1:22

Alexey

**This week, we'll talk about dataset creation and annotation – creation, curation, and annotation. We have a special guest today, Chris. Chris has six years of experience delivering natural language processing tools and services, including emails, compliance, pharma, sales industry. He has a lot of experience. I had a chat with Chris a while ago and we got to talking about this topic and I understood how much we do not talk about these things. We usually talk about models, we usually talk about things like “Okay, we have a model. How do we deploy this model?” But we usually don't spend enough time talking about dataset creation. And we thought that it would be a good idea to record something about this. That's why we have Chris today with us. Hi, Chris!**

2:19

Christiaan

Hello, Alexey. It's a real pleasure to be here. And thank you for inviting me.

2:24

Alexey

**Before we go into our main topic of dataset creation and curation, let's start with your background. Can you tell us about your career journey so far?**

# Christiaan’s background

2:33

Christiaan

Yeah. I studied AI at the University of Edinburgh. And I fell in love with natural language processing while I was there. I think it's extremely interesting “How can a computer understand language?” from a philosophical perspective. I think it's a really, really interesting question. Throughout my career, I've just been trying to work on this question and see, “What kind of understanding can we give a computer? What kind of automated decision-making can we do?” So after finishing my studies, I worked for a small email provider, where I worked on extracting different information from emails and email classifications. Then I worked at Resolver, which is a large complaint company that had over 10 million complaints in the UK and worked extremely closely with complaint departments and even the complaint authorities in the UK. So it was very interesting from that type of regulatory perspective.

After Resolver, I worked for two years at Healx, which is a rare disease drug development company that focuses on rare diseases and uses machine learning techniques to identify drug candidates in order to take to the market for patients who have rare conditions. That was really interesting in introducing me to the whole world of bio-NLP, where there's way more datasets than I think other NLP areas have due to high levels of curation. But they have their own challenges due to the ambiguity of biology added on top of the ambiguity of natural language.

Finally my journey led me to co-found Comtura, which is my own company where we work on making sure that you don't need to administer your Salesforce. We help sales teams capture useful information from sales calls, and then we extract all the useful concepts and push them into Salesforce for you using our interface. I think one of the key takeaways I had from working at Healx is that some of these technologies are extremely challenging when you need to apply them in a safety-critical environment like drug development.

So I've been enjoying the less pressure required in the sales environment because, at the end of the day, patients’ lives aren't at risk for the decisions that you're making and whether a deal will come through or not. But sales processes are also extremely interesting from an NLP perspective, as there's a lot of communication going on.

5:12

Alexey

**What led you to actually start this company? What kind of problems did you see that made you realize “Okay, now it's time to start this company.”?**

5:22

Christiaan

So, I am a massive data nerd, as you can tell by the topic of this conversation. I love to think about “How do you create datasets?” Actually, the huge, huge reason I like the idea of Comtura is because it intersects with transcription technology and CRM technology. We capture what is said in the trenches, what it said during the calls, and we help push that into Salesforce, into the CRM, where business stores all its commercial intelligence. It's extremely interesting from a supervised machine learning perspective, almost – you've got the labels and the supervised stuff in the CRM, and you've got the unlabeled source data and transcripts potentially.

There's a number of other views you can do this from, from a machine learning perspective. So I was like, “This is awesome. There's so much data available here. There's a huge impact, potentially on sales teams as well.” Selling is extremely hard – making and building up the connections and actually making sure it's a good relationship. The whole workflow is very difficult when you need to capture pages of information during a sales call and you also need to build up a relationship. The idea was “Why don't we do things that computers are good at – capturing the information – and let people focus on building relationships and selling better?”

# Usual ways of collecting and curating data

6:51

Alexey

**You mentioned that there is so much data available, but sometimes the case is that we have data in abundance, but we don't really know what there is, how to process this data, How useful this data is, and we need to somehow curate this data to make use of this – to somehow understand what is good and what is bad. What are the usual ways that we collect and curate data?**

7:20

Christiaan

I like to think about it in three general terms of dataset creation – automated, manual, and hybrid. In an automated form, you may scrape, for example, some kind of database or a number of websites to collect some data automatically. In the manual approach, you may have a number of documents, such as sales transcripts, for example, and you decide to annotate key sales concepts from them. And in a hybrid approach, which I think is becoming ever trendier in our industry currently, you mix these two together. So you use the power and scale that automation can bring you to make sure that your manual efforts are focused on the most valuable data.

8:09

Alexey

**So first you do some annotation manually, like you said with a sales call, for example. So we record the call, make a transcript, and then somebody needs to go there and say, (I don't know what exactly you do there) “This part is a good indicator that the sale will close.”**

**So first, you do this manually and then you train a model so that the next time it kind of already recognizes these clues in the transcript, and then you can train another one based on the data you collect. Maybe a person can review this and say, “Okay, this is accurate. This is not accurate.” You refine your dataset and over time, your model becomes better and better and your dataset also becomes better and better. Did I get this right?**

9:02

Christiaan

Yeah, I think this is the bottom-up view. But I think the top-down view is what I think most data scientists struggle with, actually. And I think I've made most of my mistakes from having this kind of bottom-up view rather than a more top-down view of this. I think it's really important to manage upwards and it's an extremely important skill to develop. The way I would tie that back to dataset creation is that, “What's the business value that this dataset creation will empower?” Usually if you create the dataset, there's going to be model deployment, and it will realize some kind of business value for you. I think what's really hard is that, until you have the data and until you have the model deployment, you can't really know whether the business value will be realized. That's why this is fundamentally a high-risk enterprise.

I think what's really important is de-risking this. When I do my initial annotation, I've shared that with CEOs and other execs, when I do that, I will literally walk them through this, “This is literally our source data. Do you think if we could automatically map it in some way, or do some transformation on this – do you think this would be valuable?” I personally think the hidden art in dataset creation is actually that there's a huge stakeholder management piece in it, actually. This goes back into defining the business problem you're trying to solve, and defining the conceptual framework of how you're solving that. For example, for the sales classification – let's say that we've got a sales call and we're going to try and decide whether the deal is going to win or lose. Is this going to be a successful call or not? Then the next step for me would be to think about what are the key concepts? What are the concepts a salesperson would be thinking about in this?

Let's say they would be using qualification methodologies, they would have a number of ways they can work through in breaking down the data into understandable chunks. “Which parts of these would be mappable into machine learning systems? Are we looking at it on an entity level? Are we looking at a document level? Paragraph level? You know, what is our unit of work that we're going to look at?” Then I think the really important bit is having this kind of prototype idea. This is where I think I've done things wrong, but I just love building things – and I just start building. I'm like, “Okay, let's build it. Let's do the modeling. It's so much fun. It's amazing.” It's the reason we do this work. And I just don't spend enough time on this initial bid, where I find that sharing even the first couple forms of labeled data through – I don't know, displaCy, for example – where you can actually show a document annotated with the labels on it.

If you share something like this with other execs, or you share this upwards in the initial stage of a project, I find that that really helps in getting some amazing insights in how to break down the problem from people who have huge experience in these areas. So I think stakeholder management is this kind of hidden conceptual, important thing. And in general, the other aspect is also from a bottom up view – there's going to be a process challenge. This goes back to how do you define the atomic elements of your problem? Okay, let's assume we're doing something – entity recognition. We're going to extract sales concepts. There, what you're going to have as a challenge is that these concepts – what you came up with, initially – they won't cover everything. At least I've never come up with something that covers everything. Almost always, some of them are wrong, or some of them are blind spots that I didn't even think about.

The thing that you're going to need from a process perspective is “How do you adapt to that? Can your model or system adapt to new concepts appearing – to new things of interest appearing?” I think, again, the really important thing to manage this is a process, actually – a human-centric process, in my opinion. I like to have the annotation booklet, which is just a document that has all of the tasks that you're working on. It has some task definitions, where we try and describe these concepts. We collect all of the ambiguous samples there. Then what we can do is we talk over those, both with the machine learning team and also annotators. We talked about ambiguity. We want to talk about these ambiguous cases – we collect them, we review them, and hopefully, we refine the task and the conceptual framework so that we can reduce ambiguity. I find that is super important. This booklet is part of a review process, where the annotation process would be reviewed each time. In my annotation processes, I will annotate the data – I will have some process to generate labeled data. Then my next step is, I will review that – I will have some kind of quality control on top of it. I think key parts of that would be like inter-annotator agreement, so I understand annotator performance. Then I will have some metrics around my model performance. But then I will also have feedback sessions with the annotators, ideally, where I can actually get their views back on “What was tricky? What was hard?” and have that information flow in order to learn from it.

This is one of the reasons I much prefer in-house annotation to crowdsourcing. But I think crowdsourcing can be awesome as well. In particular, when you're setting up a proof of concept or prototype, you can even get a crowdsourcing solution like Mechanical Turk to scale up real fast. But the cost of that, in my experience, is a quality and actually even a cost. So it will cost you more, I would say, for getting high-quality data and it wouldn't be as good quality as potentially using in-house annotation. In the long term, I much prefer in-house annotation, because the key advantage, in my experience, is that annotators who do a really good job are valuable. They can build up both institutional knowledge, but also their insights, their views, and if they're productive – their productivity is extremely valuable. One of the key advantages of in-house data annotation is that you can keep that relationship, and you can keep it alive, and nurture that relationship with your annotators.

15:39

Alexey

**Yeah, that was quite a lot to unpack – a lot of information. Let me try to summarize. I probably missed a few of the very important bits. When it comes to the process of actually collecting data, first of all, we need to have the process. The process is, first, we might start with external annotators to do some proof of concept, but then we should prefer internal annotators, because we can just come and sit closely to them and talk to them.**

**The process should be – annotators annotate the data, you also annotate the data, and then you collect some data, and then you monitor how good they do this. So you need to have some sort of metrics that describe how well they do the annotation. Finally, you need to sit with them and say, “What was difficult in this process?” This is how we do this, right? At the end, we'll probably save the results somewhere in a database or something like this. Then you mentioned that before you start this process, before you go to an external crowdsourcing platform, before you start working with internal annotators, you need to think about what exactly you're doing – what exactly you're annotating.**

**So you need to talk to somebody like domain experts, C-level people, and help them work through this problem of annotating with you so you understand what they want and they understand what you want from them. This way, you understand the value in this process.**

# Getting the buy-in from experts and executives

17:26

Christiaan

Yeah. Sorry, I think that was definitely quite a bit. The first point was definitely this pitch process. Because you will need some executives to kind of be your “patrons” and push your project. Most machine learning projects are big ticket items overall. In my experience, they cost between five hundred thousand to a million pounds, if you consider everything over a year, let's say. There are always some executives who are highly invested in this project. I think managing upwards there can be a real superpower for your career, and it will lead you to a better dataset as well.

# Starting an annotation booklet

18:08

Alexey

**Also, you mentioned this annotation booklet, where you collect ambiguous samples. I guess this is the feedback step, when you sit with annotators and ask them about what was difficult, and they say, “You know, this thing was tricky. I didn't really understand how to label this particular piece of text.” Then, what you do is take this thing and put it into some sort of Google document? Or?**

18:36

Christiaan

Yeah, it's just a collaborative Google document where we just keep track of the data state.

18:42

Alexey

**And what do you do with this? You collected these tricky examples, and what do you do? Do you say something like, “Next time you come across an example like this, this is how you should label it.”?**

18:53

Christiaan

Exactly. I find that the useful thing in this is trying to understand why this is ambiguous on any kind of conceptual level. Sometimes things can be two things at the same time – you just don't have a choice – and either label is correct, actually. You just have to live with it and deal with it. But sometimes it can be that you may actually need to break things down a little bit more – you need to refine your task labels. For example, back at Resolver, we did a lot of labeling of complaints in different categories of complaints. At one point, I think we had 21 different labels for labeling complaint documents. 21 labels! You have a huge amount of attention fatigue there. Nobody can label 21 things. I don't think I can keep 21 things in my mind at the same time *while* reading the document and doing some other stuff. It's very difficult for me.

That was just an example of something that we did pretty poorly. We just wanted to try it and we tried different distributions, like “What was the ideal number of labels?” when we have a huge number of labels for document classification, for example. I think it was between four and seven, depending on what we wanted and what sorts of accuracy we wanted. That's what we found. I think that the whole point of this process is – your annotation process is something that you should iterate on. And this booklet is kind of like your “problem list” or your “homework” that you need to fix. So this is your homework – sometimes you need to admit defeat and be like, “It's ambiguous. I can’t, sorry. This is it.” But other times I find that there is some conceptual work that you can do, or maybe improve the UI – maybe you can use some active learning and pre-label the documents. There's a number of process iterations that you could do there to break things down.

# Pre-labeling

20:57

Alexey

**This pre-labeling, I think I saw a tool that does something like this. Correct me if I'm wrong. We present a piece of a document and ask annotators to label it, right? It can be a part from the sales call and we say, “Okay, based on the chunk that you see, do you think the deal was closed or not?” And we already say, “We think that maybe this thing led to a closing. Do you agree with this or not?” So that would be this pre-labeling, right?**

21:32

Christiaan

Yeah. I think in general, you could do it like that. For example, with an interpretability layer, where you do some document classification and you will expose your interpretability layer to help people agree with the model or disagree. I think, for example, where this pre-labeling can massively simplify things, is named entity recognition, where potentially having to just review them quickly can have advantages. But obviously, the devil is always in the details and when you pre-label things, then things that *aren't* pre-labeled are much less likely to get labeled. It's almost like each decision you make in your dataset creation process, there are some trade-offs going on there.

What I've learned is that you just need to run experiments, at the end of the day. Often, your assumptions can be wrong. If there are highly valuable but kind of rare samples in your dataset, then you will need to do a lot more experiments, I think, and common sense can help you less because of the nature of your dataset. If you're not focusing so much on outliers, which are extremely hard to actually label, then I think pre-labeling can benefit you. But sometimes it doesn't [chuckles]. It also kind of depends on your model.

# Dataset collection

23:05

Alexey

**Now I want to take a bit of a step back and talk again about this dataset creation process. We talked a bit about this, and you mentioned that we first need to get domain experts and execs on board – work with them. Then you said how the process should be organized – you start annotating data, then review, and then get feedback. But before we can actually start annotating the data, we should think about “What is the task that people need to do?”**

**From my experience of using crowdsourcing, the quality of the data that you get highly depends on how well you define the task. Maybe we can talk a bit about this. What does the process of collecting a dataset actually look like after we talk to execs, and so on? How do we take this and create something that we can give to annotators?**

24:01

Christiaan

This is a key insight, Alexey. The really good thing is – the way I see it – is that experts and execs can give you a blueprint. You have a proposal document and you will, ideally, interview some experts as well to find out how they actually do it or how they think about this. What I find is that it's a translation problem. I take my interviews, for example, with the experts and I try and figure out what levels can I break that down into? For example, for sales modeling, maybe I will take some qualification methodologies that are about the pain points of the customers or about different attributes of the customers and I think “Okay, what would this be as a task? What would this look like as a task?”

This is why actually having this kind of business guidance is a huge benefit. Because when you have annotated the data yourself, which I think every one of us should do a lot of when starting a project, then you realize that some things are gonna be quite tricky. I think it's important to communicate clearly before you get Stockholm Syndrome from your own projects. Because after you've worked on something for a year or six months, it's very clear to kind of assume that your assumptions are correct and you're going in the right direction. But that's why these initial conversations are so powerful, because it's literally how experts have explained it to you how things are. So maybe you can use the same already working explanation to explain how things are to annotators or who will need to actually create this data for you.

26:00

Alexey

**How do you capture this? Because I guess when an expert tells you this and you record it, it's like a wall of text. That's a lot, right? You somehow need to process this and summarize it. How do you do this?**

26:14

Christiaan

I find what works really well for me is – I will run interviews with experts, where I collect this wall of text. I think it's really good to come up with a mind map where you just unfold the concepts. I really like this idea of thinking about this in conceptual terms and I think it's a bit of a blind spot in data science sometimes. We really like maths, we really like programming and sometimes we less like the communication aspect of this – and there is one. So then I come up with this mind map where I'm like, “Okay, these are, let's say, my ideas of building blocks and this is how they associate back.”

For example, with sales, let's say we're doing some sales qualification stuff and I'm extracting these sales qualification attributes and these are my building blocks – my model may just work on text. It may just work on the whole text and I'm not going to do any named entity recognition. But maybe my interpretability layer can have some insights – that would be a massive win if they can be associated somehow with the qualification methodology, for example. This mind map isn't like the way to train us down, but it's more of a way for us to think about “How is this actually built up conceptually?” And I think the really powerful thing if you have this mind map when you’re making the slides or short documents, is that this is going to be the basis of my annotation booklets.

When I do a kickoff, often it may just be like a slide-based kickoff with the annotators, and I want to actually maybe get some… when I worked with some quite experienced annotators, I want their views on what they think will be tricky about this – what they think about what could be the risks. So this is obviously more of a scenario where you've got more domain expertise involved. So for example, in bio-NLP, you definitely need to involve curators a lot more – you've got a whole level of expert consultation layer on top of this, or at this stage there. And there, I think it's a lot more difficult as well because there are no clear answers. So my mind map won't necessarily help figuring out how to distinguish some genes and proteins, for example, between each other. It's probably not going to be enough for that. But then, in that case, I think it's more about having some project managers or product managers who can also help manage some of these technical challenges and actually make them into the roadmap.

It kind of becomes a little bit of a product challenge, I think, which is extremely hard. But this conceptual framework and a mind map, I think can take you quite far ahead. It's not so simple that just words are enough or just a not so good document. Just make some slides, some pictures, and really focus on it. It's an economic thing as well, the more your annotators understand what you're doing, the better work they will do and you're going to increase your chances of succeeding.

# Human level baseline and feedback

29:28

Alexey

**In summary – first, we talk to domain experts and we have them annotate the data. We interview with them, we watch how exactly they annotate, and we record everything. Then we build a mind map and we try to annotate the data ourselves to really make sure that we understood them – what are the key cases we see ourselves? Maybe we can go back to the experts and see “Okay, what do you think about this one? I'm not sure.” This way you extract the knowledge from them and then you put them in a mind map. And then it's your turn to share this mind map and this knowledge extracted from experts with annotators. Then annotators probably go through a similar process as you just did when learning from experts, and then the process starts. Right?**

30:17

Christiaan

Yes. In my experience, experts outside of bio-NLP are usually not the experts who will do the annotation, but I would do the initial annotation instead. It's usually difficult to get sales leaders to do annotation for you, or other domain experts. So there, it’s more that I run an interview with them, then I do some basic annotation myself and then maybe I'll even send them a document and be like, “Hey, this is based on what you told me. This is how I think where the concepts are that you describe – this is where I see them.”

Obviously, I'm not going to do as good a job as somebody who's professional to identify these concepts, but it also potentially gives me a good idea of what's achievable as well. The experts are, let's say, above me in the quality of what they do, but usually, I'm above or the same level as the annotators. That also gives you a good idea of what's achievable and it gives you this kind of human-level baseline before you start into this project. I think that's a really important way of de-risking things as well. If you can use this human level baseline, the real hard thing then is tying that back to the business value. Now we've got a human level baseline, we’re almost ready to start the project – we've got an executor who's happy to support this, we've got a conceptual overview – we've kind of got an idea how we're gonna get the data.

But then comes the next leap, which is like, “What is good data? How do we tie this back to business value?” And I think this is extremely difficult. [chuckles] My approach here is trying to come up with a prototype. This is why I think sharing even some form of annotation or some form of data visualization of the human baseline is so powerful. So I want to share that to start managing expectations of what's achievable from the project. And also to get support from the business leaders as well around “How could I translate this to business value?” I want to see where *they* see that this could add value already in the beginning – even before I have my mobile deployment.

32:32

Alexey

**So the way I understood you is – humans make mistakes when annotating. It's inevitable. There will be some accuracy that humans can provide. Usually experts are most accurate, and you and the annotators are less accurate. Then you have this data with some inaccuracies and then perhaps for this sales qualification task, let's say, human data says 70% correct. Then you can come back to the experts and say, “Okay, if our system is 70% correct, do you think it will be useful for the business or not?” Right?**

33:08

Christiaan

Exactly. But I think this process is still qualitative and more consultative. What I would do is actually take some examples of, usually, what I've done. Actually, that's the level where I do this. I'll come up with my own document like, “Here's literally a transcription that is annotated and in a nice, easy-to-read way.” And I will send it to them and ask them to read it, look at it, and share their thoughts – like, what did they think?

Because sometimes, the feedback could be like, “Hey, Chris. This is okay, but you're missing three concepts here. If I was just focusing on these two things, I don't think I would be able to do this at all.” It can be something like that. Another thing, it can also be like, “Oh, this is really interesting, actually. If our customer support people had access to this level of tagging (for example) maybe we could speed up complaint resolution by 5%. This is really exciting and this could be a track where we can provide value with this feature.”

34:16

Alexey

**So not only do you understand if it's useful for the business, but you can also get some insights on how exactly it will be useful. Maybe it's different from what you initially thought, right?**

34:25

Christiaan

Yeah. In my experience, there's almost always some [chuckles] new emergent ideas that come along.

# Using the annotation booklet to boost annotation productivity

34:33

Alexey

**We talked a bit about this annotation booklet, and you mentioned that we put tricky examples there. But then you also said something like “When we start this process, we give this booklet to the annotators.” My understanding is that these tricky examples are not the only part that we put in the booklet, right? We probably put the entire task definition, we give examples, we give this mind map that we talked about, etc. So what else do we put in there?**

35:02

Christiaan

Exactly. The booklet – the way I see it – it's a complete guide to being as productive as possible in the annotation process. The objective there is to empower annotators to do as good a job as possible. I think this is a very important mindset in data creation – to have empathy towards annotation. It's a hard job. It's really difficult. And to really think about, “Okay, how can I make this easier? How can I make this work better?” The booklet, for me, is this living document that has what the task is, how we actually conceptually think about it, like, “Why is this the kind of thing we're interested in?” And the third bit is kind of more the craft aspect – here there are ambiguous ones.

I also think it's also important to allow it, in a way, for annotators to potentially share notes, for example, when they're doing annotation. And then you or a project manager can collect those notes into this annotated booklet. Then you periodically review it – you would review them initially and then you would debrief your annotation team, where you could discuss, “What are the insights from this? What are the changes that you're going to do to kind of react to that?” Now, I think this is really important, because when annotators feel that they're listened to, it's very important in a work relationship. I find that it can be a lot easier to work together.

# Putting yourself in the shoes of annotators (and measuring performance)

36:37

Alexey

**You mentioned “How can I make it easier for annotators?” and the booklet is a way to make it easier. I think for me, personally – I remember when I needed to do something like this and it involved annotators from the company where I worked – I would do this myself and then see where it's not easy. Because I think this is what we, data scientists, sometimes don't do – or don't do enough – is try to… how do you say? Eat your own dog food. Right?**

**Try to put yourself in the shoes of the annotators and then feel the struggles of how boring it is, how many actions you need to do to annotate a piece of text. Then to think, “Okay, how can I actually make it faster? Maybe instead of using the mouse a lot, maybe you can just press a key button on your keyboard?” And things like this. So this is when you get insights – when you try to do this, right? I guess you also came to a similar observation.**

37:42

Christiaan

Yeah, I think annotation user experience is massive and it's also measurable. I'm a huge fan of this whole annotation process. You can have a very quantitative and database approach to how you measure the impact of these things. For example, at Resolver, we used Prodigy spaces on the annotation interface, which has one of these beneficial aspects. It has hotkeys, for example, for doing quick acceptances of name entities or even classifications. It makes it a lot easier with the UX and we would see potentially 5-10% improvements in how many data samples we could get from an annotator in a day by iteratively improving the UX, going for a better user experience there.

I'd say that there are three metrics that are really important to keep track of. The one that I've already said is inter-annotator agreement. I think this is maybe one of the most important ones. Because if there's very low inter-annotator agreement, it means your task is very ambiguous and people have no idea what they're doing – or it's just a very difficult task and then you may need to re-figure out what you're doing. Then the second metric that’s quite important is, “Okay, how many samples of data can you get from annotators in a unit of time – in, let's say, eight hours, for example. I think it's important to keep track of this, to make sure that the performance is on track there.

One of the difficult things to model there is fatigue as well, because, again, when people are doing crowdsource annotation, they may do like a 10-12 hour shift of mechanical jerking, let's say. By hour nine… even if you do it yourself – if you did that 12 hours of annotation a day, I'm going to have very strong questions about the last few hours of your annotation and the output from there. So modeling fatigue can be a challenge there. But you *can* track that as well if you look at the rate of data and look at the rate of the quality of the data. But it's a bit harder, I think. And I think that the final piece is probably real-time model metrics around performance.

What we did at Resolver, that I thought was quite clever, is we would do a split of the data where we would leave out particular annotators’ datasets and we would test on those, for example, and see how well our models would generalize to different splits of annotators’ data in different time periods. After a while, we had some concerns that some annotators were annotating things very differently and this was something that emerged. We did a project around identifying vulnerable consumers who are most at risk when they're making a complaint. Some annotators were thinking, “Winter is a real problem.” And it is a huge problem. This was actually one of our big blind spots that we discovered through the annotators – is that in winter, in the UK – take heating, for example, your boiler breaks. It's extremely difficult, obviously. You're going to freeze to death, if nothing happens. So it’s an extremely vulnerable situation that needs top priority – needs a lot of focus and attention from companies. And companies do have specific departments for this and teams to work with these types of consumers.

But we didn't expect it to be such a huge percentage of vulnerable complaints. It was more than 10%. And we found this because there was a particular annotator who was more focused on this and we would then share these users with us. What led us to actually find this, is that we did periodic qualitative looks as well. We would periodically kind of read about 100 annotations a week by different people, so that we would get an idea of like, “What are people picking out? What are they looking at?” So I think eyeballing the data is extremely important. All of that would be very hard for me to do if I didn't start the whole process – like you said as well – myself doing a lot of annotation. When I start a project I do about… it really depends – between 500 and 1000 data points, let's say. And that's usually the point where I get somebody external involved (other than myself) which is brutal, I have to say. I find it very difficult during the 500 to a 1000 samples. But it’s very valuable.

# Active learning

42:51

Alexey

**We talked a bit about pre-filling some of the suggestions. Even at the beginning, I talked a bit about this “active learning” when you collect a bit of data, then you train your model, then you show this to annotators and then you iterate. So maybe we can talk about these things. One thing I wanted to ask you about active learning, “How do you think it's helpful in dataset collection?”**

43:18

Christiaan

I think active learning can really work and it can help you massively reduce the data amounts that you require. Sometimes it can be quite less impressive as well, in my experience. The general idea in active learning is that you get model predictions and you sample low confidence model predictions or model predictions on decision boundary, and then you annotate those. That will help push the model in the right direction. The model keeps training while you're feeding it this data. This is one of these hybrid data collection approaches.

In my experience, it hasn't worked extremely well so far. Or maybe it was hyped up a lot when I started using active learning and I was surprised by the smaller impact than I expected. When it worked for me, it was usually about 20% less data required than without active learning – when it worked. But the problem I've had – and 20% is fantastic still, so maybe it's just kind of my dream-like expectations – but I honestly felt it would be a much larger force multiplier. I thought it would be a complete game changer, let's say. And sadly, active learning is not a complete game changer, but it can work sometimes extremely well and other times it kind of falls on its face a bit.

# Distance supervision

44:50

Alexey

**Then there is another thing called “distance supervision”. Can you tell us about this thing? What is it?**

44:57

Christiaan

Yeah. So distance supervision – *that* is actually a game changer, I think. Distance supervision is the paradigm where data creation is moving towards. What distance supervision is, is when you can use some kind of programmatic approach to generate weak labels for your dataset. And then what you can do is either fine tune your model straightaway based on that, or you may decide to sample from that collection of weak labels. For example, at Resolver, we had a semi-supervised topic model and we would sample vulnerable complaints from there. And that was a force multiplier – it led to requiring 10 times less data for finding vulnerable consumers. So, it was a *huge* force multiplier.

Today, this technology has matured even more. There are tools like Snorkel, for example, where you can define these labeling functions. Snorkel has, for example, integration with spaCy, so that's quite useful. You can define named entity based labeling functions, so if there's a location in this document, then you may want to say like “It has location,” for example – even something like that. Then what you do is create all of these kinds of weak labels and then what Snorkel does is create a clever weighing on top of that to see how that aligns with the actual labels that you want to generate.

Obviously, some combinations of these labels will be more successful than others. I think this technology is extremely powerful, because it kind of allows domain experts and annotators to have a much wider range in doing this. Because when you come up with a labeling function, it may affect, I don't know, 2-3% of your data per labeling function – and that's amazing, actually. You're using a much *broader* net to collect your data. The quality that you're collecting is lower, though still. So you will still need to do some more traditional notation, or maybe a subset of the data, maybe even on data out of your distance supervision distribution, because you may have some biases there as well, that you're introducing with this. But I think distance supervision is a huge force multiplier in the industry currently. It's one of those things that really empowers you to get more done with limited time and limited budget.

# Weak labeling

47:45

Alexey

**You mentioned one of these sources for weak labeling is topic modeling. Let's say we have a huge pile of unlabeled text. It could be transcriptions from sales calls, for example. So what we can do is somehow cluster this text into a bunch of topics and then this topic that comes out of our clustering algorithm could be this “weak label”. That could be one of the sources. What about different heuristics like, “If we see a certain word, then we think that it could be *this* label.” Is this also a good source of weak labels?**

48:24

Christiaan

Yes, yes, yes. This is exactly the type of programmatic labeling functions that Snorkel, for example, allows you to create. There are some other tools as well. Or you can roll this somewhat yourself as well. Personally, I've rolled it before I knew about Snorkel myself. I think a good example to understand this better is maybe bio-NLP. When you're developing a drug you're looking for a particular drug that treats a particular disease. So if you find a sentence that has a drug and a disease entity in it, and the verb in it is “treat,” for example, then that's a good candidate. Then you're like, “Okay, this drug treats this disease based on this document.”

That's one way you could actually generate this type of label. I think you can take this quite far. And then you could maybe do a more fuzzy version of this – if there is a verb and a drug and a disease, then obviously you're gonna get a lot less lower quality because maybe it doesn't treat it. Maybe that's what the sentence is saying if it's a PubMed abstract. Or that it has negative side effects or something like that. This is the kind of fuzziness in how well you specify these labeling functions.

This is why I think it's quite important – what Snorkel brought to the table, I think, as a user, was having this type of clever weighing mechanism on top of these labeling functions. There you could define them both as, “Does it contain the string?” You could use all of these spaCy linguistic features like named entity recognition, part of speech tag, etc. all of these types of things. And then you also have a layer on top of this that can weigh that for you to make sure that you're getting the best bang for your buck and low quality labeling functions aren't pulling you down.

50:25

Alexey

**You mentioned two tools – you mentioned Prodigy at some point and you mentioned Snorkel. So what are the other options? Or is one of these two already enough to get started?**

50:37

Christiaan

Personally, if I was starting out now and I would just be doing my first proof of concept project, I would start with Prodigy, because I think Prodigy has a really good user interface. It integrates very well with spaCy because it was created by the creators of spaCy, so it has a very nice user experience. It has these hotkeys and everything. It's just a pleasure to use.

I think Docanno is a quite good open source alternative, or Labeling Studio. Those are both open source alternatives to Prodigy. They both allow you to do active learning, actually. And for distance supervision, I would recommend Snorkel, probably. Snorkel has an open source version of their tool, which they aren't developing actively anymore as they've moved into enterprise development. But it's still usable. I think it's a good entry point into distance supervision because it has a nice interface that allows you to find these labeling functions in there.

Another open source tool that has less powerful distance supervision features, but I think is quite inspired by it, is Rubrics. I haven't used that myself, I just skimmed it, actually. It looked like a similar alternative to Snorkel. If anybody watching this has some experience with Rubrics, please message me, because I'd really love to learn more – or if you have more alternatives there. Because I think this space is kind of blowing up now – as in, it's becoming extremely important. And I think this can be a large competitive advantage when you're creating your dataset.

# Dataset collection in career positioning and project portfolios

52:34

Alexey

**We have a question from the audience, which is actually similar to one of the questions I prepared for you. So “What we talked about here, how would we take this and apply to career positioning?” For example, somebody wants to change careers (to become a data scientist, for example) and they're building a project portfolio. How should they go about this? Because one thing you can do is take a project from Kaggle. It's a ready CSV file – somebody already put effort in collecting this data. So you just use Pandas, read the CSV, and then you use logistic regression fit, and you just use “predict” and here you go – this is your project portfolio. Everyone has these projects, but very few people actually work on collecting datasets. Very, very few. This could be like a good way to get noticed, right? So how would you suggest that people can do this? How can they use these things in building their projects?**

53:41

Christiaan

If I put on my hiring manager hat for just five seconds, I would *love it* if a candidate would tell me about their dataset creation experience, or they would be like, “Hey, this is a dataset I've created.” I think that would be mind-blowing. To me, that would put them into a way more mature category. I would think they were way more valuable than an entry level data scientist, because this type of conceptual thinking and looking at the data – the data is expensive to create and it's valuable to create it well. That type of maturity would be hugely, *hugely* valuable. Taking off my hiring manager hat. [chuckles]

If I was starting out on a project like this, I think the most important thing is to actually do something. In machine learning, there's millions of tools, there's so many things you could do, so many documentations, there's a huge information overload going on. My key suggestion would be “Just start doing something.” And it can be good if you do it in a simpler way initially, and it's more painful or something like that, because you will want to do it in a better way then. I think it's really important to just have a project, make it a simple project, but maybe something that's of interest for you and where you may have some competitive advantage. If you're a domain expert – and you are probably domain experts in some areas – then you should make a dataset using that domain expertise.

Then the next thing I would suggest is select your stack and just stick with it. If you would start out now – for a beginner – if you would do a project with Snorkel, it would be a pretty strong way to distinguish yourself from most candidates. It could also teach you how to use spaCy, do some of that linguistic preprocessing as well, and learn a little bit of computational linguistics through that. So as somebody who works in NLP all of these things would be extremely good signals. If you use a simpler tool, let's say, Docanno or Labeling Studio and create a dataset like that, I would still be quite impressed with that project. I think people who work through something like that, it's very impressive because a huge amount of the cost basis is there.

As we're seeing more and more modeling is becoming a commodity and anybody can kind of plug and play transformers, for example, to get to some kind of baseline, you need to think about where you can provide the competitive value that there's a lack of in the market. I would say data creation experts are something that we need more of, and we need to have more discussions about these types of things. It's less sexy, in a way. I'd say it's a lot of fun to train a model and it's *hugely* rewarding when you finally have your model and it's making some good predictions and you see the pleasure that it gives to your users – very rewarding. But building the dataset to do that is also very rewarding, because without this, you wouldn't have gotten there. I think it's really important to frame this narrative around “Let's talk more about dataset creation and also make it also.” Maybe I won't be as cool as very fancy machine learning models, but hopefully it will get a bit cooler.

# IPython widgets

57:18

Alexey

**In my personal experience, you can just start using IPython widgets, like widgets in Jupyter Notebook. It's super easy to start with. It's not as advanced as Snorkel or Prodigy, but if you need some binary classification case, then you can quickly get a few hundred examples just from using IPython widgets.**

57:44

Christiaan

If you like IPython widgets, and you're more of a beginner (but what IPython is) then I would suggest the Fast AI course, because the whole Fast AI course takes this idea of using IPython to crazy, crazy levels – to the next level in every way. It can be a great, structured way of building a project where you will have a dataset creation and modeling piece on your own, for your own domain, but getting guidance throughout the process for free. So that would be my suggestion, based on that.

58:20

Alexey

**Yeah, I realized we don't have a lot of time left. Do you have a couple of minutes?**

58:25

Christiaan

Yes, I do.

# GDPR compliance and non-English NLP

58:26

Alexey

**Good. We have a few questions. Maybe what I will do is suggest to you two questions and you will pick the one you want to answer. The first question is about dealing with GDPR – we can have sensitive data, how do we present it to annotators? And then another question is about different languages that Comtura can capture and what are the different challenges with working with various languages? Or if you have time – you can answer both.**

58:55

Christiaan

Yeah, I'll answer them both. The first question was around GDPR compliance. Essentially, GDPR, I think, is another great reason to favor in-house annotation, actually, because crowdsourcing has a huge amount of compliance risk, potentially, and you could be leaking personally identifiable data. There's a number of anonymization techniques that I've used as well to try and blank out locations, names, phone numbers, credit cards, or other personally identifiable information. But these algorithms usually aren’t perfect and some data leaks through. Like identifying names, working at Resolver, I realized there's a huge amount of different names that are not standard at all and are extremely difficult to actually capture and often they're typed with small letters as well. It can be a very, very tricky thing. Personally, I think this is a huge plus for in-house data annotation, because then you can manage the sensitivity of the data way better.

With regards to what languages Comtura can support – at the moment, we only support English. But if you're interested, get in touch with me on one of the channels that I shared and I'm happy to have more of a chat about whatever language you're interested in applying Comtura to.

60:26

Alexey

**You’re Hungarian, aren’t you?**

60:28

Christiaan

Yeah. Yes. I'm half Dutch, half Hungarian, actually. But I grew up in Hungary.

60:35

Alexey

**I'm asking this because I know – I haven't studied Hungarian – but I know I heard that this is an extremely difficult language for a foreigner to learn. I imagine that for NLP tasks, it's also quite tricky because of the grammar, because of the linguistic properties of the language. Have you worked with other languages apart from English?**

61:00

Christiaan

I have done some extremely minor work with Hungarian. Actually, Hungarian is in many ways easier than English. One of the things about English that is quite difficult – there's a lot of morphological ambiguity. When you have a word written in English, it doesn't really give you a good idea about how to pronounce it. You could pronounce the same vowels in quite a few different ways. But in Hungarian and in many, I'd say more sane languages, if you have something written in some way, it gives you 100% knowledge in how to pronounce it – you will be able to read it and you won't need any additional information to be able to read the word. So that's one benefit.

The other one is – this is a bit more around tokenization – Hungarian is an agglutinative language, so all the linguistic information is stored at the end of words in general. While English will use prefixes and suffixes potentially, but mainly prefixes to kind of load up their linguistic information. What is the challenge is potentially “How do you tokenize your strings to capture that information?” But actually, transformer models, I believe, do work in Hungarian as well. They can learn these suffixes and the linguistic information. I'm assuming this is due to the word piece level tokenization that's going on there. But I haven't looked into this extremely deeply. But surprisingly, NLP in Hungarian works quite well. The thing that makes it less advantageous is Hungary is quite a small country of only 9 million people. The impact of Hungarian NLP is more limited than English speaking NLP.

62:59

Alexey

**So that's the main reason, right? It's just how common the language is – how many people speak it. I remember I came across a tweet recently, that said that in the phrase “Pacific Ocean” all the C’s are pronounced differently. [chuckles] I guess that doesn't happen in Hungarian, does it?**

63:22

Christiaan

No, no, no, it doesn't. [chuckles] It doesn't. Yeah, I think many languages are morphologically more stable, I'd say, than English. English is not an easy language in many ways. I’d say. So [chuckles].

# Finding Christiaan online

63:40

Alexey

**Okay. Thanks so much for staying a bit longer with us and answering questions. And thanks, everyone, also for asking questions. Thanks for sharing your expertise with us. We have your contact information. We'll put this in the description. There is somebody asking to connect with you on LinkedIn. So we'll share all the information. Definitely. And I guess there are not so many Christiaan Swarts on LinkedIn.**

64:07

Christiaan

Yeah. My LinkedIn user name is Christiaan Swart with two A’s, actually. That’s the Dutch way of spelling Christiaan and my startup is called Comtura. I think if you put in my name and Comtura, you should get a hit for me. If you struggle to get in touch I have a blog – UseML.net – and you can get in touch with me there. So if anybody struggles in another way. Alexey, thank you for conducting this interview. It was a real pleasure. I think this community that you're building is awesome. I think the work that you're doing with this is really cool. I'm really happy that I could be part of it for an interview. And I'm always looking forward to seeing what you're up to.

65:02

Alexey

**Thank you for your kind words.**