[Transcript] I'd Rather Be Writing Podcast / Chatting about AI trends and tech comm with Fabrizio Ferri Benedetti

You're listening to another podcast on adratherbywriting.com.

Today I am talking with Fabrizio Ferri Benediti, who's based in Barcelona, Spain.

And we are chatting about just various news articles and particularly AI, sort of related news, more of an informal discussion and chat.

So Fabrizio, can you just introduce yourself a little bit, tell listeners kind of who you are, where you're based, what your interests are?

Right.

Well, first of all, good morning, Tom.

Thank you for inviting me to your podcast.

I work, well, I'm Fabrizio Ferri Benediti, that's the right pronunciation, by the way.

Don't worry about that.

Everybody gets it wrong.

And yeah, based in Barcelona, Spain, I work for Splunk as a principal technical writer or observability, rather technical stuff.

I do also open source documentation for open telemetry.

So that's my job currently, yeah.

Cool.

I'm excited that you've got experience in open source documentation.

I feel like that's an area a lot of people are curious about.

I mean, especially tech writers building portfolios or just people who want to get involved in that space.

So we'll jump into that a little bit later.

So I've got a list of articles that I think are going to be a good fodder for discussion.

And one of these near the top gets us right into this AI discussion.

There's a report from Vorrester that's talking about generative AI jobs, impact forecasts, and kind of like how things are going to play out.

And the gist of it is that any job that's subject to both influence and automation is kind of ripe for being eliminated.

A lot of jobs are subject to influence only by AI, meaning you could use the tools of AI to do parts of it, but not all of it.

And then other jobs where you can do all of it.

What are your thoughts on that for Vorrester?

I think this usually provokes a lot of strong reactions from tech writers, and most of us are very dismissive of it, but what's your general gist of that?

So as you might have gathered from my blog post on the matter, I'm an optimist.

Like I think that AI is not going to rob our jobs mostly, and that we can use it to do like AI-ided writing, technical writing.

I've seen that chart from Vorrester.

It puts like technical writing the dangerous part where high automation and we are just there right for being replaced by machines according to them, or at least according to the chart, depending on how you're ready.

But then I think computer science programming is like in the middle, but it might well be together with technical writing out at the top, same for creative writing.

What I think is that chart likes a third dimension, which is complexity or depth.
So depending on how you do technical writing, of course you can automate some bits, but if you do a certain type of technical writing, if you dig into some complex tutorials, if you've written a book, for example, on a topic, let's say, that's harder to automate.
And then I would also argue that the chart seems to focus a lot on the deliverables of technical writing.
Like someone thought, okay, so what do technical writers do?
Oh, documentation.
Yeah, of course, that can be automated.
Maybe they searched, they looked for some solutions, some prototype, and they thought, we can automate this.
But technical writing is not so much about the deliverable, I would say, as the process is more important, how you get to the documentation, the kind of information you collect, the processes
you set in motion.
There's lots of project management, there's lots of soft skills involved with that, and also with good programming.
So it's about focusing more about what makes our job, the human side of our job, what makes our job good to human beings rather than just the deliverable.
If we just focus on the deliverable, I think, yeah, I think our job is at risk.
Yeah, that's an interesting call out about the whole processes aspect of technical writing.
But you're certainly right.
I think that I have processes for release, of course, for sprint planning, processes around generating out new versions of docs, pushing it out.
There's a lot of complexity around that, I agree, that would be hard to be automated.
I mean, look, the reality is most people can't figure out how to even use these tools in their current jobs, let alone do it all with these tools.
So it seems very...
Well, for now, yeah.
It seems very future-facing.
I was also frustrated by this report because it doesn't really define what it means by technical writing.
It seems like the standard definition is that of a very junior writer where maybe an engineer has written something, hands it to a technical writer who tries to clean it up and make it readable and publish it.
And if that were the extent of my job, I would have left this career years and years ago because it would have been incredibly boring.
So yeah, definitely for the higher-level technical writer, where you live in complexity, in ambiguity, where you're really trying to get to the heart of what should even be written, that's a much more difficult task with AI.
But at the same time, I do want to use...
I do think, like you said, grouping technical writing with computer programming, you can use some of these tools to do parts of our job and tasks of our job.

I was listening to a podcast with Sam Altman, the head of open AI.
He was chatting with somebody.
Maybe it was Joe Rogan.
Okay, I don't usually listen to the Joe Rogan podcast, but that was one that I tuned in.
He was saying that AI is really good at tasks, not jobs, like not somebody's entire job,
but specific tasks, at least for now.
And I thought that was a good way of looking at it.
Yeah, which makes sense, you know, if you look at those as tools, I mean, we use tools
every day.
If we had a text editor, it would probably take way longer, like, you know, imagine our
life without, say, Visual Studio Code or an IDE, you know, just getting marked out right
or my table's right would be nightmare.
So it's okay to use tools.
Even if they're as complex as AI, you know, you're remarked there about how we use words
and how we define things, reminded me of one of the articles that you recently discussed
in your blog, which is the one about the word content, right?
I think it was Emma Thompson commanding about, and your times with it?
Yeah.
Commanding about the word content and how generic it feels and how dry it feels.
And again, that is, again, the problem of defining our job by the deliverable.
And in the most generic way, as content strategies, this is like a strategy of possibly everything
because its content can be everything.
So perhaps there has to be a shift there from the deliverable to the processes that we set
in motion, to the things we do in companies that are like, we are humanists in tech, I
like to say.
And so what it is that we do in tech, right?
We are not just, developers are not just churning out code, technical writers are not just writing
documentation.
They're doing something there that I think we have to define.
Let's move into that article called Emma Thompson is right.
The word content is, the word content is rude by Jason Bailey.
So yeah, this is some kind of an op-ed article in the New York Times commenting on Emma Thompson's
rejection of the word content.
And the context here is more of the writer's strike and how a lot of the creatives in Hollywood
just really feel like their content is being stuffed into the same bag, just by calling
it content, whether you write plays or you're writing TV shows, films, movies, other kind
of scripts, maybe even commercials.
And it's just like all content, which is, that really caught me by surprise because
I had never heard objections to the word content and we have that all over our profession,
content strategy, content management, content, this content, that is kind of like a substitute
for a substitute for documentation, I guess, but tries to be wider and include other sort
of deliverables outside of that.

Deliverables is another one of those generic words.
But so you're saying that this emphasis on the output is what is sort of the danger.
And you would rather see more emphasis on the process of a technical writer, that people just see us as producing documentation.
And if they see that result, it sort of leads to the conclusion that it's easy to have some machine generated or something else generated.
Yeah, yeah, I think so.
And it's now I was thinking now that, you know, like performative arts, you know, like doing a performance of something poetry, maybe, or there's another article you you brought up in your your digest about poets being hired by by EI company.
So, you know, those those performances are what makes our job not just unique and human those entertaining.
You can manage about, you know, how boring would it be to do the thing that machines do?
Right. So, yes, it's perhaps a little more difficult to to conceptualize technical writing as as poetry, we're probably not there.
But there's there's something there that that my mind is still circling around like I wrote those articles about humor in documentation, for example, like I'm trying to find the human bits.
But I'm still far from from having, you know, from from getting to anywhere.
I do think coming back to this whole content, you know, usage of the word content.
What we do isn't really all that creative and artistic, you know, I think that most creatives would probably put documentation all in the into the same bag, you know, it's not really, you know, it doesn't like release notes versus tutorials, sure, there's a mountain of difference between good ones and bad ones. But what what I think the creatives are really rebelling against is this rise of AI written content that is just kind of mediocre sludge, mediocre content. If you I swear, every time I'm on, I'm reading an article on medium, about half the time it looks AI generated, I'm getting more of a sense of it, because I use these tools a lot. And I can recognize when something just looks looks very explanatory, a little bit wordy, distanced, some diction seems off. And I think people want to separate themselves from, from that kind of content. We're going to be drowning in so much content that people can push out entire books. Like, I think Amazon created some kind of policy saying you can't publish more than three books a week or something or a day, something insane, right? It's like, but, but let's say let's play the future card here a little bit. Let's say that in two years, there's a lot of writers who are pushing out a lot of content, they're pushing out entire user guides in a day. And it's kind of lower quality than something that's more handcrafted. Maybe there'll be a desire to separate documentation based on auto generated docs versus, you know, written docs, I don't know. Perhaps, I think it's, you know, this, this connects a bit with several of the topics we mentioned and, and like things like brand recognition, for example, or, I mean, now, right now is EI's Lodge, you know, the dangers of having lots of EI generated common in the web, low quality, but we already had a crisis, like, you know, like, you know, lots of articles with keyword stuffing and we always had like a crisis like that.
Right now, I would say, yeah, so a tutorial, several tutorials per day, lots of documentation getting out. I would actually welcome that if they're useful, you know, if, if, if they're user-generated, you know, you know, you know, you know,

you know, you can, you know, you can, you know, you can, you know, you can, you know,
I would actually welcome that if they're useful, you know, if, if, if they're user-tested and,
and you see that the user accomplished something through those tutorials, that's okay.
But on the other hand, I think part of the brand differentiation, part of what make,
will make a tech brand successful, in my opinion, will be seeing the humans behind the process
and seeing the human touch. So part of that is already there in the form of DevRel,
you know, I think you have developer relationships going on, you have people
speaking at conferences, writing about their personal experience with a product,
and that complements tutorials and docs very well. Perhaps something, somewhere in between,
I think, is where the technical writers would have to find a connection.
Yeah. Hey, I want to return to this, the article on the poets that you, you brought up a little
earlier. Yeah. Because this is sort of an emerging new role in the writer-related world.
But this other article, let's see what it's titled, this is somewhere here in my list of
articles to chat about, there's one on, maybe that was a different one, but
the title was something like poets are training LLMs on, on how to write Hindi and Japanese
poetry, because these are gaps where the LLMs really fail. And for some markets, it's a super
high demand to be able to write original poetry in these languages. But in general, one job that's
emerging is kind of this content designer with LLMs, where your job is to help shape
the LLMs writing in some way, maybe not poetry, but maybe it's like, hey, are these responses
aligned with what one would expect or hope is the tone right? Or, you know, how does it deal with
certain situations of hyperbole? There's another case where a prompt engineer was doing what's
called red teaming, which is apparently working with LLMs around its responses to certain
problematic situations, like somebody wants to make dangerously spicy chili, you know,
that they're using dangerously as a hyperbole, but the LLM thinks that somebody's trying to
commit harm. And so the LLM is confused and thinks, I can't help people, you know, do that.
So this content designer role where your output is now just like working with the output of
the language model is a new one that's emerging. Is that something that appeals to you?
Or do you see that becoming more of a common role content designer with LLM?
Well, I think we're just at the initial stage. And by the looks of it,
that job doesn't look that different from content moderation, you know,
which brings lots of negative thought. We already read the articles about content moderation going
on in major social media networks, et cetera. So you have probably have to write lots of nasty
prompts, for example, for red teaming or, you know, and, you know, feeding poetry, even if it's
well paid, it kind of feels dehumanizing as you comment it. So, but I do think that the future
will look more like being like AI herders, more or less, you know, like shepherds of
artificial intelligence or LLMs. And we are entering the realm of science fiction now,
but I think it's entirely possible that we will have people specializing in communication
with AI. But LLMs is the most correct term so far now. But, you know, we just need to see these
evolving. But I do definitely see a future in this. Yeah.
I have a colleague who likes the term information reliability engineering,
where your job is kind of to make sure the information coming out of LLMs is reliable.
I don't know. But yeah, definitely, definitely, it'll be interesting to see how this plays out.
I said, I said in my comments on that article that it must be demoralizing as a poet, because like,

if you successfully train the LLM to write poetry in Japanese, for example, then your job is done. And so, and your skills are now kind of done as well. So, anytime that's the long term horizon of your job path, it seems pretty, pretty poor. Like, even outside of poetry, just getting it to write tech docs, let's say your job was to make sure that the tutorials have the right shape and the right approach and the right tone, you know, you get that working really well. And suddenly, you know, like, yeah.

Yeah. But, you know, there's a couple of closely related job profiles that can see a resurgent with this. And one is the teacher, and the other is the editor. And, you know, if you think of the typical lone writer at the startup, you know, not having enough resources to write 100 tutorials in a year. And, you know, if they get to be like an editor of an army of LLM generated, you know, content and points or whatever. That is actually a nice job to have. You know, it's nice that we record the, you know, this job description of the editor, the teacher even. And, you know, to those companies who are now trying to test LLMs and keep them poetry and keep them poetry, I would like to remind them that those jobs are existed for centuries. You know, we have teachers, we have engineers. And I think those, those roles can come back thanks to UI. Yeah. Yeah. It, we really are like entering a, a surgence of a lot of new possibilities. You mentioned having 100 tutorials or you have like hundreds of tutorials that you might kind of look over and so on. Just the ability for these machines to cover so many gaps of documentation is pretty amazing. Just yesterday, I was trying to fix like a podcast feed. And I was completely using the tutorials to understand both what was wrong with my iTunes feed and how to fix it. And this sort of like era of hyper personalized instruction, where you, you want to have, in that case, I wanted to have like a tutorial on the right tags for an iTunes feed, as well as like how to fix the ones, the problems that were, that existed with my current one. There's a southern article on this list that we've got about hyper personalized content. And I really think these tools kind of usher in this era, not just like, hey, give me a tutorial on how to implement something, but like some, a tutorial tailored to your own content, to your very specific situation with the bugs that you're seeing.

What are your thoughts on hyper personalization of content? You think this is a huge win in the help world? Well, I'm a bit torn there. I think it's, it sounds like a great idea. I just feel quite skeptical about automated personalization in general, even if it's through EI. You know, I think, first of all, I think the user might feel a bit weird about it. But like, and it is, I'm not just talking about providing sensitive data, which might happen. I mean, if you contact support from any, you know, for any company, that conversation is protected. I'm not so sure about, you know, speaking with an EI and a chat mode, or, you know, maybe like counting up this session will be wiped out once you close the window, something like that. I mean, that might work. But there's, there's that problem. And then I think there's, there's a problem of being overly specific. And, you know, once you unleash an EI, helping a user, and you cannot supervise it, there might be some risk there.

So I'm not entirely sure that that would work. I do think that hyper personalization on the other hand, to me, it means more like, you know, being, you know, using that content, and add yourself to that content as a human. And, you know, you're merging the best of two words there. But it's a completely different meaning for me. Yeah, I'm curious about more of the risks that you brought up that I'm trying to walk, trying to see like what potential risks there are, for

sure, like, coming back to my podcast feed scenario, one risk is that my learning becomes very narrow. I learn only what I need to to fix my problem. And I don't expand my knowledge in any other way. And I don't become aware of things I don't even know I don't know about. There's no sense of deeper learning. It's just like, Hey, what was the problem fixed? Bam, move on. And yeah, that probably is short citing my skills in the long run. Yeah.

Why don't we shift into a different topic? Because you mentioned that you are you work in open source. And there's an article that Daniel Beck wrote, I didn't really have a date on it. I thought it was recent, but it says the open source docs portfolio myth. Because we're talking about like learning and so on. And so when you have new technical writers who want to build their portfolios and show that they they have learned technical writing or that they're they've got some skills, they usually that the typical advice is, Hey, go find an open source project that needs documentation, go volunteer your skills to write the documentation. And, you know, a lot of these projects are super hungry for doc writers or for documentation, they, you know, like, something like 90% of people say that the docs are poor, that they need help. But Daniel says, you know what, this is kind of a myth, people, people try to volunteer their skills. And they're met with resistance, silence, it's, they don't make progress. It's, it's an experience of frustration. And he really recommends like, changing your expectations instead of thinking, Oh, I'm going to build my portfolio portfolio, think, Oh, I'm going to learn how to work with like GitHub, I'm going to learn the mechanics of, of how communication is done on an open source project. What are your thoughts on Bex article? Is he on target? Or do you think there's still value in new writers trying to build portfolios this way?

So I think, I think this article was very necessary. In that, you know, it set the expectation straight. What happens with open source is that we often forget that there's always a community aspect to it. And community means there's a social code. There's, there are people there who know each other, or have been working together for a while. So it might seem like a collective building where, you know, someone passed by and dropped some code or some docs, but it's not that casual, usually. What happens is that you have a small group of people and you have to play the anthropologist a bit, especially if you are a writer, right? So you cannot just open a huge pull request of the blue. You have to approach the project, maybe have a look at the issues. It takes time. So whenever I see, especially with things like the, what is it, the the week of docs or something like that, don't remember the name right now. I think it's a Google initiative. Yeah, yeah, I think, I think that's right. The week. Yeah, so you can see the hoarders of people going to repose and try to contribute a pull request. And I feel bad for them because they, they want to do good, right? But the communities don't seem so welcoming at the beginning. And it's true. It's just because they are, they're a social group, like any other, and you have to think in social terms, right? And I think this post has a good job at describing that situation. And I was, I was remembering a very, very interesting book on the topic. Let me just check by Alexander Kitzali, which is Stocks as Ecosystem, published by A-Press. And it really delves into these aspects of, you know, collaborating and entering a community and what that means. So that's a significant shift from, you know, just thinking about, well, this is my code contribution. It compiles. Pipelines are green. Why doesn't it get through? Well, you have to think also about the context.

Interesting. Interesting. Yes. I saw that book Surface in the Feeds and it, is it called Doxus Community? And you said the author is Alexandra Kitzali. Doxus Ecosystem by Alexander Kitzali. Yeah. Yeah. And, and I think the book was sort of rejecting this Doxus code emphasis, where really the, the whole gist of things is like learning how to, learning the software programming workflows that, that kind of make the content move along and get published. I hadn't connected it with this. This is a, it's a great connection. So by the way, is that a book you've read entirely or just part, partway? Partway, yeah. It has a huge pile of books, but it was completely spot on in regards to this blog post by David. Yeah. So seeing it as Doxus Community and joining a community might be the way to really click with that. That's really cool.
How did you get involved in open source, Fabrizio? Like you mentioned that this is what you, you currently kind of do and emphasize. I'm just curious, like, did you choose that or was that just sort of where you landed by other circumstances? So at the very, at the very beginning, it was casual. My first contributions were to, to a Hugo theme Doxy, which by the way is maintained based on Google's. And that was before I was at Splunk. Then I joined Splunk and a sizable amount of, of the products that we develop around reliability are open source. And Splunk is a big contributor to open telemetry, which is an open source project under the Linux foundation. And so, you know, I found this, these tasks to, to be super interesting upstream, as we call it, like going to, to the open source that we then distribute. And there's so many interesting interactions there because we do distribute the open source code as Splunk, but at the same time, we contribute to it. And, and it's not just code contribution, but it's also documentation. And I'm, I'm trying to improve that. And, you know, with that also came the community aspect, you know, like watching activity at the beginning, then starting to contribute, attending some, some meetings, some special interest group meetings, for example. So you have to embed yourself gently. Otherwise, even if you're part of the main contributors companies, that can be like a pushback. Yeah, you know, the, trying to connect this open source theme to LLMs, I think there is a, an interesting connection in that with so much of the content on GitHub being open, it's really ripe training grounds for a lot of LLMs, as I understand it. I mean, got open access to all, all of this code. And from that, I think a lot of these tools can sort of extrapolate the logic and provide documentation about so many of those, those details. Do you think that LLMs might kind of bridge this gap between poor documentation with open source projects and fill this need? Like you go to a project, you're trying to get it to work. It requires a bunch of steps with an unfamiliar technology to you. So, you know, you rely on other tools to kind of fill in the gaps of missing docs.
Well, especially, yes, that's right, yes, especially in those situations where you have, which happens a lot with open source, you have dependencies, which might be very fragile and, and under documented. And in that situation, I think LLMs can really help bridge the gap and, okay, I want to, you know, I have X, I want it to work. But it relies on Y and Z and, and those are not documented or very poorly documented. But there might be code samples out there, you know, and the process of doing that manually can take many hours or days. And with LLMs, at least you have a hint of how you could piece the things together. So LLMs, in that sense, are a very good tool for exploration in, in the open source wilderness.
Yeah, I was trying to use a tool. I can't remember this. Oh, no, this was, this was mark prompt. I was trying to get it to work. And a certain implementation path required a lot of

information about node. And like, you know, I know node is a huge thing, I should probably be really familiar with it and all the different tools and setup, but I wasn't. And their documentation just assumed that somebody would have that sort of node background. And it did. I used chat GPT to really fill in a lot of the blanks there. Yeah. So I think there was another article that I was recently talking about MDN's AI explain feature, where part of the challenge that we're trying to do is provide instructions for features that were split across different pages, or code samples for features that were explained on different pages. And it's the same concept here, where you've got lots of different pages. And maybe the code you need requires you to be familiar with like 10 different pages. And it's just too much to ask for that documentation to somehow explain all those other areas. Hey, I'm, I'm kind of curious, what, what sort of AI tools do you experiment with? Like, do you, do you, if you need to use an AI tool, what's your, what's your go to tool? Is it chat GPT? Is it Claude, Bard, Poe, Perplexity, even you.com? Yeah, that's true. For now, I use co-pilot at home, and, and chat GPT as well. You know, just vanilla chat GPT. I tried, I tried being searched, but I didn't find it so, so good. And mostly I use it for technical stuff, you know, from time to time, I might entertain myself with some questions about philosophy or things like that, but it's mostly technical stuff. Then I also tried out mark prompt. I think that was the very cool idea. But I think it's, it still needs lots of development. You know, there are companies out there trying it out, trying similar solutions. And you always get to the point where it's not really time to hello world, I would say is time to I don't know. And the time to I don't know is the time till you get the AI to admit that they don't know how to do something. And that time is still very short with lots of tools. Like, you know, you start asking something complex and, and all of a sudden, either the training set is not sufficient or, I don't know, maybe the settings are too conservative, but you get to the I don't know point, which is like the 404 of LLMs. I hadn't heard that, that phrase before time to I don't know, I kind of like that. So I might have been benefit. I don't know.

This brings up a question I haven't really fully resolved in my mind. The difference and the trade offs between a specialized LLM that's trained on a very specific body of docs, sort of like a micro LLM versus more of a macro LLM that has one stop shopping for almost any type of question you have. And the latter, the, the one stop shop experience is subject to a lot more hallucination than the micro LLM. What's your preference? Would you rather use a specialized LLM that's going to have very quick boundaries on what it knows? Maybe it's not trained on node, for example, and, and so it can't help you with that. It can only help you with maybe the product that you're, that you're working with. Well, ideal scenario that would be like a switch, you know, like, okay, so be the generalist, be the specialist, you know, and then you can compare. If I were a doctor, of course, yeah, I would, I would rather use the specialized one. Like, you know, especially if it's about the very sensitive stuff like surgery or things like that. I don't know if there's been any application by the way in medicine, but my, I think it's, it's okay not to drop entirely even if it hallucinates. There's a richness in the general models, you know, in the know it all. And perhaps I feel sympathy for those approaches, because I think technical writers are also generalists, and we bring many diverse skill sets and knowledge to our field, which might, you know, add something different. So I would keep the field open for the general models.

Yeah, that's a, that's a good connection about how tech writers are generalists. And so that might appeal more to that mindset, because certainly, like when I'm using technology, I'm trying to learn about something, I only need a very basic understanding of it. And especially if the information is online somewhere, then usually that, that is enough for me. But if I were a specialist engineer, working with something very kind of complex and deep, yeah, I would probably prefer the specialists LLM and be very, not have a lot of patience for error and so on. So it's interesting. Now you mentioned that you don't, you don't really use it for stuff outside of tech. What about your blog Paso Uno? And by the way, I mentioned your blog and your URL in this response, but tell me, do you not find it useful to use an AI at all with any of your blog efforts from coding to idea generation to other tasks?

So I, I might have tried a couple of times, maybe a few months ago, but the, perhaps, perhaps child GPD doesn't know that much about technical writing, I don't know, but it didn't really come up with things that, that appeal to me, you know, like the kind of topics I had to write about, you know, they were very generic, but perhaps it was also because of the kind of prompts I was providing. I found more value in, in, in generating pictures using Dali, which I think you also do. And I did that for one of my last posts, the base theory thing, you know, I wanted to create this medieval content taxonomy, but the illustration part, well, that would have taken ages. So I just relied on, on AI for that, which is in itself an example of AI assisting a writer, right, with some content. But for writing itself, no, not really. Actually, now that you said it, I might try again, but perhaps there's, there's a psychological block there. I don't want to, like, I wouldn't feel honest, you know, in doing that. So I think people come to my blog because they, they feel like what I write is very personal, very opinionated, has personality, and using child GPD, I pretty sure they would notice that something isn't quite right.

Yeah. Well, this whole topic of honesty is an interesting one. But before I do want to comment on that briefly, but coming back to the images, for sure, you know, if you just have long walls of text in your blog, the images, adding images can really spice it up and make it interesting. Your, your post on the bestiary compendium, you know, it was very creative and it had cool images. I was most fascinated, honestly, by the fact that you were able to get images that all seemed like part of a collection. That's really hard to match the same style and approach from one image to the next. Maybe it's because of the style, maybe, maybe, I don't know, but I really struggle with that. It seems like every image looks different, artistic. Yeah. But you see, this, this, the, the value that, that the human writer brings is, for example, curation, which is something that I think will be out of reach of the eyes for a long time, if always, like, you know, because you have to take a step back from, from whatever problem space that the eyes are focusing on. And, and that content switching that zooming out is not really a thing within the reach of a general purpose of artificial intelligence. Or so I like to think, but for now, that's the situation, right? So whenever there's something like editing, curating, watching for voice and tone, that's where we can really bring value. Yeah. Yeah, for sure. If you just, you know, give the AI AI kind of open reign to, to add images, it would be a disaster, right? You have to very carefully steer that and provide input and direction and, and, and other judgment. Now let's look at this other aspect of your comment about the honesty element. You know, this is something that, that I, I, yeah, I've wrestled with this as well. For example, in a lot of my newsletter summary posts, newsy summary posts, where I've,

I haven't really been doing this that long, but I try to like find interesting related news to tech com, summarize it and comment on it. Well, for the summaries, definitely rely on some summary tools just to speed things up, whether it's cloud chat, GPT, Bart or some other tool. And my feeling right now anyway, is that if people can't tell if something is machine generated or not, it really doesn't matter. I used to feel a lot more like, Oh, I'm cheating. I'm plagiarizing. I'm being dishonest. And then I thought, you know what? What's the difference between somebody who

uses Grammarly to fix all the, the grammar and style or, or uses word tune or something else, or who even has like a human editor that he or she is passing many drafts through and maybe the editor is, you know, maxing out their, their red pen and all the changes and so on. At what point do you say they're just different gradations of assistance for things? And, and why should I feel conflicted? It's not as if I'm signing, well, I guess I was going to say signing my name or something. I guess I am, but it's not like an artist who's putting their, their, their autograph on a painting, for example. It's not really a work of art. It's much more functional. But yeah, but you know, that's what you mentioned artists, right? Like great guitar players have their signature guitar or great pianists have their favorite instruments, sing for violin, etc. Painters have their tool of choice. So what I would like to see in the future, and probably this is already, you know, cooking is, let's say that tomorrow I'm writing my blog and at the same time, I'm also presenting my own LLM trained by myself with its own name and identity. So I'm not using Bard. I'm not using chat, I'm using Fabio. Let's call it like that. My own LLM. And that's his also is my own signature, just in a different form, which is a system me, but it's not the generic one. So nobody else can have it. So, you know, there's, there's also that part there. It's like your own personal Star Wars droid, so to speak, you know, yeah, I agree. Like that's an extreme case where you've like, let's say you've trained the LLM to respond in a certain way, you know, that that's definitely a reflection of your expertise. Even without going that far, if you know how to like write a prompt in a way that gets the response that you want that's really on target, there should be some kind of credit for that, you know, like, you can't just say, oh, this post was written by chat GPT as if it just like did everything. I tried to write a post the other week,

where I would kind of describe what I wanted it to say at every, for every single paragraph, which seems incredibly tedious and didn't really work out that well. But it was my attempt to kind of use these tools in a more directed way. And at the end of that, would I say, oh, this is just written by by Claude AI, you know, like, well, I mean, there's a lot of input there in direction and steering and judgment is it wasn't just like, click a button, here it is. Anyway, but yeah, that the sole issue of ethics definitely makes a lot of us uneasy, and it's different mindset. And for sure, part of the writing process that's rewarding is thinking through things yourself and and making connections and realizations and discovery along the way. So I don't want to, like, remove that element of writing. But at the same time, I don't want to be, I don't want to be antiquated and not like able to leverage tools that are available that speed up, especially the mundane parts of writing. So yeah, I totally agree. And, you know, I think that getting back to the general theme of easy, I got to get our jobs or similar themes or topics. Well, I think that, you know, we are not really antiquated when we say that this is not a problem of, you know, EIs can be problematic only if you assign initiative to them that they don't have. Like, I think we are attributing initiative and will to things that

don't have it. And that's a very human thing to do. We humanize everything. And we, you know, we think of EIs as terminator or robots with their own goals and initiative. And that's not the case. LLMs are completely passive until you provide prompt. And the element of initiative, the element of self-starting something, that's still human. And that's still on us. We give direction.

And machines cannot have that. It's practically impossible that they can have initiative. It's something that, at least in my opinion, with the little psychologist out in college, I think that belongs only to the organic matter realm. So we, well, the best we can do is not to think about LLMs as independent agents. They're not, they're not. Yeah, that's a good way of, that's a good perspective to have. And, and definitely comes back to this comment I made earlier about using these as tools to perform tasks instead of entire jobs. Because as a job, you have much more of the agents in the individual direction that is needed.

All right, Fabri, so we've talked about a lot of articles. I think let's wrap this up. Can you tell people a little bit about, like Paso Uno, how people find you online? And let's say they want to just follow you or reach out to you. What should they do?

Sure. So if you look for me, when I guess it, you will see my name in the podcast. Once it's online, you can Google me. I'm on LinkedIn, mostly, also on Twitter, and MasterDone. But my blog is self-hosted. I strongly believe in self-hosting content in terms of retaining property of it.

See, we're back at the word common. And the name is Paso Uno, which is Paso is, Paso Uno is an Italian expression for stop motion. I explain a little bit more in the about of my blog, but why I called it like that. But yeah, it's Paso.uno. You cannot miss it.

Cool. And I'll add a link to that in the show notes. And yeah, I'll include your name. You know, I messed up your name in an earlier post. Instead of Fabrizio, I said Fabio. And now I see why you thought- Well, that happens all the time. I see why you thought it was so funny though, because like the Fabio guy has such long flowing hair. Right. But I feel a little bit old with that reference. Like I feel like the younger generation did not get it. You know. Yeah.

Yeah, for sure. For sure. Anyway, thanks again for coming on here and chatting with me about all these techcom topics. I really appreciate it. Thanks.