between single and double column as it is more fit: Some large code chunks will not fit in a two-column layout, but text is more pleasant to read on two columns.