JOSSCast #1: Eva Maxfield Brown on Speakerbox – Open Source Speaker Identification for Political Science

Arfon M. Smith

Published January 25, 2024

Citation

Keywords
Josscast

Abstract
Subscribe Now: Apple, Spotify, YouTube, Stitcher In the first episode of Open Source for Researchers, hosts Arfon and Abby sit down with Eva Maxfield Brown to discuss Speakerbox, an open source speaker identification tool. Originally part of the Council Data Project, Speakerbox was used to train models to identify city council members speaking in transcripts, starting with cities like Seattle.

Copyright
Copyright © Arfon M. Smith 2024. Distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
Eva Maxfield Brown on Speakerbox - Open Source Speaker Identification for Political Science
In the first episode of Open Source for Researchers, hosts Arfon and Abby sit down with Eva Maxfield Brown to discuss Speakerbox, an open source speaker identification tool.

Originally part of the Council Data Project, Speakerbox was used to train models to identify city council members speaking in transcripts, starting with cities like Seattle. Speakerbox can run on your laptop, making this a cost-effective solution for many civic hackers, citizen scientists, and now .. podcasters!

From the advantages of fine-tuning pre-trained models for personalized speaker identification to the concept of few-shot learning, Eva walks us through her solution. Want to know how this open source project came to life? Tune in to hear about Eva’s journey with Speakerbox and publishing in JOSS!

Links
- Paper: Speakerbox: Few-Shot Learning for Speaker Identification with Transformers
Transcript

[00:00:00] Arfon: Welcome to Open Source for Researchers, a podcast showcasing open source software built by researchers for researchers.

My name is Arfon.

[00:00:11] Abby: And I’m Abby.

[00:00:12] Arfon: And we’re your hosts. Every week, we interview an author who’s published in the Journal of Open Source Software. This week, we’re going to be talking to Eva Maxfield Brown about their software, Speakerbox, few shot learning for speaker identification with transformers.

[00:00:27] Abby: Yeah. And I think this was a great interview to kick off with. Eva was really excited to talk about her work. And I thought it was very applicable for podcasting!

[00:00:35] Arfon: Absolutely. And, I don’t think I said it at the start, but this is our first ever episode, so we’re really excited to start with this really interesting piece of software.

[00:00:44] Abby: Awesome, I guess we can just dive right in.

[00:00:46] Arfon: Yeah, let’s do it.

Welcome to the podcast, Eva.

[00:00:49] Eva: Hi, how are you?

[00:00:50] Arfon: Great. Thanks for coming on.

[expand]

[00:00:52] Abby: Yeah, glad to have you in one of our early episodes.

But do you want to tell us a little bit about yourself just to kick us off?
Eva: Yeah, sure. My name is Eva Maxfield Brown. I’m a PhD student at the University of Washington Information School. My research primarily focuses on the science of scientific software more generally, but I also like to believe that I practice building scientific software. So I have experience building software for computational biology or microscopy and also political and information science.

The project that we’ll be talking about today falls into that political and information science capacity of building not just studying.

Arfon: Awesome.

Tell us about Speakerbox. Tell us about the project. This is a paper that was reviewed in JOSS about a year ago now, a little bit over, or initially submitted. I’m just curious, if you could tell us a little bit about the project, why you started it, and what kinds of problems you’re trying to solve with Speakerbox.

Eva: Yeah. I’ll have to back up for a little bit. Speakerbox is part of a larger ecosystem of tools that fall under this project that we were calling Council Data Project. This was a system that basically was like, Oh, what’s really hard for political scientists is studying local government and one of the reasons why it’s hard to study local government is most of the stuff you have to do is qualitative.

It’s very hard to get transcripts. It’s very hard to get audio. It’s very hard to get all these different things in a standardized format for all these different councils. So, Council Data Project started a long time ago and it tried to lay the groundwork of getting all those transcripts and systematizing it in a way.

But one of the longest requested features of that project was being able to say that each sentence in a transcript was said by who, like this sentence was said by Council Member A and this sentence was said by Council Member B and so on. And so there’s a problem here, right?

We could spend the time and resources to build and train individual speaker identification models for each city council. But that’s a lot of time and resources that we don’t really have access to both as researchers, but also just as people interested in contributing to the local government area.

And so Speakerbox really came out of that problem of how do you quickly annotate and train individual speaker identification models for application across a transcript, across a timestamped transcript.

Abby: Yeah, reading this paper I got really excited as a nascent podcaster. So I could see how this is immediately applicable because I have this audio files, and can we identify who’s speaking with what. Anyways, I could see how this could be useful.

Arfon: Yeah, maybe we could, maybe we should try it
[00:03:17] Abby: Yeah, we can. On this episode.

[00:03:19] Eva: There was, I think there was one of the early GitHub issues that we got after publishing the paper was from someone who was recording their research lab’s meetings and basically wanting to train a model just to tag their own lab members. And I was like, I guess that’s a use case. Sure. That makes sense.

[00:03:35] Abby: We can train it to tag ourselves as hosts identify a mysterious third person each episode.

[00:03:41] Arfon: Yeah, there you go

I was going to say, so, the Council Data Project, could you say a bit more about who the users are, who the target audience here is? Is that like studying government and how councils work? Who is your audience?

Who would use Speakerbox?

[00:03:55] Eva: Yeah. So that’s a really good question. So I would say the larger ecosystem of Council Data Project would fall into political science, especially the subdomain of urban or city level, municipal level political scholarship. Then there’s also information science questions.

And I think there’s a lot of really interesting information science questions, such as how do ideas and misinformation and disinformation enter city councils and so on. So I think those are two very core audiences of that research, but there’s also the public side.

So there’s just general users, right? Council Data Projects comes with a website and searchable index that you can search through all the meetings that we’ve transcribed and everything. So we do have general citizen users. We also have people at the cities that we support like city employees as users as well. And journalists.

So there’s a couple of different people that are interested in it from the Council Data Project angle. But specifically for Speakerbox, as we just talked about I think there’s a number of cases where this is just useful. So that was one of the reasons why we split it out as its own thing. Certain tools for Council Data Project will likely be hard coded towards our infrastructure or towards our technology stack. But, Speakerbox is just this general toolbox of quickly how to train a speaker identification model has a broader application set.

So things like podcasting, things still in the domain of journalism or in the domain of political science, sure, but also in music or in anything else that kind of deals with this very fast audio labeling type of dilemma.
Abby: Yeah, another thing that really struck me reading the paper and also just hearing you talk about the Council Data Project plus Speakerbox. There’s a lot of open source and open data tools and resources that you’re using and you’re drawing from.

Did that really affect why you’re doing this openly? Can you just tell me about the whole ecosystem?

Eva: Yeah. that’s a good question. I think from the start we positioned. Council Data Project in the genre or domain of the prior, past era of open data and civic collaboration type of, angle. We originally envisioned that a different group would be able to deploy all of our tools and technology for their city council, wherever they are.

That would just reduce the cost on us. And so that was partly the reason why moving to the open source model was really nice. All the documentation can be available if they need to make an edit for their own city, whatever it is, they can contribute back. There are also existing projects prior to us, in the history of this civic information landscape that did this pattern. But I think we’ve shifted it to our own needs a little bit. That said, the fast pace of machine learning and natural language processing tools is just so quick that you could expect a new version of some package came out and there’s going to be, a breaking change.

And to be honest, at this point, I work on a lot of different projects and it’s really nice to push it open source just because I know that there are other people, whether they’re related to Council Data Project or not, that are also going to use it and need it and can contribute to helping fix and maintain and sustain the project itself.

Arfon: So are there really obvious alternatives? I understand why you would want to publish this. There’s lots of opportunity for pushing things into the public domain but are there tools that other people were using previously that you’ve found efficient or are maybe using I don’t know, proprietary languages or something.

What were the motivations for publishing a new tool specifically?

Eva: I think there definitely are proprietary technologies that allow you to do this. I think maybe Google is probably the most famous. I think they have a whole suite of give us your data set and we will try to train a model for it.

Other companies as well. I think Microsoft probably has their own version of that. Amazon as well. So there’s definitely proprietary kind of versions of this. I think what I particularly wanted to do was: One, this is something that, again, focusing on specifically our core user base, which is that civic technology type of person, they don’t typically have access to a lot of money to pay Google or to pay Amazon or
whatever but they might have access to their laptop and that has a decent CPU, or they might have, they might even have a desktop that has a GPU or something.

And so I wanted to make it possible where it’s like we can, quickly annotate, like we can demonstrate exactly how to annotate this data and also demonstrate using consumer grade GPU. I think the GPU that I used in the example documentation for all of this is five, six years old or something at this point.

And show that it still works totally fine and it trains in under an hour. Everything is able to be done and let’s say a weekend of time. And I think that was really needed. I’ve already heard from other people, whether it’s, other research labs or other people trying to deploy all of our Council Data Project infrastructure that they are able to quickly iterate and train these models just on their own local machine.

[00:08:33] Arfon: Yeah, that’s really cool. You mentioned civic technology. I think I heard you say civic hacking space before. I think that the idea of publishing tools for communities that don’t have tons of resources and don’t necessarily, aren’t able to pay for services is really important.

And I was fortunate to be in Chicago when the civic, hacking scene was really starting up in the late

[00:08:54] Eva: Oh,

[00:08:55] Arfon: 2000, Yeah, about 2009 2010, and that was a really vibrant community, and I think still is, and there’s just lots

[00:09:01] Eva: Yeah. Chi Hack Night is

[00:09:03] Arfon: Chi Hack Night, there you go, yeah, lots of open, civic tech is, it’s a really cool, space, yeah, cool, okay.

Okay, so I can run this tool on, a relatively modest GPU, but what’s actually happening under the hood?

So there’s this transformer thing, there’s some learning that happens, what, does it look like to actually train Speakerbox for your project, for your dataset?

[00:09:27] Eva: Yeah. fortunately for us, Transformers is I think two things. So when people say the word Transformers, it’s, a couple of things all in one. There’s Transformers the architecture, which I think we can maybe shove aside and say it’s the kind of quote foundation of many modern contemporary machine learning and natural language processing models.

It’s very good at what it does and it works based off of trying to look for patterns across the entire sequence, right? So looking for whole sequences of data and saying where are these, things similar? So that’s great. But there’s also
Transformers, the larger Python package and ecosystem, which is run by a company called Hugging Face.

And that’s, I think, where, at least now, honestly, maybe most people think of oh, I’m using Transformers. Specifically in my case, and in the Speakerbox case, we built it off of Transformers because there are tons of existing models that we can use to make this process of speaker identification or speaker classification much, much, easier.

Specifically there are models already pre trained and available on I think it’s called the Vox Fox something data set. I forget what the name is. But the general idea was that, people have already trained a transformer model to identify something like 1000 different people’s voices.

And to my understanding, they’re all celebrities, right? So you can imagine this. training process as, okay, given a bunch of different audio samples of celebrities voices, can we identify which celebrities voices these are? In our case, we don’t care about the celebrities, right? For many, if you want to train a model for your lab or you want to train it for a city council meeting or whatever it is you care about the people in your meeting.

And so the process there is called fine tuning, where really the main task is you’re taking off that last layer of the model and instead of saying, I want you to pick between one out of a thousand celebrities, I want you to pick out of one out of in this call, I want you to pick out of one of three people, right?

And by doing so, you can quickly give it some extra samples of data. So we can give it a couple, let’s say 20 samples of Arfon speaking. We can give it 20 samples of Abby speaking. We can give it 20 samples of me speaking. And hopefully just because of the existing pre training it, it’s learned the basics of how people speak and how people, converse.

And all we’re really doing is saying. Don’t care about the basics of how people speak and how people converse. Really focus on the qualities of this person’s voice and the qualities of this person’s voice, and so on. Hopefully that answers the question.

[00:11:50] Arfon: is that why, is that, those examples, is that what we mean by the few shot learning? So you give a few examples, effectively, of labeled This is Abby speaking, or this is Eva speaking. is that what we mean by the few shot?

[00:12:03] Eva: Exactly. In our case we’re doing a little bit of a hack. We say few shot learning, but it’s actually not really because it’s very common for few shot learning to literally be five examples or something less than 10. But really because audio samples can become really long, like I’m answering this question in a one minute phrase or a one minute response we can split up that audio into many, smaller chunks that act as potentially better smaller examples.
And so you may give us five examples, but in the process of actually training the model in Speakerbox, we might split that up into many smaller ones. So we’re kind of cheating, but,

[00:12:38] Arfon: Interesting. Thanks.

[00:12:40] Abby: Yeah, that’s awesome. And I do really like the, the Hugging Face platform, where there’s so many models that you can tweak. Are you sharing back to the Hugging Face platform at all? Or,

[00:12:50] Eva: I haven’t, I’m not. We have used some of the models trained from Speakerbox in our own work with Council Data Project specifically for the Seattle City Council, because that’s, my home council. And we’ve trained a model to identify city council members for Seattle, and we’ve applied it across a number, I think, almost 300 transcripts.

But that was just purely for analysis. We haven’t pushed that model up or anything, and we probably should. That should probably even be part of the process or part of the pipeline. Yeah, to come. I’m sure other people who may have used the system might have or something, but,

[00:13:20] Abby: you did publish in JOSS, the Journal of Open Source Software. Can you talk a bit about why you decided to publish there, and how the experience was?

[00:13:27] Eva: Yeah. So to be honest, I love JOSS. This is my second article in JOSS. We’ve also published one for the larger ecosystem of Council Data Projects as well. I think after the first time publishing with JOSS, I was just really, really, impressed with the reviewing process. More so than some other journals, right?

I think the requirements of having reviewers try to use your package that you’re pushing, saying this is ready for use, or this is useful in X domain. I think having reviewers try to use it is such a it’s such a good thing to do simply because one, like we know that it’s available, we know it’s documented, but also we, know that it works and I think I’ve just been frustrated by the number of times I’ve seen a paper on arxiv or something that links to their code, but there’s no real documentation or something.

And so, to me, it’s like I don’t know, I, just, one, like to keep supporting what JOSS does because I really appreciate, I’ve, read a number of papers from JOSS where I’m like, that tool seems great, I’m going to go use it, right? Yeah, I don’t know, it just, it seems like the default place in my mind now, where if I want, to come off as like a trusted developer of software or a trusted developer of research software, that’s the place to go.

[00:14:35] Arfon: Yeah, and it’s funny you say having people actually use your software. This is one of the founding questions that a few of us were asking, which was, it was possible to publish a paper about software in some disciplines, but the
question I was asking people on Twitter at the time, I’m not really very active there now, but was like, when you have managed to publish a paper about software, did anybody look at the software?

And people were like, nope. Universally, nobody ever said yes. And I was like, huh, that seems really weird to me, that you would publish software, but nobody would look at the software, that seems. I’m glad that resonates for you and provides value.

[00:15:17] Eva: I was gonna, I was gonna add, I think the open reviewer process is very, very, nice. Because at least when I was writing the first paper that we wrote for JOSS, I could go and look at, other review processes and say, okay, what are they expecting? What might we need to change ahead of time, etc. And I could just, prepare myself, especially as like a, now a PhD student, sometimes publishing is scary and being able to look at a review process drastically lowers that concern or that worry as well.

[00:15:47] Arfon: Yeah, definitely. I mean, it’s good and bad with open review. It’s good for transparency and people are generally very well behaved. It’s bad because it’s, I think, amongst other things, intimidating is probably, it could be exclusionary to some folks who don’t feel like they could participate in that kind of review.

I’m, but on balance, I feel like it’s the right decision for this journal. I definitely think it’s an important part of what JOSS does.

I was going to ask if there was anything it sounds like you’d eyeballed and seen some reviews before you’d submitted, but was there anything particularly important about the review for Speakerbox that was a particularly good contribution from the reviewers or anything that was surprising, , anything particularly memorable from that review?

[00:16:33] Eva: Speakerbox , because it’s designed to act in a workflow manner, it’s not so much, a library of here’s a single function or here’s like a nice function, two or three functions to use for your processing. It’s very much designed as a workflow, right?

Where you say, I have some set of data. I need to annotate that data first and then I need to, prep it for training. And then I need to finally train the model. And I might need to evaluate as well. And because it’s laid out in that workflow kind of style. The reviewers were having trouble from a normal package where you say, I’m just going to go to the readme, look at a quick start, and use it.

And I think that, honestly, the biggest piece was just, the reviewers just kept saying, I’m a little confused as to how to actually use this. And ultimately I very quickly made a video demonstrating exactly how to use it and I think that was the best
I see your confusion here. And ultimately leading me to just be like, okay, I’m going to take everything that they just said, make a video and also take any of the things that I did in the video. And add them to the readme, right?

By literally forcing myself to do it on a toy data set, I can add all the little extra bits of places that were possibly confusing for them as well. Using my own data versus using some random other data, there’s always going to be, things that you forget to add to the readme or whatever it is.

[00:17:46] Abby: That’s awesome. And one thing I really, I do like about the JOSS Review System is just how it makes, it just adds so many best practices for open source and makes it so much easier for others to, to run it and to contribute. So is Speakerbox open for a contribution if people wanted to jump in? Can they?

[00:18:02] Eva: Yeah, totally. Speakerbox. Yes, I think there are some existing minor, little bugs, I think available. There’s also probably tons of optimizations or just UX improvements that could definitely happen. Yes, totally open for contribution, totally happy to review PRs or respond to bugs and stuff.

[00:18:20] Abby: That’s great, especially since you published this a while ago, that you’re still being responsive there on GitHub. But what skills do people need if they want to jump in?

[00:18:28] Eva: I think the primary thing, there’s maybe two areas, one is, I think a very good background in let’s say data processing and machine learning in Python. Like those two things together as a single trait might be nice. Or on the opposite end of things, if you just want to try and train your own, like just use the system as is. You might experience bugs or something and just making a documentation post, right? like say "I experienced this bug, here’s how I got around it". Or, posting a GitHub issue and saying, there’s this current bug for someone else to gather. I think truly trying to fix stuff if you have a machine learning background or just trying to build a model are both totally welcome things.

[00:19:06] Abby: That’s awesome. Yeah, and I love how you’re seeing like using the model as a contribution, and just like documenting what you run into.

[00:19:12] Arfon: So I was gonna I’m gonna put my product manager hat on for a minute and ask you what questions should we have asked you today that we haven’t done yet?

Arfon: Okay if there isn’t a question, but it’s the magic wand question for, what, could this product do if we if it could do anything? What should we have asked you as hosts that we didn’t, that we didn’t check in with you about today?

Eva: I think it’s not so much a question, maybe what’s like the feature that you dream of,

Arfon: Yeah,

Eva: if, it’s true, if it’s truly like a product manager hat on then I think the feature we’ve previously talked a lot about okay, the, workflow itself is pretty simple and there are pretty good packages nowadays for shipping a webpage via Python. And it’d be very nice to just have the entire workflow as a user, like as a nice user experience process, like you you launch the server on your terminal and then you can do everything in the webpage especially for the users that aren’t as comfortable in the terminal, right?

That would be very, very, nice. And something that I just never got around to. it’s not my priority. But something that we’ve thought about as it would definitely reduce the barrier of use or the barrier of effort or whatever.

Abby: Yeah. Oh, that’s really cool. Especially in the civic tech space, I can see that being like really game changing for a lot of cities.

Arfon: Yeah.

Eva: Yeah. I think civics and journalists as well, I think. But yeah.


Hey, Eva, this has been a great, conversation. Thanks for telling us about Speakerbox. Just to close this out, I was curious how can people follow your work, keep up to date with what you’re working on today? Is there any place you would want people to go to keep track of what you’re working on?

Eva: Yeah my website is evamaxfield.github.io. I will occasionally post updates on papers and stuff there. I think Twitter might be the best place you can find me on Twitter at @EvaMaxfieldB as well. So

Arfon: Fantastic. thanks for being part of the podcast. It’s been really fun to learn about your software and thanks for your time today.

Eva: yeah, thanks for having me.

Abby: Thank you so much for listening to Open Source for Researchers. We love to showcase open source software built by researchers for researchers, so you can hear more by subscribing in your favorite podcast app.
Researchers is produced and hosted by Arfon Smith and me, Abby Cabunoc Mayes, edited by Abby, and the music is CC-BY Boxcat Games.

[/expand]

Video
Er is een fout opgetreden.

Bekijk deze video op www.youtube.com of zet JavaScript aan als dit uitstaat in je browser.