EPISODE 1254

[INTRODUCTION]

[00:00:00] JM: Natural Language Processing is a branch of artificial intelligence concerned with giving computers the ability to understand text and words. Understanding includes intent, sentiment, and what's important in the message. NLP powers things like voice-operated software, digital assistants, customer service chat bots, and many other academic consumer as tools. The company bought press provides open source developer tools to create NLP tools for process and FAQ automation. They use the latest NLP models for domain-specific, contextual and goal-oriented conversations. This technology is free and available through simple API routes. They also maintain integrations with popular messaging services like Facebook Messenger, Slack and Microsoft Teams. For other proprietary systems, they provide a raw messaging API.

In this episode, we talked to Sylvain Perron, CEO of Botpress. Sylvain was previously a director of engineering at Protorisk and a software developer at ArcBees before that. We discuss the current advances in natural language processing and how NLP empowers Botpress.

A few announcements before we get started. One, if you like Clubhouse, subscribe to the Club for Software Daily on Clubhouse. It's just Software Daily. And we'll be doing some interesting Clubhouse sessions within the next few weeks. And two, if you're looking for a job, we are hiring a variety of roles. We're looking for a social media manager. We're looking for a graphic designer. And we're looking for writers. If you are interested in contributing content to Software Engineering Daily, or even if you're a podcaster, and you're curious about how to get involved, we are looking for people with interesting backgrounds who can contribute to Software Engineering Daily. Again, mostly we're looking for social media help and design help. But if you're a writer or a podcaster, we'd also love to hear from you. You can send me an email with your resume, jeff@softwareengineeringdaily.com. That's jeff@softwareengineeringdaily.com.

[INTERVIEW]

[00:02:06] JM: Sylvain, welcome to the show.

[00:02:07] SP: Thanks for having me.

[00:02:09] JM: You are the founder of Botpress, which is a system for building chatbots. And whenever I do a show about chatbots, the first thing I always like to ask is what happened? Like there was the period of great chatbot hype. And then things basically died down. But you still see chatbots everywhere, when you hit an airline website or a customer service center. So how big is the market for chatbots? Is there still a big demand for chatbots? What's going on there?

[00:02:44] SP: Yeah, for sure. And I think pretty much everybody's wondering what's happening with the bots. There was indeed a big hype. And I think the main problem there was that it wasn't really backed by any kind of significant breakthrough in technology, right? So it was mostly based on Facebook opening up their developer API's and allowing people to build those, I don't know if I should say it, but dumb bots with buttons that didn't really have any kind of language understanding behind them. And so I think the hype kind of fell flat. And, right now, I think that it's starting to pick it up again because there's been indeed like recent developments in NLP, which now allows us to start again looking at this problem and having more concrete applications.

And like the market is very big for something that works. So the market is expected to grow into 25 billion in 2025. And, again, growing 40% year over year to reach 48 billion in 2028. So there's a big market. And the value for business is quite clear. It allows you to basically take human-to-human conversations, which are very hard to scale, and replace them by machines with infinite scale, right? And they don't take pause. They don't take naps. They don't make mistakes. And so that's a very attractive return on investment for businesses, given that they work, which we've had a big struggle getting them to work up to now.

[00:04:49] JM: What are those struggles of getting chatbot systems to work?

[00:04:55] SP: I think there're multiple facets to that problem, and the first one is NLP. So NLP is in a big kind of improvement – In the big improvement cycle right now, but basically you're trying to understand what the user said. And then you need to interpret and deduct like a meaning, and that has to be contextual. If it's a question you have to find the right knowledge base and find the right answer within that knowledge base. And then you have to generate that human-like response. You have to do all of that contextually depending on what type of business and what's your customer use cases, and you have to do that in hundreds of languages.

And, obviously, we train NLP systems on very clean data sets. And problems start to happen when you train, like when you try to apply that to user utterances. Because people make mistakes, people talk in ways that it's not written like this on Wikipedia, which is how we train the algorithms right now, the language models. And so I think there's a big gap in the data sets that we have and the sort of tasks that we train those NLP models on, and how people really talk in real life. And you have to do that for hundreds of languages. And so that's a big challenge in itself.

The second one is you have to build that messaging infrastructure. It's not only about the NLP. You have to connect with all those different chat platforms. And you have to scale that and you have to maintain the features that those different chat platforms offer you. And also, another challenge is that this is a new kind of application. Frankly, nobody knows how to build those applications as best as possible. There's no standard. There're no clear guidelines. And so we're all sort of in this new learning mode sort of like mobile apps back in 2008.

[00:07:16] JM: Can you give me a little bit idea of the tooling around chatbots? Like there's Facebook Messenger, and Intercom, and Drift, I know, that can be used for various chatbot applications. But are you providing infrastructure that plugs into those different systems? Or just give me the lay of the land in terms of tooling that people use and what you're plugging into.

[00:07:41] SP: Yeah, for sure. So the first component is the messaging platforms itself. So WhatsApp, Telegram, Facebook Messenger, Microsoft Teams, Slack. And so that's kind of the user interface that end users actually use. And so you have to connect to that platform. So that's the first step. And the second part is the actual engine, the chatbot engine, that will understand the language, generate a response. Potentially integrate with your various in-house systems and then spit back a response. And that's the layer we provide.

So our platform, essentially, is a developer stack that provides connectivity to chat platforms that provides natural language understanding. Soon we'll offer information retrieval and answer answering – Answer a question answering engine, and also a WordFlow automation, so that you can actually do things not just answer basic FAQs. So these are like the four major components of a complete chatbot platform. And then once you published your bot, you're probably getting a lot of messages that you're bot didn't understand. And so there's also a suite of tools to allow you to reclassify, re-label those messages, understand what's going on with your users. Maybe pick up new trends, new questions that you haven't thought about. And so we also ship past deployments, a module called misunderstood, that allows you to re-label that data.

[00:09:27] JM: Can you give me an example of a chatbot application that somebody might build using Botpress?

[00:09:36] SP: Yeah, the most obvious use case right now is support automation. So contact center automation and service desk automation. So imagine you have a big contact center with hundreds of people, you'll find that probably 25% to 50% of all the queries that your support agents answer are basically very repetitive and low-level – Well, high-level. Very repetitive kind of questions that could be very easily understood by today's chatbots and automated completely. So that's providing a really good user experience, because you're not in line waiting for an agent and you get an instant answer.
Some chat bots that are a bit more elaborate today can also do basic tasks that a human would normally do. And so this involves multi-turn conversation with the user to ask additional information, execute certain state statements on backend systems and then fulfill the user's request.

[00:10:54] JM: And how does the human in the loop play a role? Like how do you build a system where, if necessary, the chatbot can kick a request to a human?

[00:11:09] SP: Yeah, so that's a very important part of any chatbot system. You have to assume that the bot won't be able to handle 100% of the questions. I've never seen a bot that achieves that level. So it's really important to loop in the human when possible. And there're kind of two modes. The first one is online, where if the bot doesn't know the answer, it can sort of route the chat over to human for live handoff. And so the human will just pick up the conversation and help the user. The second mode is offline, where if the bot doesn't know how to help, it will create a support ticket. And so it will be picked up at a later time by a support agent. And that really depends on how you want to do it as a business, how you operate. What's your existing kind of process like? And both are, in my opinion, super necessary. And we do provide – Inside the Botpress stack, we have a human in the loop module that allows you to do online escalation. And that's super helpful, because you can kind of see the conversation that the bot had as a human, and you can also train and re-enable those utterances that the bot didn't understand live as a human agent. And so you can kind of make it smarter on the go.

[00:12:42] JM: What role does natural language understanding and natural language processing play in building an effective chatbot?

[00:12:53] SP: Well, that's the most important part. That's like the brain of your bot, right? So if it doesn't understand, then mostly the user will have a very bad experience. And so that's super, super critical. And it’s, to me, like the first step that you have to get right, but it's not everything. So you need to be able to identify what the user said in terms of intention. So identifying what's the real – Like what the user actually wanted to accomplish. That's the first thing that your bot should try to do. Then you can get into, once that's done, a bit more advanced use cases like extracting the parameters of the user's request.

So for example, if the user intention is, “I want to book a flight from Quebec to Montreal.” Then the intention is book flight. And that's the first thing you have to extract. But in order to successfully fulfill that book flight, in a way that the user will be satisfied, because he did provide the destination and the departure. So you have to be able to extract those two parameters as well. If the user asked a question, then you have additional work to do. So for example, if you ask a question like, “Can I bring my beta board in flight da-da-da?” Then the answer might not be that straightforward. And you'd likely haven't created a pair of question answer for every different scenarios. But you probably have like a pet policy on your website or knowledge base somewhere that we can query. And so the NLU engine also has a component of question answering information retrieval where you'll need to find that article and you'll need to extract the answer out of this knowledge base. And then you'll have to generate a human-like response back to the user. And so that's slightly more complicated. But this is where this year and next year we'll see more of those platforms like Botpress. We're currently in to works of that where we can see the question answering algorithms being like really good. And we, in the lab, see really good results. And so this is kind of like the next generation of bot where they'll be able to answer very complex questions.

[00:15:41] JM: And can you tell me more about your NLP, NLU stack? Like are you taking libraries off the shelf? Or like how do you train your libraries? Just tell me more about what you've built or what you've taken off the shelf?

[00:15:55] SP: Yeah, for sure. So we used to do everything ourselves in terms of training our own algorithms. And that's because back in 2017 there weren't much like language models out there that were pre-trained and that worked great. Today, we're mostly using pre-built language models, because the research is evolving so quickly that we take the pre-trained language models like fastText, GloVe, but also like bigger generative models like T5, and Big Bird and GPT, and we fine tune those models. Often those models will serve as features or inputs to our own in-house models that will accomplish their tasks based on those models. So we're using transfer learning in some way to accomplish different tasks. But their reality is that there's just so much tasks that we have to do. And that's the very complicated part, because every task that we train has a certain percentage of error or accuracy. And that error gets spilled over to the next task in the pipeline.

So for example, if you have a language model trained for French, then that language model obviously won't be perfect in the way that it understands French. And so that language embedding will spill error down to all the tasks below it, or that uses that as its input. And if you have, for example, type of detection or spell checking as part of one of your first tasks in the pipeline, and task has, I don't know, like 10% error rate, then this 10% error rate will propagate down to all the other tasks later in the pipeline. And that's really like the challenging part of what we have to do here, because all these errors propagating down leads to pretty bad results, like the amplify, as they go down. And the result sometimes is really bad all because you had a very bad input at the very beginning of the pipeline. So we try to stay state of the art and to pick up like the latest trends in NLP. And so a big part of that is being able to switch those components as they get released to test on different datasets and to assess the quality of those tasks in the downstream result.

[00:18:45] JM: What are some of the prototypical problems that exist for chatbots today that you think will be ironed out in the near future?

[00:18:55] SP: Yeah. I think one of the big example is question answering right now is not part of the chatbot world. And that's being solved right now. So most chatbot will detect that you have a question about a certain topic and it will sort of answer with a very vague and very generic answer to that question. And the ability to understand deeply what the question is about and also pick up the subtleties in that question. And that's being solved-out right now.

Another exciting trend is to be able to query using natural language structured tables. So imagine you have a database table about flights. The ability for the user to query that table without having to train any kind of example on it before is really powerful. So think about like all the BI applications will be using that. And if you've used Google Analytics recently, you can see at the right hand side, you have kind of like an input text box where people can ask questions using natural language. And that's been working quite well actually recently. And so we'll see more of that.

The other big thing is generating responses that match the personality of the bot. And so, right now we're just getting with GPT3, like pretty decent generation of text. We can see blog articles being published that looks fairly human. I mean, there are some quirks still, but I think it's getting really good. And the next kind of step there is to be able to personalize the kind of output that will sort out so that you can match your company's brand. And imagine you're creating a chatbot inside a video game, say, GTA. I don't know what number we're at, but it's like, say, GTA 8. Then you have personas that can talk like gangsters, for example, right? And so the ability to generate language that has a personality attached to it is also very exciting, especially for brands, where they want to differentiate themselves from all the other brands out there, it's important to have the ability to customize the accent of the bot so that it doesn't just look like bland, just like all the other brands out there.

[00:21:47] JM: Describe how you've differentiated yourself from the other bot platforms out there.

[00:21:56] SP: The biggest competitors are the Google and Microsoft, where they offer natural language as a service. And I think that's very difficult to get something high-quality out of those service. And that's kind of like the Firebase approach versus Postgres, whereas with Firebase, like you don't control really well like all the configuration and options. And the way Botpress works is, and we're sort of the only one that does that, it's an open source stack. You run it on your computer. You can actually customize everything behind. And so you can really get the extra juice out of the engine. You can really fine tune anything you want. And also, the other advantage is that you can actually host that platform anywhere you want. So if you want to deploy on AWS or on Azure, you can do that, whereas if you go with the major cloud platform, you're actually stuck with that vendor. And so it's not very flexible. And so imagine you're your bank or healthcare provider, the idea of streaming all of your customers' interactions over to Google might be frightening. So for any kind of application, I think developers want this kind of experience where they have control over the stack. And I don't think it feels natural to use just like an HTTP service that does that for you and you have no control. It's like a black box and anything can break at any moment. With Botpress, it's much more natural. It feels like regular software.

[00:23:50] JM: So you have a system, like a UI for designing chat workflows. Can you talk a little bit more about the design of the UI and how people within a company use it?

[00:24:05] SP: Yep. So the UI is used for three things. The first one is labeling the data. And so you have – In order to train the bot to understand certain intentions and use cases, you have to basically classify and label those utterances and entities into specific boxes. And so we provide a UI to do that, because using markdown in JSON is pretty hard, especially for end users. If you're not going to do the labeling yourself, you probably want a user interface to be able to do that. And so that's one part. The second part is designing the workflows. When you're designing multi-turn conversation, it gets pretty – If you're doing through code with a bunch of if and else statements, you end up with like a very big kind of spaghetti and hard to visualize workflow. And so and that's really hard to refactor. Like however you turn it, it's messy. And so we found it's much easier to design workflows visually to kind of have this high-level picture of how the conversation can and will go.

The third interface that we have is a conversation debugger. So just like you would have in Visual Studio where you can step into line by line and function by function, and inspect the data as you go, we have the exact same thing for a conversation. So as the messages gets processed in and sent out, you can kind of step in every step of the pipeline for the NLU, but also the response duration in the workflow step so you can troubleshoot incoming problem and fix that.

[00:25:56] JM: So what aspects of building my chatbot do I need to use a workflow editor for and what do I need to use actual programming for?

[00:26:08] SP: Yeah. So say you want to automate employee onboarding. So you have a new employee coming into your company and you want to provision, I don't know, licenses, do a bunch of software, and you want to give them a nice experience, where it feels like an HR person guiding you through all the different software you have to set up and everything. So that's a fairly complex workflow. And you would design all of this visually where you would have integrations with the software to generate the licenses and then you would – As a developer, you would write code to generate those things and actually like do actions on those backend systems.

And so, having a visual editor really helps you as a developer just visualize the whole flow, but also helps non-technical people or subject matter experts to see what's going on, and sometimes even modify the workflows themselves. As a developer, you have to build all the different integrations with those systems. So if you have to provision, I don't know, like a Google Suite user, then there's code to write there. And that's a developer that has to do that. And within the UI, we provide like a code editor, but you can totally use like Visual Studio. And so everything you do in Botpress, basically, the UI, just generates files, JavaScript files, and JSON files, so that whatever way works for you, you can do through code, or you can also do with the visual editor.

[00:27:57] JM: How has Botpress evolved over time as you've engaged with different users of the system and you've seen people build different kinds of chat bots? How has the product evolved?

[00:28:13] SP: Yeah, at the beginning, it was just an open source project. And this is how it got turned into a company, is we only had this backend kind of framework with no UI just the way to design conversation and kind of catch and understand just like Reg X's to understand the messages and provide a response back. And actually, like a fortune 500 company called me and said, “I like the framework, but I would like – My conversations are getting a bit messy. I would like to have a visual WordFlow editor. And I would also like to have SSO and, and this and that.” And so we started working with that big company for many months trying to cater to their needs. And so we designed the UI there to help their employees to manage the conversations. So that's just one example. And as we go with customers, we kind of see trends where the bots become more and more evolved over time. And so we kind of adapt as we go, working hand in hand with the customers. As they face problems that they can't solve with today's technology, this is where we innovate. And that's where like we had those features. And today, like most customers can do basic FAQs. They can do simple WordFlows. But the thing that they can't do is answer very complicated question. And that's one example of it. And right now we can fix it with today’s NLP algorithms. And so this is the current focus. But we're super kind of customer-driven where we work hand in hand with them and see what the bottlenecks are. And when the technology evolves, and we see a fit, then we develop that new feature for the customer.

[00:30:23] JM: How do you see the chatbot category evolving in the near future? What predictions do you have about what people want out of chatbots and how consumers will be engaging with them?

[00:30:37] SP: I think conversational applications will be one of the main way you interact with software, once we nail it, which should be within two or three years, I would say. Because the way – When you think about it, the way that you build software right now is kind of broken. A customer face an issue and then you build technology that solves that pain. And then once that's done, you wrap it up in a nice GUI. And then you write documentation for that software and then you ship it to the customer and you iterate from there.

But by doing it this way, you did eventually create a layer of friction between the pain and the cure, right? The customer had that pain, and then you solve it in your own way. But by creating a GUI, you've essentially created something that isn't necessarily natural to that user. And you had to write documentation on how to use it. So conversational UI is in a way the ultimate user experience where the user tells you what he wants. And it's your responsibility as a developer, as a business, to actually go and solve that specific query that the customer has. And so the responsibility is shifted from the user having to learn the machine over to the machine has to learn how to understand and serve the user. And that's a very big paradigm shift. But once the machine pick up language and become much better, ultimately, that's going to be the new standard for how to consume software. And this will lead to many, many applications that we simply can't do today. Many things are not digital yet because of that very limitation that we have with GUI. And for example, that's the difference between going on bestbuy.com versus going to BestBuy in store, is that when you go into store, you have those people that helps you that knows about the products in certain categories, and ask you questions about your budgets, and your use case, and the size of your apartment, and this and that with preferences basically. And they give you the right product for your specific needs. And that has to be conversational in order to figure out all these things about you and what you like.

And so, as a user, being able to express yourself is very important. And that's something very hard to do with buttons and dropdowns, and mouse clicks. But if you can speak it, and the computer is able to pick it up, then that's really exciting. And it opens up a whole lot of application that we just couldn't do before. So I'm very excited about this, because we'll have video games that are conversational in nature. This will be really exciting. You have professional services, just like psychology, and lawyers will be able to basically offer their service through a digital assistant. And this will make it much more accessible to everyone to consume. And yeah, and that's pretty much it. Yeah.

[00:34:53] JM: So what's the pace of innovation there? Because it feels like – I mean my sense is that the state of the art or bot interactions as improved at a snail's pace ever since bots were a thing, four years ago, or a mainstream thing. When do we get this kind of next level step function integration? How long till that actually comes to fruition?

[00:35:23] SP: My estimate is by 2024. And the reason why – It's really improving exponentially, and people don't really see it in today's bots, because the challenge remains that the NLP models are evolving, but they're actually really hard to productionize. And when we'll be able to productionalize those models, then that's where people will see whatever has been going on for the past two years. And also that research is really evolving exponentially, because there're big advancements in hardware. There're big advancements in infrastructure, machine learning infrastructure. There's big investment in just software, and delivery, and how easy nowadays it is, just like there's Hugging Face, which is doing an awesome job of just abstracting all of those models, and that's just one company, but there's also NVIDIA that does pretty gnarly stuff in there as well. And there's also the data that is completely exploding right now. So those four things combined together is really what's propelling that exponential improvements in NLP.

But still, we're really focusing on individual tasks. And so we're getting super good at machine translation, super good at question answering, super good at generating text, and sort of all these things individually. But nobody's ever, ever put all of these tasks back to back so that they can work in a coherent way and offer a good customer experience. And that's really the challenge that we're trying to do at Botpress is saying, “Hey, okay, let's put all of this state of the art together and make it abstract so that developers don't need to learn those things. And they can actually build something very quickly and deliver to end customers.” And I think it's not really a problem of right now the AI, because it's there, and it has significantly improved. But it's really hard to deploy it to end customers, because it requires you significant resources. Siri, today, I think is one of the best digital assistant. It's not like it's mind-blowing. It does pretty decent at answering requests, but you have to know that there's like 800 people working behind the scene on Siri. And so that's not accessible to most companies. So we're trying to cut that down to a single developer can build a Siri-like experience and even better, because I don't think that Siri is state of the art.

[00:38:17] JM: Well, since you mentioned Siri, is there a connection between chatbot applications and voice interfaces?

[00:38:27] SP: There is. So the definition there, it gets a bit blurry. But a chatbot is essentially a software that communicates with humans using natural language. And that can include in its definition voice. But we see it's slightly different where ASR, automatic speech recognition, and text to speech, are actually two tasks that are getting pretty good. And we don't consider that to be a significant problem in the near future. I mean, that problem is considered solved. We're just trying to get it slightly more accurate, but it's there, and it's working. And there're many companies that tackle that problem. So it's just a matter of integration, right? And I think most of the messaging channels will be offering out of the box ASR in text to speech. So for us, it's not a big priority on the roadmap, because we think it's commoditized, and will be even more down the road. The real, real challenge is actually understanding the real meaning of text and spinning back like a natural response.

[00:39:53] JM: As we begin to wrap up, maybe we can talk a little bit about your infrastructure. Can you just tell me about the building blocks of Botpress? Tell me about some infrastructure decisions you've made and the architecture for the company.

[00:40:06] SP: Yeah. Well, since it's on-prem, and it's open source software, and you have to be able to run it on your computer, this led some of the infrastructure decisions that we made early on where we wanted to be as simple as possible for developers to get started. And that means very minimal kind of requirements. The stack is built on NodeJS and TypeScript. But we wanted to make it so that developers don't have to install the NodeJS runtime, read the right version, and have all of those, because we're using machine learning as well. And so you need like node bindings, compiler with the right version for your kernel, etc. So that was creating a lot of issues on GitHub. And so we decided to make it so that Botpress is just like a binary file where you double click on it and it just works. And so we're using a package created by Vercel, which is awesome, **[inaudible 00:41:07]**, where you can have wrap NodeJS inside your binary so that it's self-contained. And that has worked pretty well for us so far. No issues on people trying to set up Botpress anymore. Really trying to simplify the developer experience is our main focus, which led us to, by default, having no requirements for database. So we use SQLite by default, for messaging, queuing, and retries, and caching. So by default, we do in-memory cache. And when you deploy to production, you deploy the exact same binary or Docker and you simply have to provide a database URL, which is Postgres and the Redis URL to scale the caching layer, and everything works out of the box. And so the same environment that you downloaded locally and you ran locally on your computer trains the AI, generates models, and you can take the same binary and the same models, put them on the cloud, and that should work just as well. You don't have anything special to do.

And we also wanted to make it really simple to scale. Some of our customers have the requirements to handle millions of conversations per day. And that's quite challenging to do with a very simple architecture. But with just Redis and Postgres, we've managed to make it horizontally scalable. And that's been quite successful as well for us, because there's only one architecture, one way to deploy it. It runs fast, and it works on everybody's machine. Obviously, this is just a game of trade off. The tradeoff that we've made is this prevents us from using Python, for example, as a language for machine learning. And that was something that we were initially okay with, because the models that we were using weren't that's performing less than what we would get with like a PyTorch.

But as we evolve, we see less and less companies having a need, especially in the Fortune companies where they have like a pretty archaic infrastructures. But we see less companies that are trying to stay away from containers. And so as containers become more mainstream, we are evolving into microservices architecture based on those containers. And we kind of – This opens up like much better and much easier kind of architectural choice for us where we can use – Finally can use Python and all those fancy libraries that we just can't use before just for developer convenience. So that's currently in the works this year where we will transition to a container-based strategy.

[00:44:27] JM: Great. Well, it's been a real pleasure talking to you. Thanks for coming on the show.

[00:44:31] SP: Thanks, Jeff. It's been a pleasure.

[END]